Journal of Network and Computer Applications 174 (2021) 102894

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Tensor based framework for Distributed Denial of Service attack detection

João Paulo A. Maranhão [a,*], João Paulo C.L. da Costa [a,b], Elnaz Javidi [c], César A. Borges de Andrade [a], Rafael T. de Sousa Jr. [a]

[a] Department of Electrical Engineering, University of Brasília (UnB), 70.910-900, Brasília-DF, Brazil
[b] Department 2-Campus Lippstadt, Hamm-Lippstadt University of Applied Sciences, 59063, Hamm, Germany
[c] Department of Mechanical Engineering, University of Brasília (UnB), 70.910-900, Brasília-DF, Brazil

ARTICLE INFO

Keywords:
Network attack detection
Machine learning
Supervised classification
Multiple denoising
Tensor decomposition

ABSTRACT

Distributed Denial of Service (DDoS) attacks are one of the most important security threats, since multiple compromised systems perform massive attacks over a victim, overwhelming its bandwidth and/or resources. Such attacks can be detected, for example, by using supervised machine learning based solutions previously trained on large DDoS attack datasets in order to automatically identify malicious patterns present in the incoming traffic. In addition, since large datasets show inherent multidimensional structures, tensor based detection techniques can outperform the matrix based counterparts. In this context, the development of a DDoS attack detection framework which exploits both machine learning and tensor based approaches is crucial. To face this challenge, this paper proposes a novel tensor based framework for DDoS attack detection using concepts of multiple denoising, tensor decomposition and machine learning supervised classification. Moreover, we also propose an extension of the recent Multiple Denoising algorithm such that the noise present in the dataset instances is more efficiently attenuated. Finally, we validate the effectiveness of our proposed framework through comparison with state-of-the-art low-rank approximation techniques as well as with related works. The proposed approach outperforms its competitor schemes in terms of accuracy, detection rate and false alarm rate.

1. Introduction

Network intrusion detection plays a fundamental role in the process of protecting critical networks by monitoring and analyzing suspicious activities, incidents, threats and violations. In this context, since Intrusion Detection Systems (IDS) provide forewarnings about malicious behaviors, such as intrusion attempts and malware, they are used by security administrators in order to detect and countermeasure sophisticated network attacks (Sharafaldin et al., 2018).

Distributed Denial of Service (DDoS) attacks are one of the most important threats to network security. For example, in 2018 the developer platform GitHub was hit by a huge DDoS attack which reached 1.35 terabits per second during a period of 15 to 20 min (GitHub.com, 2018). Additionally, according to the Cisco Annual Internet Report (2018–2023), DDoS attacks correspond to 25% of the total Internet traffic of a country during their occurrence (Cisco, 2019). Still according to Cisco (2019), there will be 14.5 million of DDoS attacks in the world in 2022, implying into major security threats to governments and corporations.

In order to launch DDoS attacks into target systems, hackers can make use of legitimate third part components (normally web servers) through the combined effort of thousands of compromised machines known as "zombies". Such zombies establish a "zombie network" that exhausts victim's bandwidth or resources through a massive traffic attack, while hiding the attacker's identity. Moreover, the weaknesses of different network layer protocols are exploited by an attacker and, consequently, the victim uses huge CPU and memory resources to process intensive operations (Mahjabin et al., 2017).

In this sense, it is fundamental that network administrators adopt accurate and efficient schemes in order to detect and prevent DDoS attacks in their organizations. For instance, tensor based signal processing techniques have attracted an increasing attention in the last years since they allow us to better exploit the inherent multidimensional structure of large datasets (Zanatta et al., 2019; Gomes et al., 2019). Furthermore, supervised machine learning (ML) based methods can provide an efficient way to detect DDoS attacks (Hosseini and Azizi,

* Corresponding author.
E-mail address: joaopaulo.maranhao@ieee.org (J.P.A. Maranhão).

https://doi.org/10.1016/j.jnca.2020.102894
Received 19 November 2019; Received in revised form 7 May 2020; Accepted 26 October 2020
Available online 1 November 2020

2019; Wang et al., 2020). As ML algorithms can be trained on benchmark datasets provided by cybersecurity institutes (Canadian Institute for Cybersecurity, 2009, 2019), such schemes can be used to identify, with high reliability, malicious patterns eventually present in the input network traffic in an automated fashion.

Since tensor based signal processing techniques as well as machine learning based algorithms have shown high performance when considering large datasets and network intrusion detection problems, respectively, we propose a novel framework for DDoS attack detection which exploits both approaches. The proposed architecture is composed by four steps: data preprocessing, dataset splitting, dataset denoising and machine learning classification. Moreover, in the third step we propose an extension of the recent Multiple Denoising (MuDe) technique, which attenuates the noise present in the dataset instances. Experiments show that the proposed framework achieves satisfactory performance, with outstanding values of accuracy, detection rate and false alarm rate when compared with traditional low-rank approximation techniques as well as with related works.

Hence, the major contributions of our work are summarized as follows:

- We propose a novel architecture that combines the benefits of both multidimensional signal processing techniques and supervised machine learning classification algorithms with the aim of providing accurate and efficient DDoS attack detection.
- We extend a recent multidimensional noise reduction technique known as Multiple Denoising (MuDe) in order to attenuate the noise present in the instances of DDoS attack detection datasets. The traditional MuDe was proposed by Gomes et al. in Gomes et al. (2019) and originally was intended to reduce the noise level in measurement data collected by multidimensional sensor arrays in radio communication systems. Our proposed extension is based on two main factors: (i) the application of the traditional MuDe directly on the dataset instances, and (ii) the inclusion of a second denoising stage performed by HOOI low-rank approximation such that a higher degree of noise reduction is achieved, with significant gain on the overall DDoS attack detection performance.

The remainder of this paper is organized as follows. Section 2 presents the related works. In Section 3, the data model is introduced. Section 4 presents the proposed framework for DDoS attack detection. In Section 5, simulation results are presented and discussed. Section 6 shows the computational complexity of the proposed extended MuDe technique. Section 7 draws the conclusions and the suggestions for future work.

2. Related works

To face the challenge of DDoS attack detection, schemes based on traditional signal processing techniques have attracted a great attention in the last decades. A Model Order Selection (MOS) technique for blind automatic malicious activity detection in distributed honeypots was proposed by David et al. in David et al. (2011), where human intervention or information about attacks were not required. In line with the ideas of David et al. (2011), Da Costa et al. proposed a blind automatic scheme to detect malicious traffic in network data collected at honeypot systems (da Costa et al., 2012a) as well as the R-D Akaike Information Criterion and the R-D Minimum Description Length to automatically identify malicious activities in honeypots (da Costa et al., 2012b). In addition, it is worth to mention the recent work of Vieira et al. which proposed a framework in order to detect the number of port scanning and flood attacks by analyzing the largest eigenvalues in time frames after applying MOS and similarity analysis on the dataset (Vieira et al., 2017). However, since the approaches in David et al. (2011), da Costa et al. (2012a,b) and Vieira et al. (2017) are not tensor based solutions and do not consider automatic learning, we fill those gaps by exploiting the inherent tensor structure present in large datasets as well as by applying classic machine learning classification algorithms such that the proposed technique learns to recognize patterns in multidimensional data.

Finally, machine learning based schemes have also been successfully used for DDoS attack detection. Osanaiye et al. (2016) presented an ensemble based multi-filter feature selection method for DDoS attack detection in cloud computing where the output of filter methods are combined to achieve an optimum selection. Furthermore, a model based on artificial neural networks and black hole optimization algorithm to detect DDoS attacks in cloud computing was presented in Kushwah and Ali (2017). Moreover, in Hosseini and Azizi (2019), the authors proposed a hybrid framework based on data stream approach for DDoS attack detection where the computational load is divided between the client and proxy side. Finally, Wang et al. proposed to combine feature selection with multilayer perceptron to select the optimal features as well as designed a feedback mechanism to perceive detection errors dynamically (Wang et al., 2020). Thus, despite the machine learning based schemes proposed by Hosseini and Azizi (2019), Wang et al. (2020), Osanaiye et al. (2016) and Kushwah and Ali (2017) show high performance in terms of DDoS attack detection, they did not exploit multidimensional techniques. Therefore, we also fill such research gap by adopting and extending tensor based denoising approaches, particularly the recent MuDe scheme, such that the inherent tensor structure of large datasets can be exploited more efficiently.

3. Data model

This section presents the data model adopted in this work and is divided into four subsections. First, the mathematical notation used in this work is shown in Section 3.1. Next, Section 3.2 provides a brief description about the data modeling. Then, Section 3.3 presents the analyzed scenario as well as details about different types of DDoS attacks. Finally, Section 3.4 introduces the DDoS attack datasets used in this work, namely, CICDDoS2019 and NSL-KDD.

3.1. Mathematical notation

This subsection presents the mathematical notation used along this work. Italic letters (a, b, A, B) denote scalars, lowercase bold letters denote column vectors $(\mathbf{a}, \mathbf{b})$ and uppercase bold letters denote matrices $(\mathbf{A}, \mathbf{B})$. Higher order tensors are represented by uppercase bold calligraphic letters $(\mathcal{A}, \mathcal{B})$. The superscripts $\{.\}^{\mathsf{T}}$ and $\{.\}^{\mathsf{H}}$ are used for transposition and Hermitian of a matrix, respectively. The operator $\mathrm{diag}(\cdot)$ transforms its argument vector into the main diagonal of a diagonal matrix.

The element in the nth row and mth column of a matrix $\mathbf{X} \in \mathbb{R}^{N\times M}$ is denoted as $x_{n,m}$, while $\mathbf{X}_{n,:}$ and $\mathbf{X}_{:,m}$ represent its nth row and mth column, respectively. The Khatri–Rao product, outer product and Kronecker product are represented by operators $\diamond$, $\circ$ and $\otimes$, respectively.

The matrix $[\mathcal{X}]_{(r)}$ corresponds to the rth mode unfolding of the tensor $\mathcal{X}$ and can be obtained by varying the rth index along the rows and stacking all other indices along the columns of $[\mathcal{X}]_{(r)}$. The r-mode product between the tensor $\mathcal{X}$ and matrix $\mathbf{B}$ is given by $\mathcal{Y} = \mathcal{X} \times_r \mathbf{B}$, which can also be expressed in a matricized fashion as $[\mathcal{Y}]_{(r)} = \mathbf{B}[\mathcal{X}]_{(r)}$.

3.2. Data modeling

Throughout this work, we assume that $\mathbf{X} \in \mathbb{R}^{N\times M}$ and $\mathbf{y} \in \mathbb{R}^{M}$ are respectively the dataset matrix and the corresponding class label vector, defined as

$$\mathbf{X} = \mathbf{X}_0 + \mathbf{N}, \tag{1}$$

$$\mathbf{y} = [y_1 \quad y_2 \quad \cdots \quad y_M], \tag{2}$$

where $\mathbf{X}_0 \in \mathbb{R}^{N\times M}$ is the free-noise dataset matrix, $\mathbf{N} \in \mathbb{R}^{N\times M}$ is the noise matrix, N is the number of features and M is the number of data instances. Each row $\mathbf{X}_{n,:}$ for $n = 1, \ldots, N$ corresponds to the nth dataset feature, whereas each column $\mathbf{X}_{:,m}$ for $m = 1, \ldots, M$ corresponds to the mth dataset instance.

If the dataset presents a multidimensional structure, the $(R+1)$-way noisy dataset tensor $\mathcal{X} \in \mathbb{R}^{N_1\times\cdots\times N_R\times M}$ can be expressed as

$$\mathcal{X} = \mathcal{X}_0 + \mathcal{N}, \tag{3}$$

where $\mathcal{X}_0 \in \mathbb{R}^{N_1\times\cdots\times N_R\times M}$ is the free-noise dataset tensor and $\mathcal{N} \in \mathbb{R}^{N_1\times\cdots\times N_R\times M}$ is the noise tensor, both obtained by arranging $\mathbf{X}_0$ and $\mathbf{N}$ as tensors of order $R + 1$. Here we consider that each column of a given matrix is reshaped as a tensor with dimensions $N_1 \times \cdots \times N_R$ and stacked along the $(R + 1)$th dimension, generating the corresponding $(R + 1)$-way tensor, with $N = \prod_{r=1}^{R} N_r$.

Finally, the rth unfolding matrix $[\mathcal{X}]_{(r)} \in \mathbb{R}^{N_r\times\prod_{j\neq r} N_j M}$ for $r = 1, \ldots, R$ is given by (Cichocki et al., 2015)

$$[\mathcal{X}]_{(r)} = [\mathcal{X}_0]_{(r)} + [\mathcal{N}]_{(r)}, \tag{4}$$

where $[\mathcal{X}_0]_{(r)} \in \mathbb{R}^{N_r\times\prod_{j\neq r} N_j M}$ and $[\mathcal{N}]_{(r)} \in \mathbb{R}^{N_r\times\prod_{j\neq r} N_j M}$ denote the rth unfolding matrices of $\mathcal{X}_0$ and $\mathcal{N}$, respectively.

3.3. Types of DDoS attacks

DDoS attacks are one of the most important threats in existence today and can be motivated by several reasons, such as financial benefits, ideological beliefs, intellectual challenges or cyberwarfare (Mahjabin et al., 2017). The basic mechanism of a DDoS attack is composed by four different components: an attacker, multiple control masters or handlers, multiple slaves or zombies, and a victim or target (Douligeris and Mitrokotsa, 2004).

First, the attacker creates several control handlers which are used to control the slave machines after installing malicious codes on them. Such network composed by multiple compromised machines or zombies is called botnet (Liu et al., 2009). Next, the attacker sends codes and commands to the control masters which forward them to all their slave machines. Finally, attacks are executed after the attacker's command, where a massive traffic is flooded to the target by millions of compromised machines over the Internet. Usually the attacker uses spoofed IP addresses in order to hide his identity as well as the slaves' identities.

There are different types of DDoS attack classification in the literature. According to Mahjabin et al. (2017), DDoS attacks can be divided into four classes:

- Bandwidth Depletion Attack: an attacker intends to consume all of the network bandwidth of the victim's system by using a protocol exploit attack or an amplification attack. In the former case, vulnerabilities of different network layer's protocols are exploited by an attacker, such as UDP and ICMP flood attacks. In the latter case, large responses generated from small requests are sent to the target, consuming its bandwidth, for instance, in DNS and NTP amplification attacks.
- Resource Depletion Attack: in this type of attack, the victim's resources are exhausted by protocol exploit attacks, such as the well-known TCP SYN flood attack, or by malformed packet attacks. In the latter case, the victim is confused by a packet intentionally deformed by an attacker, for example, in Ping of Death and Teardrop attacks.
- Infrastructure Attack: corresponds to the most dangerous DDoS attack where both bandwidth and resources are depleted by an attacker. Usually such attacks target hierarchical structures such as root DNSs in order to amplify its damage ability.
- Zero-Day Attack: in this type of attack, zero-day vulnerabilities are explored by an attacker, which can result in potential damage to the victim's network.

3.4. NSL-KDD and CICDDoS2019 datasets

In this work, we use subsets of the NSL-KDD and CICDDoS2019 benchmark datasets, both provided by the Canadian Institute for Cybersecurity (CIC) of the University of New Brunswick (UNB). NSL-KDD is a well-known dataset used to design network intrusion detection systems proposed by Tavallaee et al. (2009). Its training and testing instances are contained in two different datasets called KDDTrain+ and KDDTest+, respectively. NSL-KDD has 41 features and different types of network attacks divided into four major categories: probe, denial of service (DoS), user to root (U2R) and remote to local (R2L) (Bamakan et al., 2016). In a probing attack, an attacker gathers information about a network in order to find its vulnerabilities. On the other hand, in DoS attacks, legitimate users are prevented from using a service after a massive attack is launched on the target server. Moreover, in U2R attacks, an attacker with access to a normal user account exploits some vulnerability in order to gain super-user privilege. Finally, in R2L attacks, an attacker tries to gain access to the victim's machine without having an account on it (Gogoi et al., 2012). The DoS attacks present in the NSL-KDD dataset and used in this work are: Neptune, Teardrop, Smurf, Pod, Back, Land, UDPStorm, Apache2, ProcessTable and MailBomb.

Introduced by Sharafaldin et al. (2019), CICDDoS2019 is the most up-to-date DDoS attack dataset available on the web. It is completely labeled and contains more than 80 network traffic features with millions of instances and different DDoS attack types. The authors divided the DDoS attacks present in the CICDDoS2019 dataset into two categories: reflection-based and exploitation-based. In the first category, an attacker, using spoofed IP addresses, sends several request packets to a server which replies directly to the forged IPs, overwhelming the victim's bandwidth or resources. Such attacks become more efficient when traffic amplification is used, i.e., when the response size is much larger than the request size. The reflection-based DDoS attacks used in this work include DNS, LDAP, MSSQL, NetBIOS, NTP, SNMP, SSDP and SSDP based attacks. On the other hand, exploitation-based DDoS attacks usually consume server resources by exploiting protocol vulnerabilities. In this work, we use SYN flood and UDP flood as exploitation-based attacks. Such attacks can also be performed by a reflection structure composed by several compromised machines and spoofed IP addresses, similarly to the reflection-based category.

The DDoS attack types and the corresponding number of instances collected from NSL-KDD and CICDDoS2019 datasets are detailed in Section 5.

4. Proposed tensor based framework for DDoS attack detection

This section introduces the proposed tensor based framework for DDoS attack detection, which is represented by the block diagram illustrated in Fig. 1. Such framework is composed by four major blocks, namely: Data Preprocessing, Dataset Splitting, Dataset Denoising and Machine Learning Supervised Classification. Particularly, in the third block we propose an extension of the recent MuDe technique for noise attenuation. Data preprocessing and dataset splitting are detailed in Sections 4.1 and 4.2, respectively. Next, Sections 4.3 and 4.4 present the proposed extended MuDe technique as well as an overview of machine learning supervised classification, respectively.

4.1. Data preprocessing

Initially, the DDoS attack dataset is sent to a data preprocessing unit, as depicted by block 1 of Fig. 1, where some operations are performed, such as data cleansing, feature scaling and label encoding. Data instances which contain only "Not a Number" (NaN) values as well as zero-valued feature vectors are removed. Additionally, if the mth instance $\mathbf{X}_{:,m}$ contains some element $x_{n,m}$ with a NaN value, such element is replaced by the mean value of the nth feature vector. Such

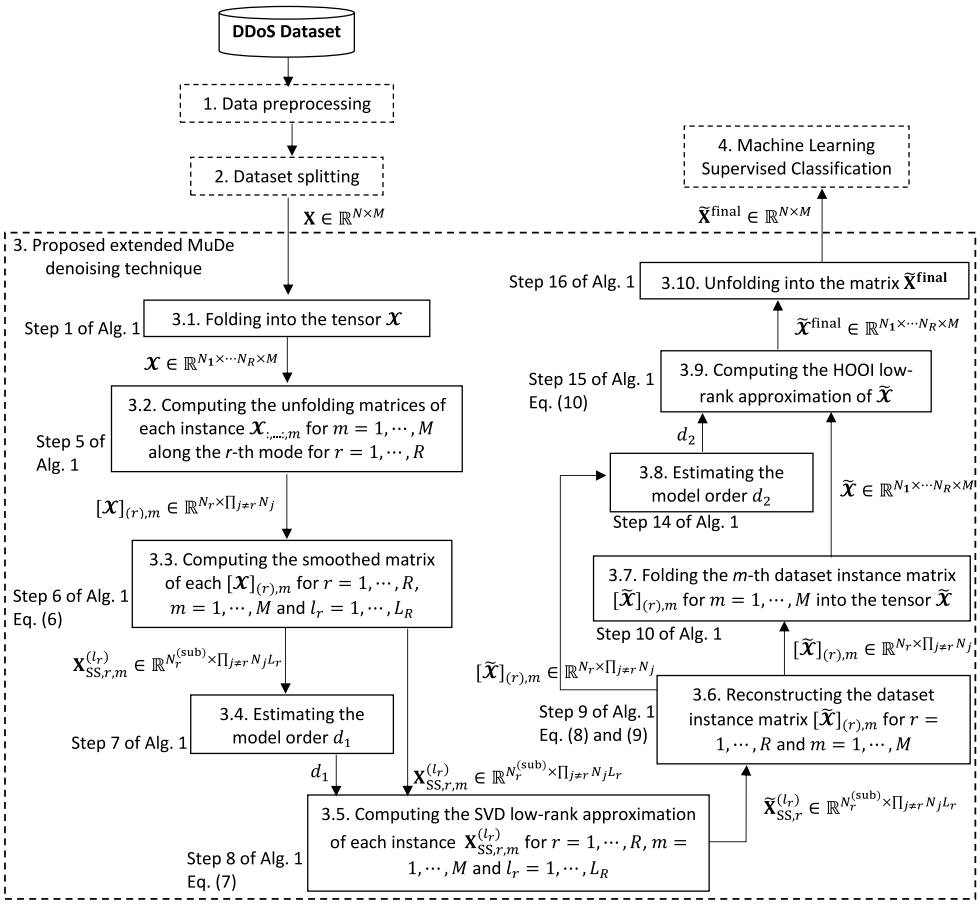


Fig. 1. The proposed tensor based framework for DDoS attack detection. The extended MuDe denoising technique is detailed in Boxes 3.1 to 3.10 with the respective references to steps of Algorithm 1 and equations of Section 4.3.

method, known as "mean imputation", is a common missing data handling approach in the machine learning area such that missing values are eliminated from the dataset while preserving the mean of the corresponding feature values. Moreover, all features are re-scaled to the range $[0, 1]$, in a process called "normalization", given by

$$x_{nm} \leftarrow \frac{x_{nm} - \min(\mathbf{X}_{n,:})}{\max(\mathbf{X}_{n,:}) - \min(\mathbf{X}_{n,:})}, \tag{5}$$

where $\min(\mathbf{X}_{n,:})$ and $\max(\mathbf{X}_{n,:})$ are the minimum and maximum values of the nth dataset feature $\mathbf{X}_{n,:}$, respectively. Such feature scaling is done because a particular feature with higher order of magnitude could dominate other dataset features, generating skewed results.

Finally, since most of the machine learning algorithms can only process numerical variables, class labels must be converted from categorical to numerical values. As our work is developed only for binary classification, legitimate traffic is labeled as "0", while DDoS attacks are labeled as "1".

4.2. Dataset splitting

After the preprocessing step, the dataset is split into training and testing sets, as depicted by block 2 of Fig. 1. In this work, we adopt the k-fold cross validation technique, where data are randomly partitioned into k equally sized samples. In the first iteration, one sample is used for testing data, while the other $k-1$ samples are used as training data. Such process is repeated k times so that each instance is used once for testing and $k-1$ times for training. The final results correspond to the average of the results obtained for each round.

The value of k must be chosen such that both training and testing sets are large enough to be statistically representative of the entire dataset. Empirically, the value of k is chosen as 5 or 10. For larger values of k, the difference in size between the training set and the resampling subsets becomes smaller, which reduces the biases, despite the higher computational cost (Kuhn and Johnson, 2013). Since we are dealing with large datasets, training and testing sets which are statistically representative of the original dataset can be obtained even for smaller values of k. Consequently, throughout this work we adopt $k = 5$ in order to obtain a more computationally efficient framework.

4.3. Proposed extended MuDe technique

After the dataset splitting, the training set is forwarded to the denoising module, as depicted by block 3 of Fig. 1. Similar procedure is adopted for the testing set during the testing phase. For the sake of simplicity, we refer to the dataset matrix simply as $\mathbf{X} \in \mathbb{R}^{N \times M}$, which can be the training or testing set depending on the respective phase.

In the context of machine learning supervised classification, datasets are considered to be noise free because we are not able to make assumptions about their base noise type and level (Sáez et al., 2013). However, several factors such as data source and how the information is collected affect the data quality and, consequently, noise is introduced into the dataset. For example, instances can be incorrectly labeled due to the subjectivity during the labeling process, or dataset features can present corrupted values (Garcia et al., 2013). In this context, our idea is to apply the Multiple Denoising technique directly on each dataset instance such that better classification results can be achieved.

Since MuDe algorithm is a fundamental part of our proposed framework, next we briefly overview some of its main characteristics. The

Multiple Denoising scheme was originally proposed by Gomes et al. in Gomes et al. (2019) in order to denoise measurement data collected by multidimensional sensor arrays by applying successive HOSVD low-rank approximations. Such algorithm achieves a higher noise reduction by using a mean based reconstruction method where the output signals of several subarrays are spatially smoothed and then averaged along the rth spatial dimension. Given the outstanding performance presented by MuDe in Gomes et al. (2019) and Maranhão et al. (2019), such scheme shows a good potential for dataset denoising in order to obtain more accurate attack detection techniques. Nonetheless, MuDe cannot be directly applied to entire datasets because instances with different class labels would be averaged each other along the dataset dimensions, leading to data corruption. Therefore, we include two more contributions on our paper by extending the original MuDe algorithm in two ways: (i) by applying the traditional multiple denoising technique directly on each dataset instance, and (ii) by including a HOOI low-rank approximation based denoising module. In the former case, we intend to attenuate noise from each instance individually. Furthermore, in the latter case, we want to eliminate, from the entire dataset, noise residuals not removed by MuDe scheme. Appendix presents the mathematical concepts about the main schemes used for noise attenuation of matrices and tensors in our proposed extended MuDe technique.

The proposed extended MuDe technique is composed by 10 steps, as depicted by Blocks 3.1 to 3.10 of Fig. 1. Initially, in Box 3.1, the dataset matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ is folded into the $(R+1)$-way tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M}$. Next, Box 3.2 computes the unfolding matrix $[\mathcal{X}]_{(r),m} \in \mathbb{R}^{N_r \times \prod_{j \neq r} N_j}$ of each dataset instance $\mathcal{X}_{:,\ldots,:,m} \in \mathbb{R}^{N_1 \times \cdots \times N_R}$ for $m = 1, \ldots, M$ along the rth mode for $r = 1, \ldots, R$. Then, in Box 3.3, each unfolding matrix $[\mathcal{X}]_{(r),m}$ for $m = 1, \ldots, M$ and $r = 1, \ldots, R$ goes through a process known as smoothing. The rth smoothed matrix $\mathbf{X}_{\mathrm{SS},r,m}^{(L_r)} \in \mathbb{R}^{N_r^{(\mathrm{sub})} \times \prod_{j \neq r} N_j L_r}$ of the mth dataset instance can be expressed as (Thakre et al., 2010)

$$\mathbf{X}_{\mathrm{SS},r,m}^{(L_r)} = [[\mathcal{X}]_{(r),m}^{(1)}, \ldots, [\mathcal{X}]_{(r),m}^{(L_r)}], \tag{6}$$

where $[\mathcal{X}]_{(r),m}^{(l_r)} \in \mathbb{R}^{N_r^{(\mathrm{sub})} \times \prod_{j \neq r} N_j}$ for $l_r = 1, \ldots, L_r$ corresponds to the output of the l_rth subarray for the mth instance in the rth dimension, and $N_r^{(\mathrm{sub})} = N_r - L_r + 1$ is the size of each subarray.

In the following, given $\mathbf{X}_{\mathrm{SS},r,m}^{(L_r)} \in \mathbb{R}^{N_r^{(\mathrm{sub})} \times \prod_{j \neq r} N_j L_r}$, Box 3.4 of Fig. 1 estimates the model order d_1 by using model order selection (MOS) schemes, such as the Minimum Description Length (MDL) (Wax and Kailath, 1985; Barron et al., 1998), Efficient Detection Criterion (EDC) (Zhao et al., 1986), Akaike's Information Theoretic Criteria (AIC) (Wax and Kailath, 1985; Akaike, 1974), Stein's Unbiased Risk Estimator (SURE) (Ulfarsson and Solo, 2008) or RADOI (Radoi and Quinquis, 2004; da Costa et al., 2011). Next, given d_1, the SVD low-rank approximation of the smoothed matrix $\mathbf{X}_{\mathrm{SS},r,m}^{(l_r)}$ is computed in Box 3.5 as follows

$$\tilde{\mathbf{X}}_{\mathrm{SS},r,m}^{(l_r)} = [[\tilde{\mathcal{X}}]_{(r),m}^{(1)}, \ldots, [\tilde{\mathcal{X}}]_{(r),m}^{(l_r)}] = \mathbf{U}_s \mathbf{\Sigma}_s \mathbf{V}_s^{\mathsf{H}}, \tag{7}$$

where the columns of $\mathbf{U}_s \in \mathbb{R}^{N_r \times d_1}$ and $\mathbf{V}_s \in \mathbb{R}^{\prod_{j \neq r} N_j \times d_1}$ correspond to the singular vectors of $\tilde{\mathbf{X}}_{\mathrm{SS},r,m}^{(l_r)}$, whereas the diagonal of $\mathbf{\Sigma}_s \in \mathbb{R}^{d_1 \times d_1}$ contains the singular values of $\tilde{\mathbf{X}}_{\mathrm{SS},r,m}^{(l_r)}$.

In sequence, Box 3.6 of Fig. 1 reconstructs each dataset instance $[\tilde{\mathcal{X}}]_{(r),m} \in \mathbb{R}^{N_r \times \prod_{j \neq r} N_j}$ for $m = 1, \ldots, M$ and $r = 1, \ldots, R$. The multiple denoised unfolding matrices $[\tilde{\mathcal{X}}]_{(r),m}$ as well as their nth row are given by (Gomes et al., 2019)

$$[\tilde{\mathcal{X}}]_{(r),m} = \begin{bmatrix} [\tilde{\mathcal{X}}]_{(r),m}(1,:) \\ [\tilde{\mathcal{X}}]_{(r),m}(2,:) \\ \vdots \\ [\tilde{\mathcal{X}}]_{(r),m}(N_r,:) \end{bmatrix}, \tag{8}$$

$$[\tilde{\mathcal{X}}]_{(r),m}(n,:) = \frac{1}{l} \sum_{i=1}^{l_r} [\tilde{\mathcal{X}}]_{(r),m}^{(i)}(n-i+1,:), \tag{9}$$

where l is the number of times in which $[\tilde{\mathcal{X}}]_{(r),m}^{(i)}(n-i+1,:)$ is a valid output in the l_rth subarray of the rth dimension. After exploiting all possible subarrays $l_r = 1, \ldots, L_r$ in each dimension $r = 1, \ldots, R$, the dataset tensor $\tilde{\mathcal{X}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M}$ is reconstructed by arranging (8) as a tensor of order $R+1$. Next, in Box 3.7 of Fig. 1, each mth dataset instance matrix $[\tilde{\mathcal{X}}]_{(r),m} \in \mathbb{R}^{N_r \times \prod_{j \neq r} N_j}$ for $m = 1, \ldots, M$ and $r = 1, \ldots, R$ is folded back into the tensor form $\tilde{\mathcal{X}}_{:,\ldots,:,m} \in \mathbb{R}^{N_1 \times \cdots \times N_R}$, generating the dataset tensor $\tilde{\mathcal{X}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M}$.

The following step of the proposed extended MuDe technique is to compute the HOOI low-rank approximation (Lathauwer et al., 2000) of the dataset tensor $\tilde{\mathcal{X}}$, which is depicted in Box 3.9 of Fig. 1. Here our main idea is to eliminate noise residuals which eventually were not removed by the successive SVD low-rank approximations performed on the dataset instances. First, we estimate the model order d_2 in Box 3.8 of Fig. 1 by applying a given MOS technique on the dataset instance $[\tilde{\mathcal{X}}]_{(r),m} \in \mathbb{R}^{N_r \times \prod_{j \neq r} N_j}$ for $m = 1, \ldots, M$ and $r = 1, \ldots, R$. Then, the HOOI low-rank approximation of $\tilde{\mathcal{X}}$ is performed in Box 3.9 of Fig. 1 as follows

$$\tilde{\mathcal{X}}^{\mathrm{final}} = \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_{R+1} \mathbf{U}_{R+1}, \tag{10}$$

where $\mathcal{S} \in \mathbb{R}^{I_1 \times \cdots \times I_{R+1}}$ is the truncated core tensor and $\mathbf{U}_r \in \mathbb{R}^{N_r \times I_r}$ for $r = 1, \ldots, R+1$ are the truncated singular matrices. Furthermore, since M is much larger than N_r for $r = 1, \ldots, R$, we consider $I_1 = N_1$, ..., $I_R = N_R$ and $I_{R+1} = d_2$, i.e., only the $(R+1)$th dimension is truncated by HOOI.

Next, according to Box 3.10 of Fig. 1, the denoised dataset tensor $\tilde{\mathcal{X}}^{\mathrm{final}} \in \mathbb{R}^{N_1 \times \cdots \times N_R \times M}$ is unfolded into the matrix form $\tilde{\mathbf{X}}^{\mathrm{final}} \in \mathbb{R}^{N \times M}$ and can be forwarded to machine learning algorithms for classification. The proposed extended MuDe denoising technique is summarized in Algorithm 1.

Algorithm 1: The proposed extended Multiple Denoising (MuDe) algorithm

Input:
- Dataset matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$
- Maximum number of subarrays in each instance dimension: L_r

Output:
- Denoised dataset matrix $\tilde{\mathbf{X}}^{\mathrm{final}} \in \mathbb{R}^{N \times M}$

Algorithm Steps:

Fold the dataset matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ into the tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times \ldots \times N_R \times M}$
for $m = 1$ to M **do**
  for $r = 1$ to R **do**
    for $l_r = 1$ to L_r **do**
      Compute the unfolding matrix $[\mathcal{X}]_{(r),m} \in \mathbb{R}^{N_r \times \prod_{j \neq r} N_j}$ of the mth instance $\mathcal{X}_{:,\ldots,:,m} \in \mathbb{R}^{N_1 \times \ldots \times N_R}$ along the rth mode
      Compute the smoothed matrix $\mathbf{X}_{\mathrm{SS},r,m}^{(l_r)} \in \mathbb{R}^{N_r^{(\mathrm{sub})} \times \prod_{j \neq r} N_j L_r}$ of the unfolding matrix $[\mathcal{X}]_{(r),m} \in \mathbb{R}^{N_r \times \prod_{j \neq r} N_j}$ along the rth mode for the l_rth subarray size as in (6)
      Estimate the model order d_1 by using a MOS scheme
      Compute the SVD low-rank approximation $\tilde{\mathbf{X}}_{\mathrm{SS},r,m}^{(l_r)} \in \mathbb{R}^{N_r^{(\mathrm{sub})} \times \prod_{j \neq r} N_j L_r}$ of the smoothed matrix $\mathbf{X}_{\mathrm{SS},r,m}^{(l_r)} \in \mathbb{R}^{N_r^{(\mathrm{sub})} \times \prod_{j \neq r} N_j L_r}$ along the rth mode for the l_rth subarray size as in (7)
      Reconstruct the instance matrix $[\tilde{\mathcal{X}}]_{(r),m} \in \mathbb{R}^{N_r \times \prod_{j \neq r} N_j}$ as in (8) and (9)
      Fold the instance matrix $[\tilde{\mathcal{X}}]_{(r),m} \in \mathbb{R}^{N_r \times \prod_{j \neq r} N_j}$ into the tensor $\tilde{\mathcal{X}}_{:,\ldots,:,m} \in \mathbb{R}^{N_1 \times \ldots \times N_R}$
    end
  end
end
Estimate the model order d_2 by using a MOS scheme
Compute the HOOI low-rank approximation $\tilde{\mathcal{X}}^{\mathrm{final}} \in \mathbb{R}^{N_1 \times \ldots \times N_R \times M}$ of the tensor $\tilde{\mathcal{X}} \in \mathbb{R}^{N_1 \times \ldots \times N_R \times M}$ as in (10)
Unfold the denoised dataset tensor $\tilde{\mathcal{X}}^{\mathrm{final}} \in \mathbb{R}^{N_1 \times \ldots \times N_R \times M}$ into the matrix $\tilde{\mathbf{X}}^{\mathrm{final}} \in \mathbb{R}^{N \times M}$

Table 1
Confusion matrix.

		Predicted class	
		Positive	Negative
Actual class	Positive	TP	FN
	Negative	FP	TN

4.4. Machine learning supervised classification

The final step of the proposed tensor based framework for DDoS attack detection corresponds to the machine learning supervised classification, which is represented in Block 4 of Fig. 1. In general, classification algorithms are applied on an input dataset in order to build a model that best fits a relationship between data instances and the respective class labels. The key idea is to correctly predict the class labels of previously unknown instances, with a good generalization capability (Tan et al., 2006).

During the training phase, the denoised dataset matrix $\tilde{\mathbf{X}}^{\text{final}}$ originated from Box 3.10 of Fig. 1 is used by a machine learning classification algorithm in Box 4 to build a given classification model. Next, in the testing phase, the trained model is applied on an unseen data instance in order to predict whether it is a legitimate traffic or a DDoS attack. In this work, we adopt the following state-of-the-art classification algorithms: AdaBoost (AB), Linear Discriminant Analysis (LDA), Logistic Regression (LR) and Random Forest (RF). For a more detailed explanation about such classifiers, we refer the reader to Tan et al. (2006) and Gualberto et al. (2020).

5. Simulation results

This section presents the simulation results and is subdivided into seven subsections. First, Section 5.1 presents the evaluation metrics adopted in this work. Next, Section 5.2 details the instances and types of DDoS attacks collected from the CICDDoS2019 and NSL-KDD datasets. Then, Section 5.3 describes the main simulation parameters used in our research. Simulation results obtained by adopting different MOS schemes are shown in Section 5.4. Following, Section 5.5 shows the performance evaluation when considering different signal-to-noise ratios. Next, Section 5.6 presents the experiment results for different training dataset size proportions. Finally, Section 5.7 compares the proposed tensor based framework with related works in the literature.

5.1. Evaluation metrics

This subsection presents details about the evaluation metrics used in simulations. All performance metrics can be extracted from the well-known confusion matrix, which is represented in Table 1. Such matrix allows us to visualize all the possible cases of classification. Based on the values of True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) provided by the confusion matrix, several metrics commonly used for performance evaluation can be computed. Here we consider macro-averaged metrics, i.e., each metric value corresponds to the average between the values obtained for classes 0 and 1.

In this work, we adopt Accuracy (Acc), Detection Rate (DR) and False Alarm Rate (FAR), which are defined as follows:

- Accuracy: the ratio between the correctly predicted instances and the total number of instances,

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \tag{11}$$

- Detection Rate: the ratio between the correctly predicted positive instances and the total number of actual positive instances,

$$\text{DR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{12}$$

- False Alarm Rate: the ratio between the number of negative instances wrongly classified as positives and the total number of actual negative instances,

$$\text{FAR} = \frac{\text{FP}}{\text{TN} + \text{FP}}. \tag{13}$$

It is worth to highlight the difference between accuracy and detection rate. The former tells us how well the framework performs generally and is very useful for balanced datasets, i.e., when the number of instances with classes 0 and 1 are approximately equal. However, accuracy must be analyzed along with other metrics for an efficient performance evaluation. For example, consider an imbalanced dataset where 99% of its instances present label 0. If our framework classifies all instances as 0, we have an accuracy of 99%, i.e, we fail to capture the minority class. On the other hand, detection rate is usually applied on imbalanced datasets, which are very common in network intrusion detection problems where attacks occur far less frequently than normal traffic. Also known as true positive rate, such metric informs how well the framework is able to detect attacks.

5.2. DDoS attack datasets

This subsection describes the number of instances as well as the DDoS attack types applied on numerical simulations. In the experiments of Sections 5.4 to 5.6, we use a subset of the CICDDoS2019 dataset in order to evaluate the performance of the proposed framework for several configuration scenarios. However, since CICDDoS2019 is a novel dataset, to the best of our knowledge there are no other works in the literature which applied it for validation. Therefore, in Section 5.7, we also performed simulations by using a subset of the NSL-KDD benchmark dataset such that the proposed framework could be compared with related works which used the same dataset for performance evaluation.

CICDDoS2019 dataset contains millions of instances with 87 network features per instance. In this work, we use a subset of the CICDDoS2019 composed by 32,000 instances and 64 features plus the class label. The selection process of such 64 from the 87 features is summarized as follows. Nine features were eliminated from the CICDDoS2019 dataset because they presented only zero values and, consequently, would not provide any contribution to the machine learning classification algorithms. Such features are: FwdURGFlags, BwdPSHFlags, BwdURGFlags, FwdAvgBytes_Bulk, FwdAvgPackets_Bulk, FwdAvgBulkRate, BwdAvgBytes_Bulk, BwdAvgPackets_Bulk and BwdAvgBulkRate. Moreover, one feature was wrongly written twice in the dataset (FwdHeaderLength and FwdHeaderLength_1) and, consequently, one of such identical copies was ignored. Furthermore, as we intend to develop a DDoS attack detection system regardless of IP addresses, protocols and date/time information, we neglected four more features: Source IP, Destination IP, Protocol and Timestamp. Finally, we also eliminated the features Unnamed_0, FlowID, SimilarHTTP, Inbound, FlowIATStdDev, FwdIATStdDev, BwdIATStdDev, FwdPUSHFlag and StdDevTimeIdleFlow since they did not provide any useful information for our proposed technique. Table 2 describes all CICDDoS2019 dataset features used in this work. More details about each feature can be found in Canadian Institute for Cybersecurity (2017).

Further, Table 3 details the instances collected from the CICDDoS2019 dataset. Since DDoS attacks are not as frequent as normal traffic, 80% of the dataset is represented by legitimate traffic. All DDoS attack instances are labeled as 1, while legitimate traffic is labeled as 0.

Table 2
CICDDoS2019 dataset features used in this work.

Nr	Feature name	Nr	Feature name	Nr	Feature name
1	Source Port	23	Fwd IAT Max	44	CWE Flag Count
2	Destination Port	24	Fwd IAT Min	45	ECE Flag Count
3	Flow Duration	25	Bwd IAT Total	46	Download/Upload Ratio
4	Total Fwd Packet	26	Bwd IAT Avg	47	Avg Packet Size
5	Total Bwd Packet	27	Bwd IAT Max	48	Avg Fwd Segment Size
6	Total Length Fwd Packet	28	Bwd IAT Min	49	Avg Bwd Segment Size
7	Total Length Bwd Packet	29	Fwd Header Length	50	Subflow Fwd Packets
8	Fwd Packet Length Max	30	Bwd Header Length	51	Subflow Fwd Bytes
9	Fwd Packet Length Min	31	Fwd Packet/s	52	Subflow Bwd Packets
10	Fwd Packet Length Avg	32	Bwd Packet/s	53	Subflow Bwd Bytes
11	Fwd Packet Length Std Dev	33	Packet Length Min	54	Fwd Win Bytes
12	Bwd Packet Length Max	34	Packet Length Max	55	Bwd Win Bytes
13	Bwd Packet Length Min	35	Packet Length Avg	56	Fwd Active Data Packet
14	Bwd Packet Length Avg	36	Packet Length Std Dev	57	Fwd Min Segment Size
15	Bwd Packet Length Std Dev	37	Packet Length Var	58	Avg Time Active Flow
16	Flow Bytes/s	38	FIN Flag Count	59	Std Dev Time Active Flow
17	Flow Packets/s	39	SYN Flag Count	60	Max Time Active Flow
18	Flow IAT Avg	40	RST Flag Count	61	Min Time Active Flow
19	Flow IAT Max	41	PUSH Flag Count	62	Avg Time Idle Flow
20	Flow IAT Min	42	ACK Flag Count	63	Std Dev Time Idle Flow
21	Fwd IAT Total	43	URG Flag Count	64	Min Time Idle Flow
22	Fwd IAT Avg				

Table 3
DDoS attack types used in this work as well as the corresponding number of instances when considering the CICDDoS2019 dataset.

Traffic type	Total
legitimate	32,000
DNS-based DDoS	800
LDAP-based DDoS	800
MSSQL-based DDoS	800
NetBIOS-based DDoS	800
NTP-based DDoS	800
SNMP-based DDoS	800
SSDP-based DDoS	800
UDP flood	800
SYN flood	800
TFTP-based DDoS	800
Total legitimate traffic	**32,000**
Total DDoS attack	**8,000**
Total Nr of instances	**40,000**

Table 4
NSL-KDD features used in this work.

Nr	Feature name	Nr	Feature name
1	duration	19	srv_count
2	src_bytes	20	serror_rate
3	dst_bytes	21	srv_serror_rate
4	land	22	rerror_rate
5	wrong_fragment	23	srv_rerror_rate
6	urgent	24	same_srv_rate
7	hot	25	diff_srv_rate
8	num_failed_logins	26	srv_diff_host_rate
9	logged_in	27	dst_host_count
10	num_compromised	28	dst_host_srv_count
11	root_shell	29	dst_host_same_srv_rate
12	su_attempted	30	dst_host_diff_srv_rate
13	num_root	31	dst_host_same_src_port_rate
14	num_file_creations	32	dst_host_srv_diff_host_rate
15	num_shells	33	dst_host_serror_rate
16	num_access_files	34	dst_host_srv_serror_rate
17	is_host_login	35	dst_host_rerror_rate
18	is_guest_login	36	dst_host_srv_rerror_rate

Table 5
DoS attack types used in this work when considering the NSL-KDD dataset. The training and testing instances were extracted from the KDDTrain+ and KDDTest+ datasets, respectively.

Traffic type	Training set	Testing set	Total
Legitimate	67,343	9710	77,053
Neptune	41,214	4657	45,871
Teardrop	892	12	904
Smurf	2646	665	3311
Pod	201	41	242
Back	956	359	1315
Land	18	7	25
UDPStorm	0	2	2
Apache2	0	737	737
ProcessTable	0	685	685
MailBomb	0	293	293
Total legitimate traffic	**67,343**	**9710**	**77,053**
Total DDoS attack	**45,927**	**7458**	**53,385**
Total Nr of instances	**113,270**	**17,168**	**130,438**

In addition, we apply the 5-fold cross validation technique for dataset splitting, i.e., at each round, 80% of the dataset is used for training and 20% for testing. Consequently, each data instance is used once for testing and four times for training.

As previously stated, the NSL-KDD dataset is applied only for comparison between our proposed approach and related works which used the same data in their experiments. Next we detail the NSL-KDD features as well as the types of DDoS attacks used in this work. Here we ignored five of its 41 features: num_outbound_cmds, count, protocol_type, service and flag. The first and second one were neglected because they presented only zero values, while the other three features were deleted because they were all nominal. Table 4 illustrates the NSL-KDD features used in this work, whereas Table 5 details the types of DDoS attacks and their respective number of instances present in the NSL-KDD subset. Similarly to CICDDoS2019, all legitimate and DDoS attack instances are labeled as 0 and 1, respectively.

5.3. Simulation parameters

This subsection presents details about the main parameters adopted in our simulations. All experiments were executed on a desktop computer with processor Intel Core i7-2600 3.40 GHz and 16 GB of RAM. MATLAB R2018a software is used to simulate data preprocessing and tensor decompositions, whereas machine learning classifier algorithms are implemented in the Python Scikit-Learn library. In this work, we evaluate the classifiers AdaBoost (AB), Linear Discriminant Analysis (LDA), Logistic Regression (LR) and Random Forest (RF).

As described in (1), the dataset matrix is given by $\mathbf{X} \in \mathbb{R}^{N \times M}$, where N is the number of features and M is the number of dataset instances. For CICDDoS2019 dataset, all of the $N = 64$ features described in

Table 2 were considered. Moreover, 40,000 instances were collected from the CICDDoS2019 dataset, as shown in Table 3, such that $M = 32{,}000$ training instances and $M = 8000$ testing instances are taken in each iteration of the 5-fold cross validation, corresponding to 80% and 20% of the total collected examples. In addition, in order to simplify the computations performed in tensor decompositions, we consider a three dimensional dataset for all experiments. In other words, the dataset matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ is folded into a 3-way tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times M}$ as in (3), where N_1 and N_2 correspond to the number of features along the 1-st and 2-nd dimensions, respectively. Furthermore, we assume $N_1 = N_2 = 8$ for simplicity.

Moreover, a similar procedure is adopted when the NSL-KDD dataset is applied. All of the $N = 36$ features described in Table 5 are used in simulations. Nonetheless, since NSL-KDD is composed by separate training and testing sets, there is no need for dataset splitting. Thus, $M = 113{,}270$ training instances and $M = 17{,}168$ testing instances were directly collected from the corresponding datasets, as shown in Table 5. Once again we consider a three-dimensional dataset tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times M}$, where $N_1 = N_2 = 6$.

Additionally, for all experiments, we adopt the subarray lengths $L_1 = L_2 = 2$ in the MuDe computations described in (6). The traditional Multiple Denoising technique presents a higher noise reduction ability, which demands higher processing times due to its L_r for $r = 1, \ldots, R$ truncated SVD computations (Gomes et al., 2019). Therefore, L_r must be chosen such that a good trade-off between denoising capability and computational cost is achieved, which is well accomplished when $L_1 = L_2 = 2$.

Finally, since the initial noise type and level present in DDoS attack datasets are unknown, they are assumed to be noise free (Sáez et al., 2013). Consequently, in all experiments we pre-define signal-to-noise ratio (SNR) values by adding a zero-mean white gaussian noise into the dataset features in a supervised manner such that the framework performance can be assessed for different noise levels. The SNR range considered in this work is between −5 dB and 15 dB.

5.4. Performance evaluation for different model order selection schemes

In this subsection, we present the performance evaluation of our proposed framework for different MOS schemes, namely, AIC (Wax and Kailath, 1985; Akaike, 1974), EDC (Zhao et al., 1986), MDL (Wax and Kailath, 1985; Barron et al., 1998), RADOI (Radoi and Quinquis, 2004) and SURE (Ulfarsson and Solo, 2008). In all simulated scenarios, we consider SNR = 30 dB and a three-dimensional tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times M}$, with $N_1 = N_2 = 64$ and $M = 40{,}000$, where the training and testing datasets are generated by using the 5-fold cross validation. Moreover, the subarray lengths $L_1 = L_2 = 2$ are considered in MuDe computations. Table 6 presents the experiment results for AB, LDA, LR and RF classification algorithms. From the values observed in such table, it is clear that all MOS schemes present higher accuracy when random forest algorithm is applied for classification. Such performance may be explained due to the fact that random forests are composed by multiple decision trees combined together which can handle large datasets with high dimensionality, obtaining more stable and robust predictions. On the other hand, for some MOS schemes, AdaBoost outperforms RF in terms of detection rate and false alarm rate. The AdaBoost algorithm is a strong classifier generated from a combination of several weak learners and, depending on the dataset as well as some adopted parameters, can yield more accurate results when compared to random forests.

Since detection rate and false alarm rate are one of the most important metrics used for performance evaluation of network attack detection models, we select the MOS scheme which provides the best results when considering those metrics. Therefore, the best MOS techniques along with the respective machine learning classifiers are as follows: (i) AdaBoost: MDL; (ii) Linear Discriminant Analysis: EDC; (iii) Logistic Regression: SURE; and (iv) Random Forest: EDC. Furthermore, from the results shown in Table 6, we observe that, for AIC and MDL, the proposed framework presents similar performance for some machine learning algorithms. Such MOS schemes are eigenvalue based techniques where the information criterion is a function of the geometric and arithmetic means of the k smallest eigenvalues of the covariance matrix computed from the corresponding dataset matrix. Since the basic difference between such techniques is a penalty function used to account for potential overfitting, they estimate similar model orders and, consequently, our proposed technique shows slightly different results. In addition, despite RADOI is based on empirical functions which use some remarks on the behavior of the autocorrelation matrix eigenvalue (Radoi and Quinquis, 2004), the proposed framework presents results similar to the ones shown by AIC and MDL schemes. On the other hand, our proposed approach showed slightly worse detection rate and false alarm rate for some classifiers when SURE was applied because such scheme requires that a certain percentage of the smallest eigenvalues is only composed of noise, which is not necessarily guaranteed. Hence, accordingly to the findings shown in Table 6 as well as the discussion presented in this paragraph, throughout this work we use MDL, EDC and SURE as the MOS schemes along with the respective machine learning classifier where our proposed framework presented the best performance.

5.5. Performance evaluation for different signal-to-noise ratios

This subsection presents the performance evaluation of the proposed framework against three state-of-the-art low-rank approximation techniques, namely, HOOI (Lathauwer et al., 2000), HOSVD (Haardt et al., 2008; Rajwade et al., 2013) and SVD (Guo et al., 2016; Jha and Yadava, 2011). We consider a 3-way dataset tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times M}$, where $N_1 = N_2 = 64$, $M = 32{,}000$ for the training set and $M = 8000$ for the testing set, with both sets generated according to the 5-fold cross validation. Additionally, in all simulated scenarios, we adopt the subarray lengths $L_1 = L_2 = 2$ for MuDe computations and SNR ranging from −5 dB to 15 dB. Nonetheless, since SVD is a matrix based denoising technique, in this case we consider the dataset in its matrix form, $\mathbf{X} \in \mathbb{R}^{N \times M}$, with $N = 64$ and M given by the same values of the multidimensional case. Fig. 2 shows the SNR of $\mathcal{X}^{\text{final}}$ (SNR_{out}) versus the SNR of $\mathcal{X}$ (SNR_{in}) for the proposed framework as well as the SVD, HOSVD and HOOI techniques. Moreover, we also include simulation results for raw data, where no denoising scheme is applied to the original dataset $\mathcal{X}$. In other words, such graphic demonstrates the SNR gain after a given denoising technique is applied to the dataset $\mathcal{X}$. If $\mathcal{X}_0$ and $\mathcal{X}^{\text{final}}$ are the free-noise and the denoised dataset tensors, respectively, the SNR_{out} is defined as

$$\text{SNR}_{\text{out}} = 10 \cdot \log_{10}\left(\frac{\|\mathcal{X}^{\text{final}}\|_F^2}{\|\mathcal{X}^{\text{final}} - \mathcal{X}_0\|_F^2}\right) (\text{dB}), \tag{14}$$

where we consider a controlled scenario such that the free-noise dataset tensor $\mathcal{X}_0$ in (3) is available.

By observing the results shown in Fig. 2, we can see that $\text{SNR}_{\text{out}} \approx \text{SNR}_{\text{in}}$ for raw data because no denoising technique is applied to the original dataset. Furthermore, note that SVD outperforms raw data since the dataset matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ is truncated to the signal subspace. Moreover, HOOI and HOSVD naturally give better results than SVD because both schemes perform higher order SVDs along the data dimensions, exploiting the tensor structure of the dataset $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times M}$. At last, from Fig. 2 it is clear that the proposed framework delivers the best SNR_{out} among all compared techniques, showing a SNR gain up to 5 dB when compared to raw data. Such results provide evidences that our proposed approach outperforms its counterpart techniques due to the multiple denoising sequentially applied along the R dimensions of each dataset instance.

Next, the accuracy, detection rate and false alarm rate as a function of the signal-to-noise ratio (SNR) are assessed in Fig. 3 for the proposed

Table 6
Performance evaluation of the proposed framework when considering different MOS techniques.

MOS	Accuracy				Detection rate				False alarm rate			
	AB	LDA	LR	RF	AB	LDA	LR	RF	AB	LDA	LR	RF
AIC (Wax and Kailath, 1985; Akaike, 1974)	0.9944	0.9616	0.9852	0.9966	0.9926	0.9275	0.9695	0.9932	0.0103	0.1287	0.0565	0.0125
EDC (Zhao et al., 1986)	0.9903	0.9645	0.9849	0.9966	0.9837	0.9381	0.9688	0.9934	0.0271	0.1057	0.0576	0.0122
MDL (Wax and Kailath, 1985; Barron et al., 1998)	0.9953	0.9620	0.9852	0.9964	0.9935	0.9273	0.9695	0.9924	0.0094	0.1299	0.0564	0.0142
RADOI (Radoi and Quinquis, 2004)	0.9937	0.9614	0.9849	0.9968	0.9910	0.9262	0.9688	0.9933	0.0135	0.1320	0.0578	0.0217
SURE (Ulfarsson and Solo, 2008)	0.9945	0.9274	0.9852	0.9949	0.9918	0.8846	0.9698	0.9887	0.0128	0.1860	0.0558	0.0217

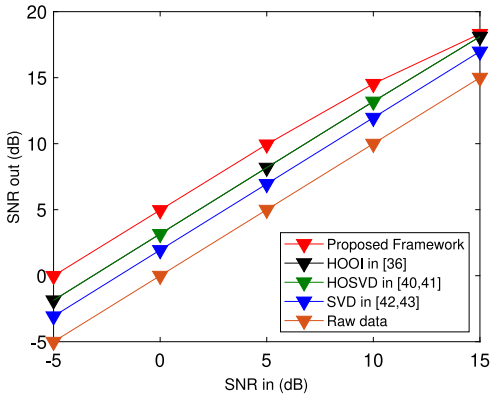


Fig. 2. SNR_{out} versus SNR_{in} for a third-order dataset tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times M}$, with $N_1 = N_2 = 64$, $M = 32{,}000$ for the training set and $M = 8000$ for the testing set.

framework as well as the SVD, HOSVD and HOOI methods when considering AB, LDA, LR and RF classifiers. By observing the results shown in Fig. 3, it is clear that all techniques show better performance as the SNR is higher. Since data corruption in datasets can degrade the performance of machine learning classification algorithms, network attack detectors usually present better results for low noise levels. Furthermore, we observe that the proposed framework delivers better detection rate and false alarm rate when compared to its competing schemes for almost all SNR range, regardless of the classifier, especially for low SNR values. In such cases, the benefits of our proposed extended MuDe algorithm are more evident because it provides a higher noise reduction due to the multiple denoising performed along the rth dimension of the mth dataset instance for $r = 1, \ldots, R$ and $m = 1, \ldots, M$. On the other hand, the accuracy of our proposed approach is outperformed by its competitor methods for AdaBoost and Logistic Regression classifiers when the SNR is low. In this situation, despite the superior performance in terms of detection rate and false alarm rate, the proposed framework achieves a higher number of false negatives, i.e., true attack traffic is wrongly classified as legitimate. In addition, note that the competing techniques present slightly different performances, regardless of the classifier, for all SNR range. Since the third dimension (corresponding to the number of instances M) of the dataset tensor $\mathcal{X}$ is much larger than the first and second dimensions (corresponding to the number of features N_1 and N_2, respectively), the classic multidimensional techniques, HOSVD and HOOI, do not provide significant gain when compared to the matrix based scheme, SVD.

Therefore, the results shown in Fig. 3 highlight the higher robustness of our proposed framework against data corruptions caused by, for instance, uncalibrated measures or transcription errors. The multiple denoising introduced by our approach along different modes of the dataset instances as well as between instances (through HOOI) provides a better classification performance when compared to the state-of-the-art low-rank approximation techniques.

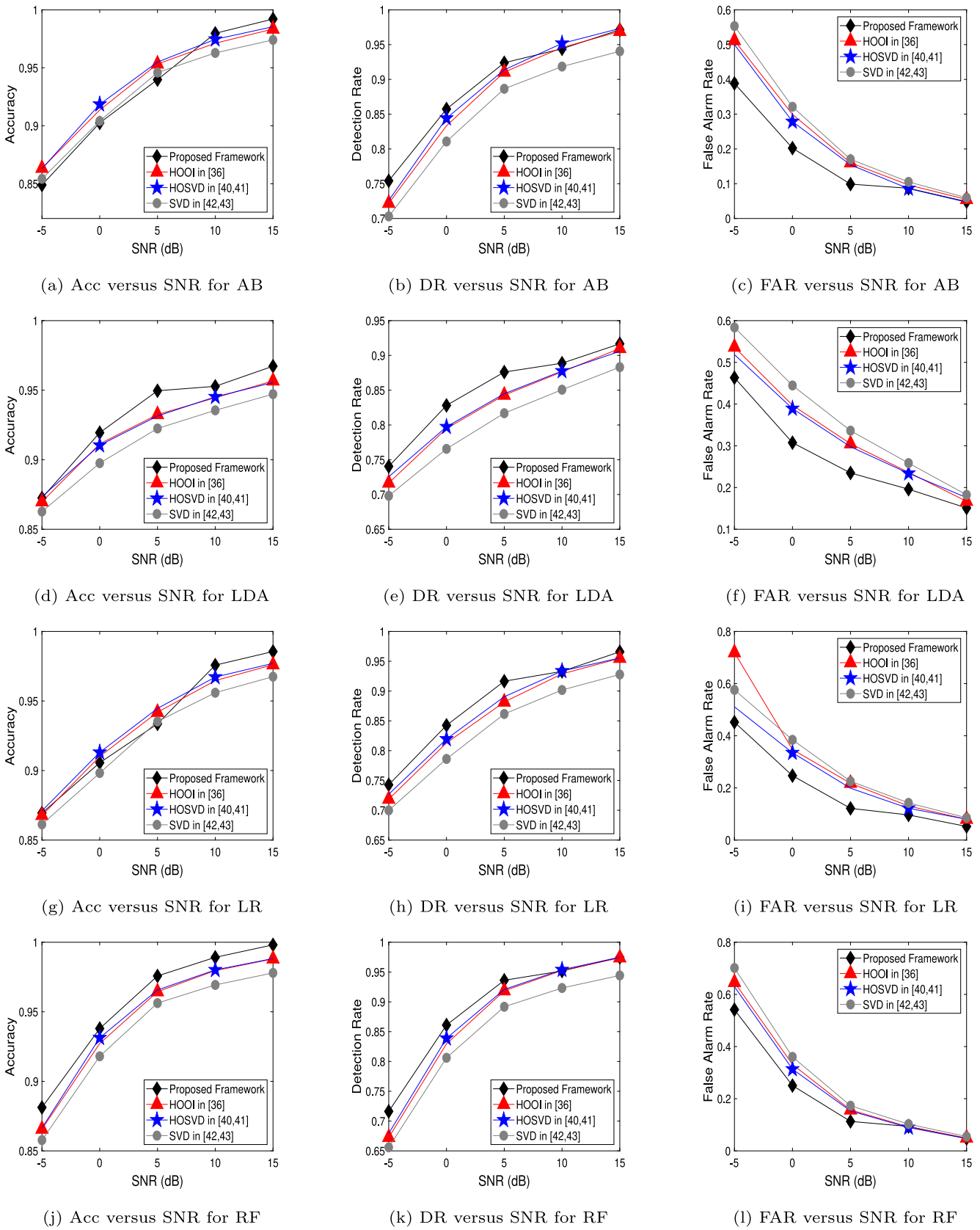
5.6. Performance evaluation for different training dataset size proportions

In this subsection, the proposed framework is assessed against the SVD, HOSVD and HOOI low-rank approximation techniques for different training dataset size proportions. The dataset is split into training and testing sets, where the proportion of the training data ranges from 20% to 70% of the original dataset. Once again we consider a three-dimensional dataset tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times M}$, with $N_1 = N_2 = 8$, $M = 32{,}000$ for the training set and $M = 8000$ for the testing set. The subarray lengths for MuDe are given by $L_1 = L_2 = 2$. Additionally, all experiments are performed in high noise level conditions, with SNR fixed in 0 dB. Similarly to the performance evaluation in Section 5.5, we consider a dataset matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ for SVD technique, with $N = 64$. Fig. 4 shows the accuracy, detection rate and false alarm rate as a function of the training size proportion (TSP) when considering AB, LDA, LR and RF classification algorithms. From the results shown in Fig. 4, we note that all compared techniques present better performance as the training dataset size proportion grows. This fact is observed due to the fact that machine learning classification algorithms need more training samples when the dataset has a large number of features. In this case, the classification model is better fit to the training data after adjusting several parameters such that the errors between the actual and predicted classes are minimal, despite the higher risk of overfitting. Furthermore, it can be seen that our proposed framework outperforms the other competitor methods for almost all training size proportion range. Particularly for LDA classifier, the proposed approach is very robust to the variation of the training dataset size, keeping a regular performance along all range. This can be explained because LDA projects high-dimensional data onto a lower dimensional space such that the separation of instances from different classes is maximized, while the dispersion of instances from the same class is minimized (Ji and Ye, 2008). Since our proposed approach performs multiple denoising along the instance dimensions (within-instances) as well as along the entire dataset through HOOI (between-instances), the data projected by LDA technique present a larger separability between classes and, consequently, better classification results are obtained, regardless of the training dataset size. Moreover, differently from the results shown in Section 5.5, the difference in performance between SVD and its multidimensional counterparts, HOSVD and HOOI, is more noticeable in Fig. 4. Consequently, we conclude that matrix based denoising techniques are more sensitive to the variation of training dataset size. On the other hand, tensor based schemes perform a more efficient noise attenuation through multiple SVDs along the dataset dimensions, which partially compensates the lack of training data in machine learning classifiers. Finally, in line with the findings from Section 5.5, note that all compared techniques are not so efficient in terms of network attack detection because we consider a high noise level scenario, with signal-to-noise ratio fixed in 0 dB. Our intent is to simulate an environment where dataset is highly corrupted and, consequently, the performance of the machine learning classification algorithm is degraded.

Hence, from the results shown in Fig. 4 as well as the discussion presented in the previous paragraph, we conclude that our proposed framework is very robust against the variation of the training dataset size proportion, which once again highlights the benefits of the multidimensional denoising introduced on the dataset by our proposed extended MuDe algorithm.

5.7. Performance comparison with related works

This subsection presents the experiment results obtained from the



(a) Acc versus SNR for AB (b) DR versus SNR for AB (c) FAR versus SNR for AB

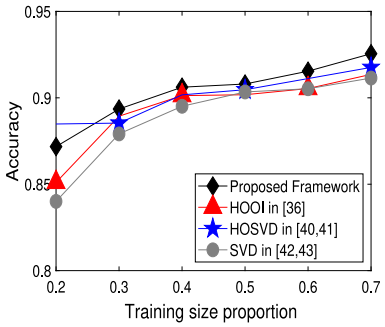
(d) Acc versus SNR for LDA (e) DR versus SNR for LDA (f) FAR versus SNR for LDA

(g) Acc versus SNR for LR (h) DR versus SNR for LR (i) FAR versus SNR for LR

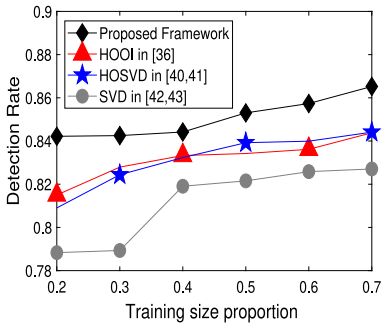
(j) Acc versus SNR for RF (k) DR versus SNR for RF (l) FAR versus SNR for RF

Fig. 3. Plots of accuracy, detection rate and false alarm rate as a function of the signal-to-noise ratio (dB) when considering different classification algorithms.

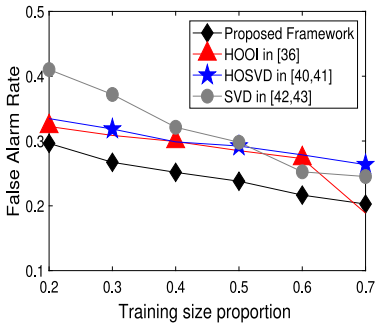
comparison between our proposed approach and related works. As stated in Section 5.2, CICDDoS2019 is a novel dataset and, consequently, we could not find other works in the literature which applied such dataset for validation. Therefore, we also performed experiments on a subset of the traditional NSL-KDD dataset (Tavallaee et al., 2009) such that we could compare our proposed framework with works that



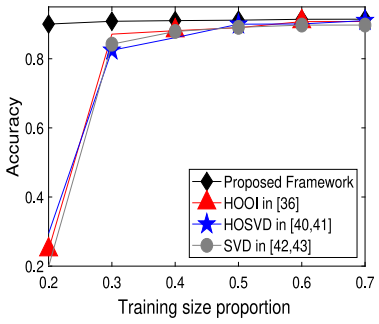
(a) Acc vs TSP for AB



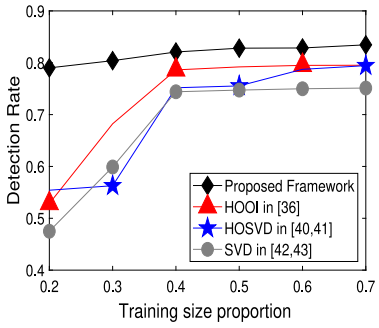
(b) DR vs TSP for AB



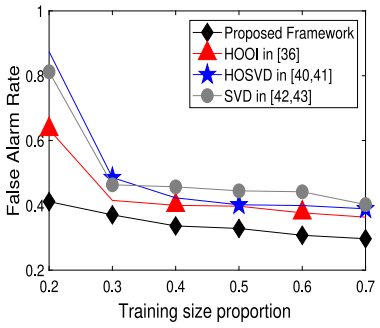
(c) FAR vs TSP for AB



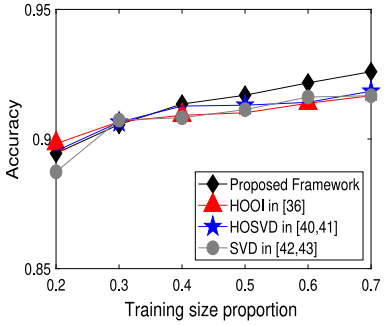
(d) Acc versus TSP for LDA



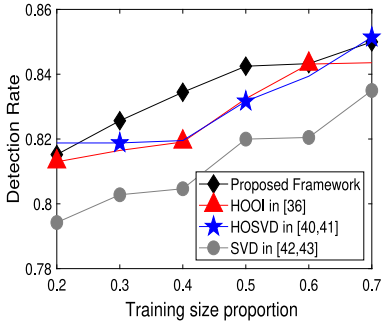
(e) DR versus TSP for LDA



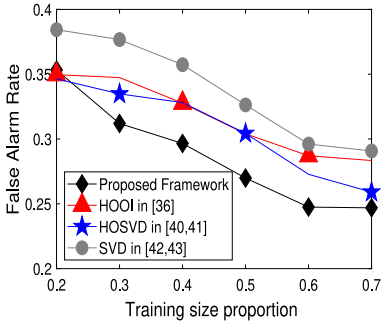
(f) FAR versus TSP for LDA



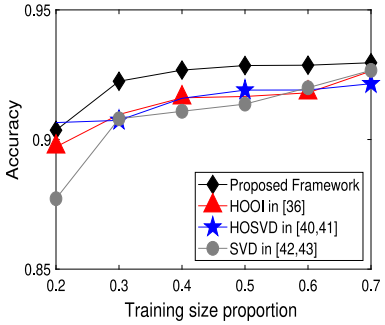
(g) Acc versus TSP for LR



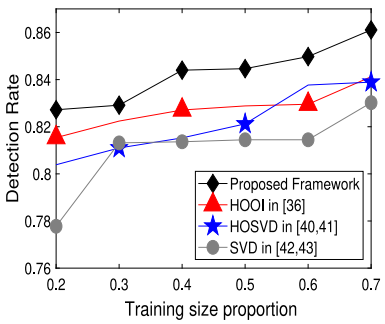
(h) DR versus TSP for LR



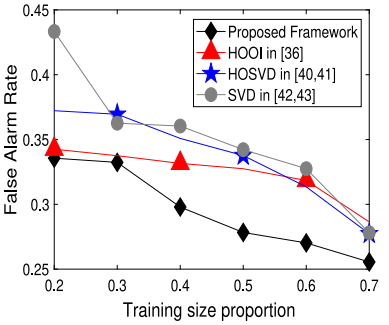
(i) FAR versus TSP for LR



(j) Acc versus TSP for RF



(k) DR versus TSP for RF



(l) FAR versus TSP for RF

Fig. 4. Plots of accuracy, detection rate and false alarm rate as a function of the training dataset size proportion when considering different classification algorithms.

used the same dataset for performance evaluation. In simulations, we consider a three-dimensional dataset $\mathcal{X} \in \mathbb{R}^{M_1 \times M_2 \times N}$, with $N_1 = N_2 = 6$ features, $M = 113,270$ training instances and $M = 17{,}168$ testing instances, with subarray lengths $L_1 = L_2 = 2$. Furthermore, since the related works consider noise-free datasets, we simulate a scenario with a very low noise level by fixing the SNR in 40 dB.

Table 7
Performance evaluation of the proposed techniques and related works through experiments using a subset of the NSL-KDD dataset for DDoS attack detection.

Work	DM	NF	Acc (%)	DR (%)	FAR (%)
Proposed framework	AB	36	98.40	98.63	0.72
	LDA	36	98.54	98.58	1.08
	LR	36	98.70	99.02	0.06
	RF	36	98.78	99.06	0.14
Wang et al. (2020)	SBS-MLP	31	97.66	94.88	0.62
	SFS-MLP	35	97.61	94.71	0.60
	CTSBS-MLP	41	97.61	94.78	0.63
Kushwah and Ali (2017)	MLP	41	96.30	97.91	5.00
Gogoi et al. (2012)	TUIDS	41	96.55	98.88	1.12
Hosseini and Azizi (2019)	NB	12	93.10	N/A	N/A
	DT	10	98.20	N/A	N/A
	MLP	20	96.10	N/A	N/A
	*k*NN	11	97.70	N/A	N/A
Yusof et al. (2017)	ELM	17	91.70	N/A	N/A

Table 7 shows the detection model (DM), the number of dataset features (NF) as well as the values of accuracy, detection rate and false alarm rate obtained by our proposed approach and the related works. Some metrics are represented as "Not Available" (N/A) because the corresponding works did not present such values. It can be observed that our proposed framework outperforms all related works in terms of accuracy, regardless of the DM. In addition, despite the proposed approach does not outperform some compared works in terms of detection rate and false alarm rate, it still presents an outstanding DDoS attack detection performance. For instance, despite Wang et al. (2020) presented a lower false alarm rate when compared to the proposed scheme with AB and LDA, our approach is far superior is terms of detection rate. The method proposed by Wang et al. reconstructs the detection model by dynamically perceiving the occurrence of detection errors through a feedback mechanism, which can explain its better performance on false alarm rates. Nonetheless, the best detection rate of Wang et al. was 94.88%, whereas this current proposal obtained 99.06% with RF classifier. Another example is the work of Gogoi et al. which provided a detailed analysis of the NSL-KDD dataset and proposed two real life network intrusion datasets. The method shown by such authors outperforms the proposed framework with AB and LDA classifiers in terms of detection rate, but our approach shows lower false alarm rate, regardless of the classification algorithm. While the worst FAR presented by the proposed framework was 1.08% with LDA classifier, Gogoi et al. showed a false alarm rate of 1.12%.

Therefore, from the results shown in Table 7, it is evident that the proposed tensor based framework for DDoS attack detection delivers significantly better results when compared to similar works existing in the literature, in accordance with the findings reported in Sections 5.5 and 5.6.

6. Computational complexity

This section discusses the computational complexity of the proposed extended MuDe technique. Here we do not consider the computational cost related to folding and unfolding of matrices and tensors since such functions are about data representations. For simplicity, the computational complexity is analyzed for a three-dimensional dataset tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times M}$. We provide an analysis of the asymptotic time cost as a function of the largest contributions of the most important variables, namely, N_1, N_2, M and d, where d corresponds to the model order estimated by a given MOS technique.

First, we show the computational complexity of the traditional MuDe technique when applied to M dataset instances. The time cost of the SVD low-rank approximation of a matrix with dimensions $(N_r - l_r + 1) \times (\prod_{j \neq r} N_j l_r)$ truncated to d is given by $\mathcal{O}[(\prod_{j \neq r} N_j)(N_r - l_r + 1)dl_r]$ (Gomes et al., 2019). Since MuDe computes a total of L_r truncated SVDs for each dimension r, its overall computational complexity for all M dataset instances is given by

$$\mathcal{O}[\text{MuDe}] = \mathcal{O}\left[M \sum_{r=1}^{2} \sum_{l_r=1}^{L_r} N_j(N_r - l_r + 1)dl_r\right], j \neq r. \quad (15)$$

Next, according to Ballester-Ripoll et al. (2015), the overall computational complexity of HOOI can be expressed as

$$\mathcal{O}[\text{HOOI}] = \mathcal{O}(N_{\max}^3 dJ) + \mathcal{O}(N_{\max}^2 d^2 J) + \mathcal{O}(N_{\max}^3 d) + \mathcal{O}(N_{\max} d^3), \quad (16)$$

where $N_{\max} = \max\{N_1, N_2, M\}$ and J is the number of iterations of the HOOI algorithm.

Finally, the overall computational complexity of the proposed extended MuDe technique corresponds to the sum of the above mentioned complexities, i.e.,

$$\mathcal{O}[\text{Framework}] = \mathcal{O}[\text{MuDe}] + \mathcal{O}[\text{HOOI}]. \quad (17)$$

7. Conclusion and future works

In this work, we propose a novel framework for DDoS attack detection which exploits both multidimensional signal processing techniques and machine learning based classification algorithms. The proposed framework is composed by four main blocks: (i) data preprocessing, (ii) dataset splitting, (iii) dataset denoising, and (iv) machine learning supervised classification. Particularly considering the third block, we propose an extension of the recent MuDe technique, which was originally applied for denoising of measurement data collected by multidimensional sensors arrays.

The proposed framework was validated through comparison with state-of-the-art low-rank approximation techniques, such as SVD, HOSVD and HOOI, by using the CICDDoS2019 benchmark dataset. Nonetheless, since CICDDoS2019 is a very recent dataset, as far as we know there are no other works in the literature that applied such dataset. Consequently, we used the well-known NSL-KDD dataset in order to compare the proposed technique with related works which used the same dataset in their experiments. According to the obtained results, our proposed approach considerably outperforms the competing techniques, showing outstanding values of accuracy, detection rate and false alarm rate. Therefore, we concluded that the proposed extended MuDe technique provides a great performance gain on DDoS attack detection.

As a future work we intend to evaluate the performance of the proposed framework by using different state-of-the-art tensor decomposition techniques such as CANDECOMP/PARAFAC, or by detecting different types of network attack. Additionally, one also could apply deep learning based classification algorithms, such as convolutional neural networks, on the proposed approach.

CRediT authorship contribution statement

João Paulo A. Maranhão: Conceptualization, Methodology, Formal analysis, Software, Validation, Writing - original draft. **João Paulo C.L. da Costa:** Writing - review & editing, Project administration, Supervision. **Elnaz Javidi:** Resources, Data curation. **César A. Borges de Andrade:** Resources, Software. **Rafael T. de Sousa Jr.:** Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the CNPq - Brazilian National Research Council [Grants 312180/2019-5 PQ-2, BRICS2017-591 LargEWiN and 465741/2014-2 INCT on Cybersecurity]; the CAPES - Brazilian Higher Education Personnel Improvement Coordination [Grants 23038.007604/2014-69 FORTE and 88887.144009/2017-00 PROBRAL]; the FAP-DF - Brazilian Federal District Research Support Foundation [Grants 0193.001366/2016 UIoT and 0193.001365/2016 SSDDC]; the Brazilian Ministry of the Economy [Grants 005/2016 DIPLA and 083/2016 ENAP]; the Institutional Security Office of the Presidency of Brazil [Grant ABIN 002/2017]; the Administrative Council for Economic Defense [Grant CADE 08700.000047/2019-14]; the General Attorney of the Union [Grant AGU 697.935/2019]; the DPI/DPG/UnB - Decanates of Research and Innovation and Post-Graduate Studies of the University of Brasilia; the Post-Graduate Program on Electrical Engineering of the University of Brasilia (PPGEE/UnB), with resources from CAPES/PROAP; and the Ministry of Defence/Brazilian Army (MD/EB), through the Department of Science and Technology/Systems Development Center (DCT/CDS).

Appendix. Noise attenuation in matrices and tensors: Mathematical concepts

In this appendix, we present the mathematical concepts regarding the main schemes used for noise attenuation of matrices and tensors in our proposed extended MuDe technique, particularly the SVD and HOOI low-rank approximations.

The SVD low-rank approximation is widely applied for noise reduction in data matrices. In order to show how denoising can be performed by Singular Value Decomposition, let us first present the basic concepts about covariance matrix and eigenvalue decomposition. The data matrix can be modeled as $\mathbf{X} = \mathbf{X}_0 + \mathbf{N}$, where $\mathbf{X}_0 \in \mathbb{R}^{N\times M}$ is the noise-free matrix and $\mathbf{N} \in \mathbb{R}^{N\times M}$ is the noise samples matrix. Assuming that the noise and the signal are independent and zero-mean, the covariance matrix $\mathbf{R}_{\text{xx}} \in \mathbb{R}^{N\times N}$ of $\mathbf{X}$ is given by (Haardt et al., 2014)

$$\begin{aligned}\mathbf{R}_{\text{xx}} &= E\{\mathbf{X}\mathbf{X}^{\text{H}}\} \\ &= E\{\mathbf{X}_0\mathbf{X}_0^{\text{H}}\} + E\{\mathbf{N}\mathbf{N}^{\mathbf{H}}\} \\ &= \mathbf{R}_{\text{ss}} + \mathbf{R}_{\text{nn}},\end{aligned} \tag{A.1}$$

where $E\{\cdot\}$ denotes the expectation function. Moreover, $\mathbf{R}_{\text{ss}} = E\{\mathbf{X}_0\mathbf{X}_0^{\text{H}}\} \in \mathbb{R}^{N\times N}$ is the signal covariance matrix and $\mathbf{R}_{\text{nn}} = E\{\mathbf{N}\mathbf{N}^{\mathbf{H}}\} = \sigma^2\mathbf{I}_N \in \mathbb{R}^{N\times N}$ denotes the noise covariance matrix, where σ^2 is the noise power and $\mathbf{I}_N \in \mathbb{R}^{N\times N}$ is the identity matrix.

If the rank $d < \min\{M, N\}$ of the dataset matrix $\mathbf{X}$ is known, the eigenvalue decomposition of $\mathbf{R}_{\text{xx}}$ in (A.1) can be expressed as

$$\begin{aligned}\mathbf{R}_{\text{xx}} &= \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^{\text{H}} + \mathbf{U}_n\mathbf{\Lambda}_n\mathbf{U}_n^{\text{H}} \\ &= \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^{\text{H}} + \sigma^2\mathbf{U}_n\mathbf{U}_n^{\text{H}},\end{aligned} \tag{A.2}$$

where $\mathbf{\Lambda}_s = \text{diag}\{\lambda_1, \dots, \lambda_d\} \in \mathbb{R}^{d\times d}$ and $\mathbf{\Lambda}_n = \text{diag}\{\sigma^2, \dots, \sigma^2\} = \sigma^2\mathbf{I}_{N-d} \in \mathbb{R}^{(N-d)\times(N-d)}$ are matrices containing the signal and the noise eigenvalues in their diagonals, respectively. Furthermore, since the columns of $\mathbf{U}_s$ and $\mathbf{U}_n$ contain the signal and the noise eigenvectors of $\mathbf{R}_{\text{xx}}$, they are known as signal subspace matrix and noise subspace matrix, respectively. The value of d can be computed by state-of-the-art model order selection techniques, such as AIC (Wax and Kailath, 1985; Akaike, 1974), EDC (Zhao et al., 1986) and MDL (Wax and Kailath, 1985; Barron et al., 1998).

Instead of computing the eigenvalue decomposition in (A.2), we can perform the SVD of the dataset matrix $\mathbf{X}$ and, consequently, the signal subspace $\mathbf{U}_s$ can be obtained from the d dominant left singular vectors (Haardt et al., 2014). In this sense, SVD performs the noise attenuation of the dataset matrix $\mathbf{X}$ by truncating the matrices $\mathbf{U} \in \mathbb{R}^{N\times M}$, $\mathbf{\Sigma} \in \mathbb{R}^{M\times M}$ and $\mathbf{V} \in \mathbb{R}^{M\times M}$ up to the signal subspace. Therefore, the SVD low-rank approximation of $\mathbf{X}$, denoted as $\tilde{\mathbf{X}} \in \mathbb{R}^{N\times M}$, can be expressed as

$$\tilde{\mathbf{X}} = \mathbf{U}_s\mathbf{\Sigma}_s\mathbf{V}_s^{\text{H}}, \tag{A.3}$$

where the diagonal of $\mathbf{\Sigma}_s \in \mathbb{R}^{d\times d}$ contains the d singular values of $\tilde{\mathbf{X}}$, whereas the columns of $\mathbf{U}_s \in \mathbb{R}^{N\times d}$ and $\mathbf{V}_s \in \mathbb{R}^{M\times d}$ correspond to the singular vectors of $\tilde{\mathbf{X}}$.

Furthermore, a multidimensional extension of the SVD low-rank approximation can be defined for higher order tensors. In such case, the SVD is performed on each unfolding of the data tensor by using the Higher Order Singular Value Decomposition (HOSVD) technique. If $\mathcal{X} = \mathcal{X}_0 + \mathcal{N}$ is the noisy data tensor, where $\mathcal{X}_0 \in \mathbb{R}^{N_1\times\cdots\times N_R\times M}$ is the noise-free tensor and $\mathcal{N} \in \mathbb{R}^{N_1\times\cdots\times N_R\times M}$ is the noise samples tensor, the SVD of $[\mathcal{X}]_{(r)}$ for $r = 1, \dots, R$ is given by (Haardt et al., 2014)

$$[\mathcal{X}]_{(r)} = \left[\mathbf{U}_r^{[s]}\mathbf{U}_r^{[n]}\right] \cdot \begin{bmatrix}\mathbf{\Sigma}_r^{[s]} & \mathbf{0}_{d\times(N-d)} \\ \mathbf{0}_{(N-d)\times d} & \mathbf{\Sigma}_r^{[n]}\end{bmatrix} \cdot \left[\mathbf{V}_r^{[s]}\mathbf{V}_r^{[n]}\right]^{\text{H}}, \tag{A.4}$$

where $\mathbf{U}_r^{[s]} \in \mathbb{R}^{N_r\times I_r}$ and $\mathbf{U}_r^{[n]} \in \mathbb{R}^{N_r\times(N_r-I_r)}$ are the basis for the rth space as well as its orthogonal complement, respectively, and I_r is the r-rank of the noise-free tensor $\mathcal{X}_0$.

From the signal r-spaces $\mathbf{U}_r^{[s]}$, the truncated core tensor $\mathcal{S}^{[s]} \in \mathbb{R}^{I_1\times\cdots\times I_R\times I_{R+1}}$ can be estimated as follows

$$\mathcal{S}^{[s]} = \mathcal{X} \times_1 \mathbf{U}_1^{[s]\text{H}} \dots \times_R \mathbf{U}_R^{[s]\text{H}} \times_{R+1} \mathbf{U}_{R+1}^{[s]\text{H}}. \tag{A.5}$$

Finally, from $\mathcal{S}^{[s]}$ in (A.5), the denoised data tensor $\tilde{\mathcal{X}}$ can be expressed as

$$\tilde{\mathcal{X}} = \mathcal{S}^{[s]} \times_1 \mathbf{U}_1^{[s]} \dots \times_R \mathbf{U}_R^{[s]} \times_{R+1} \mathbf{U}_{R+1}^{[s]}. \tag{A.6}$$

Similarly to the HOSVD, the Higher Order Orthogonal Iteration can be considered as a multilinear generalization of the SVD low-rank approximation for matrices. However, HOOI is an iterative technique which finds the best rank-$(I_1, \dots, I_{R+1})$ tensor $\tilde{\mathcal{X}} \in \mathbb{R}^{N_1\times\cdots\times N_R\times M}$ such that the least squares cost function $\|\mathcal{X} - \tilde{\mathcal{X}}\|^2$ is minimal (Lathauwer et al., 2000).

The HOOI algorithm estimates the singular matrices $\mathbf{U}_r^{[s]}$ for $r = 1, \dots, R$ by sequentially applying the SVD at each iteration $j = 0, \dots, J$ until some stopping criterion is satisfied, i.e.,

$$\begin{aligned}&[\mathcal{X} \times_2 \mathbf{U}_2^{(j)[s]\text{H}} \times_3 \mathbf{U}_3^{(j)[s]\text{H}} \dots \times_{R+1} \mathbf{U}_{R+1}^{(j)[s]\text{H}}]_{(1)} = \mathbf{U}_1^{(j+1)[s]}\mathbf{\Sigma}_1^{(j+1)[s]}\mathbf{V}_1^{(j+1)[s]\text{H}}, \\ &[\mathcal{X} \times_1 \mathbf{U}_1^{(j+1)[s]\text{H}} \times_3 \mathbf{U}_3^{(j)[s]\text{H}} \dots \times_{R+1} \mathbf{U}_{R+1}^{(j)[s]\text{H}}]_{(2)} = \mathbf{U}_2^{(j+1)[s]}\mathbf{\Sigma}_2^{(j+1)[s]}\mathbf{V}_2^{(j+1)[s]\text{H}}, \\ &\vdots \\ &[\mathcal{X} \times_1 \mathbf{U}_1^{(j+1)[s]\text{H}} \times_2 \mathbf{U}_2^{(j+1)[s]\text{H}} \dots \times_R \mathbf{U}_R^{(j+1)[s]\text{H}}]_{(R+1)} = \mathbf{U}_{R+1}^{(j+1)[s]}\mathbf{\Sigma}_{R+1}^{(j+1)[s]}\mathbf{V}_{R+1}^{(j+1)[s]\text{H}},\end{aligned} \tag{A.7}$$

where, at $j = 0$, the HOOI is initialized with the matrices obtained from the HOSVD of $\mathcal{X}$.

From the singular matrices $\mathbf{U}_r^{[s]}$ for $r = 1, \dots, R$ obtained after the Jth iteration of HOOI in (A.7), the core tensor $\mathcal{S} \in \mathbb{R}^{I_1\times\cdots\times I_R\times I_{R+1}}$ can be computed as follows

$$\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^{[s]\text{H}} \dots \times_R \mathbf{U}_R^{[s]\text{H}} \times_{R+1} \mathbf{U}_{R+1}^{[s]\text{H}}. \tag{A.8}$$

Finally, from the core tensor $\mathcal{S}$ computed in (A.8), the denoised dataset tensor $\tilde{\mathcal{X}} \in \mathbb{R}^{N_1\times\cdots\times N_R\times M}$ is given by

$$\tilde{\mathcal{X}} = \mathcal{S} \times_1 \mathbf{U}_1^{[s]} \dots \times_R \mathbf{U}_R^{[s]} \times_{R+1} \mathbf{U}_{R+1}^{[s]}. \tag{A.9}$$

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Automat. Control 19 (6), 716–723. http://dx.doi.org/10.1109/TAC.1974.1100705.

Ballester-Ripoll, R., Suter, S.K., Pajarola, R., 2015. Analysis of tensor approximation for compression-domain volume visualization. Comput. Graph. 47, 34–47. http://dx.doi.org/10.1016/j.cag.2014.10.002.

Bamakan, S.M.H., Wang, H., Yingjie, T., Shi, Y., 2016. An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomput. 199, 90–102. http://dx.doi.org/10.1016/j.neucom.2016.03.031.

Barron, A., Rissanen, J., Yu, B., 1998. The minimum description length principle in coding and modeling. IEEE Trans. Inform. Theory 44 (6), 2743–2760. http://dx.doi.org/10.1109/18.720554.

Canadian Institute for Cybersecurity, 2009. NSL-KDD dataset. https://www.unb.ca/cic/datasets/nsl.html, Last accessed on 2019-06-02.

Canadian Institute for Cybersecurity, 2017. CICFlowMeter: Network traffic flow analyzer. http://www.netflowmeter.ca/netflowmeter.html, Last accessed on 2019-04-22.

Canadian Institute for Cybersecurity, 2019. DDoS Evaluation Dataset (CICDDoS2019). https://www.unb.ca/cic/datasets/ddos-2019.html, Last accessed on 2019-10-02.

Cichocki, A., Mandic, D., Lathauwer, L.D., Zhou, G., Zhao, Q., Caiafa, C., Phan, H.A., 2015. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. IEEE Signal Process. Mag. 32 (2), 145–163. http://dx.doi.org/10.1109/MSP.2013.2297439.

Cisco, 2019. Cisco Visual Networking Index: Forecast and Trends, 2017–2022. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html, Last accessed on 2020-04-09.

da Costa, J.P.C.L., de Freitas, E.P., David, B.M., Serrano, A.M.R., Amaral, D., de Sousa, Jr., R.T., 2012a. Improved blind automatic malicious activity detection in honeypot data. In: Proc. 7th Int. Conf. Forensic Comput. Sci. (ICoFCS 2012). pp. 46–55. http://dx.doi.org/10.5769/C2012008.

da Costa, J.P.C.L., de Freitas, E.P., Serrano, A.M.R., de Sousa, Jr., R.T., 2012b. Improved parallel approach to PCA based malicious activity detection in distributed honeypot data. Int. J. Forensic Comput. Sci. 2, 8–20. http://dx.doi.org/10.5769/J201202001.

da Costa, J.P.C.L., Roemer, F., Haardt, M., de Sousa, Jr., R.T., 2011. Multi-dimensional model order selection. EURASIP J. Adv. Signal Process. 2011 (26), 1–13. http://dx.doi.org/10.1186/1687-6180-2011-26.

David, B.M., da Costa, J.P.C.L., de Freitas, E.P., Serrano, A.M.R., de Sousa, Jr., R.T., 2011. A parallel approach to PCA based malicious activity detection in distributed honeypot data. Int. J. Forensic Comput. Sci. 1, 8–27. http://dx.doi.org/10.5769/J201101001.

Douligeris, C., Mitrokotsa, A., 2004. DDoS attacks and defense mechanisms: classification and state-of-the-art. Comput. Netw. 44 (5), 643–666. http://dx.doi.org/10.1016/j.comnet.2003.10.003.

Garcia, L.P.F., de Carvalho, A.C.P.L.F., Lorena, A.C., 2013. Noisy data set identification. In: Hybrid Artif. Intell. Syst.. pp. 629–638. http://dx.doi.org/10.1007/978-3-642-40846-5_63.

GitHub.com, 2018. February 28th DDoS Incident Report. https://github.blog/2018-03-01-ddos-incident-report/, Last accessed on 2019-05-14.

Gogoi, P., Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K., 2012. Packet and flow based network intrusion dataset. In: Contemporary Comput.. pp. 322–334. http://dx.doi.org/10.1007/978-3-642-32129-0_34.

Gomes, P.R.B., da Costa, J.P.C.L., de Almeida, A.L.F., de Sousa, Jr., R.T., 2019. Tensor-based multiple denoising via successive spatial smoothing, low-rank approximation and reconstruction for R-D sensor array processing. Digit. Signal Process. 89, 1–7. http://dx.doi.org/10.1016/j.dsp.2019.01.005.

Gualberto, E.S., de Sousa, Jr., R.T., Vieira, T.P.B., da Costa, J.P.C.L., Duque, C.G., 2020. From feature engineering and topics models to enhanced prediction rates in phishing detection. IEEE Access 8, 76368–76385. http://dx.doi.org/10.1109/ACCESS.2020.2989126.

Guo, Q., Zhang, C., Zhang, Y., Liu, H., 2016. An efficient SVD-based method for image denoising. IEEE Trans. Circuits Syst. Video Technol. 26 (5), 868–880. http://dx.doi.org/10.1109/TCSVT.2015.2416631.

Haardt, M., Pesavento, M., Roemer, F., Nabil El Korso, M., 2014. Chapter 15 - subspace methods and exploitation of special array structures. In: Zoubir, A.M., Viberg, M., Chellappa, R., Theodoridis, S. (Eds.), Academic Press Library in Signal Processing: Volume 3. In: Academic Press Library in Signal Processing, vol. 3, Elsevier, pp. 651–717. http://dx.doi.org/10.1016/B978-0-12-411597-2.00015-1.

Haardt, M., Roemer, F., Del Galdo, G., 2008. Higher-Order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems. IEEE Trans. Signal Process. 56 (7), 3198–3213. http://dx.doi.org/10.1109/TSP.2008.917929.

Hosseini, S., Azizi, M., 2019. The hybrid technique for DDoS detection with supervised learning algorithms. Comput. Netw. 158, 35–45. http://dx.doi.org/10.1016/j.comnet.2019.04.027.

Jha, S.K., Yadava, R.D.S., 2011. Denoising by singular value decomposition and its application to electronic nose data processing. IEEE Sens. J. 11 (1), 35–44. http://dx.doi.org/10.1109/JSEN.2010.2049351.

Ji, S., Ye, J., 2008. Generalized Linear Discriminant Analysis: A unified framework and efficient model selection. IEEE Trans. Neural Netw. 19 (10), 1768–1782. http://dx.doi.org/10.1109/TNN.2008.2002078.

Kuhn, M., Johnson, K., 2013. Applied Predictive Modeling. Springer, New York, NY.

Kushwah, G.S., Ali, S.T., 2017. Detecting DDoS attacks in cloud computing using ANN and black hole optimization. In: 2017 2nd Int. Conf. Telecommun. Netw. (TEL-NET). pp. 1–5. http://dx.doi.org/10.1109/TEL-NET.2017.8343555.

Lathauwer, L.D., Moor, B.D., Vandewalle, J., 2000. On the best rank-1 and rank-(R_1, R_2, …, R_n) approximation of higher-order tensors. SIAM J. Matrix Anal. Appl. 21 (4), 1324–1342. http://dx.doi.org/10.1137/S0895479898346995.

Liu, J., Xiao, Y., Ghaboosi, K., Deng, H., Zhang, J., 2009. Botnet: Classification, attacks, detection, tracing, and preventive measures. EURASIP J. Wirel. Commun. Netw. 2009 (1), 692654. http://dx.doi.org/10.1155/2009/692654.

Mahjabin, T., Xiao, Y., Sun, G., Jiang, W., 2017. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. Int. J. Distrib. Sensor Netw. 13 (12), 1550147717741463. http://dx.doi.org/10.1177/1550147717741463.

Maranhão, J.P.A., da Costa, J.P.C.L., Javidi, E., Junior, J.M., de Sousa, Jr., R.T., 2019. Multidimensional antenna array based framework for drone localization in multipath environments. In: 2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS). pp. 1–8. http://dx.doi.org/10.1109/ICSPCS47537.2019.9008610.

Osanaiye, O., Cai, H., Choo, K.R., Dehghantanha, A., Xu, Z., Dlodlo, M., 2016. Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. EURASIP J. Wirel. Commun. Netw. 130, http://dx.doi.org/10.1186/s13638-016-0623-3.

Radoi, E., Quinquis, A., 2004. A new method for estimating the number of harmonic components in noise with application in high resolution radar. EURASIP J. Adv. Signal Process. 2004 (8), 615890. http://dx.doi.org/10.1155/S1110865704401097.

Rajwade, A., Rangarajan, A., Banerjee, A., 2013. Image denoising using the Higher Order Singular Value Decomposition. IEEE Trans. Pattern Anal. Mach. Intell. 35 (4), 849–862. http://dx.doi.org/10.1109/TPAMI.2012.140.

Sáez, J.A., Galar, M., Luengo, J., Herrera, F., 2013. Tackling the problem of classification with noisy data using multiple classifier systems: Analysis of the performance and robustness. Inform. Sci. 247, 1–20. http://dx.doi.org/10.1016/j.ins.2013.06.002.

Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy - Volume 1: ICISSP, INSTICC. SciTePress, pp. 108–116. http://dx.doi.org/10.5220/0006639801080116.

Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A., 2019. Developing realistic Distributed Denial of Service (DDoS) attack dataset and taxonomy. In: 2019 Int. Carnahan Conf. Security Technol. (ICCST). pp. 1–8. http://dx.doi.org/10.1109/CCST.2019.8888419.

Tan, P.N., Steinbach, M., Kumar, V., 2006. Introduction to Data Mining. Pearson Education.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symp. Comput. Intell. Security Defense Appl.. pp. 1–6. http://dx.doi.org/10.1109/CISDA.2009.5356528.

Thakre, A., Haardt, M., Roemer, F., Giridhar, K., 2010. Tensor-based spatial smoothing (TB-SS) using multiple snapshots. IEEE Trans. Signal Process. 58 (5), 2715–2728. http://dx.doi.org/10.1109/TSP.2010.2043141.

Ulfarsson, M.O., Solo, V., 2008. Dimension estimation in noisy PCA with SURE and random matrix theory. IEEE Trans. Signal Process. 56 (12), 5804–5816. http://dx.doi.org/10.1109/TSP.2008.2005865.

Vieira, T.P.B., Tenório, D.F., da Costa, J.P.C.L., de Freitas, E.P., Galdo, G.D., de Sousa Júnior, R.T., 2017. Model order selection and eigen similarity based framework for detection and identification of network attacks. J. Netw. Comput. Appl. 90, 26–41. http://dx.doi.org/10.1016/j.jnca.2017.04.012.

Wang, M., Lu, Y., Qin, J., 2020. A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Comput. Secur. 88, 101645. http://dx.doi.org/10.1016/j.cose.2019.101645.

Wax, M., Kailath, T., 1985. Detection of signals by information theoretic criteria. IEEE Trans. Acoust. Speech Signal Process. 33 (2), 387–392. http://dx.doi.org/10.1109/TASSP.1985.1164557.

Yusof, A.R., Udzir, N.I., Selamat, A., Hamdan, H., Abdullah, M.T., 2017. Adaptive feature selection for denial of services (DoS) attack. In: 2017 IEEE Conf. Appl., Inf. Netw. Secur. (AINS). pp. 81–84. http://dx.doi.org/10.1109/AINS.2017.8270429.

Zanatta, M.R., de Mendonça, F.L.L., Antreich, F., de Lima, D.V., Miranda, R.K., Galdo, G.D., da Costa, J.P.C.L., 2019. Tensor-based time-delay estimation for second and third generation global positioning system. Digit. Signal Process. 92, 1–19. http://dx.doi.org/10.1016/j.dsp.2019.04.003.

Zhao, L.C., Krishnaiah, P.R., Bai, Z.D., 1986. On detection of the number of signals in presence of white noise. J. Multivariate Anal. 20 (1), 1–25. http://dx.doi.org/10.1016/0047-259X(86)90017-5.

João Paulo Abreu Maranhão received his bachelor degree in telecommunications engineering in 2003 and his M.Sc. degree in systems and computing in 2014 both from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil. Currently, he is a Ph.D. student at the Graduate Program in Electrical Engineering at the University of Brasilia (UnB), Brazil, researching on multidimensional signal processing and machine learning applied to network intrusion detection systems.

João Paulo Carvalho Lustosa da Costa received the diploma degree in electronic engineering in 2003 from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil, his M.Sc. degree in telecommunications in 2006 from University of Brasilia (UnB), Brazil, and his Ph.D. degree in electrical and information engineering in 2010 at TU Ilmenau, Germany. Since 2010, he coordinates the Laboratory of Array Signal Processing (LASP). Since 2014 he coordinates the main project related to distance learning courses at the National School of Public Administration and a special visiting researcher (PVE) project related to satellite communication and navigation with the German Aerospace Center (DLR) supported by the Brazilian government.

Elnaz Javidi received her master degree and her bachelor degree in biomedical engineering from the Ilmenau University of Technology (TU Ilmenau) in 2019 and from the Islamic Azad University in 2011, respectively. Since 2019, she is a Ph.D. student of the Post-graduate Program in Mechatronic Systems (PPMEC) in the area of intelligent systems at the University of Brasília (UnB). Her research interests include automotive testing engineering, array signal processing, biomedical signal processing and ophthalmologic engineering.

César Augusto Borges de Andrade received his bachelor degree in data processing in 1997 from the Mackenzie Presbiterian University, São Paulo, Brazil, and his M.Sc. degree in systems and computing in 2013 from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil. Currently, he is a Ph.D. student at the Graduate Program in Electrical Engineering at the University of Brasilia (UnB), Brazil, researching on machine learning applied to malicious software detection systems.

Rafael T. de Sousa Jr. was born in Campina Grande, Brazil, in 1961. He graduated in electrical engineering from the Federal University of Paraíba (UFPB), Campina Grande, Brazil, 1984, and the Ph.D. degree in telecommunications from the University of Rennes 1, Rennes, France, 1988. He was a Software and Network Engineer in the private sector, from 1989 to 1996. Since 1996, he has been a Network Engineering Professor with the Electrical Engineering Department, University of Brasília, Brazil. From 2006 to 2007, his work was supported by the Brazilian R&D Agency CNPq. He took a sabbatical year with the Group for the Security of Information Systems and Networks, Ecole Superiéure d'Electricité, Rennes. He is currently a member of the Post-Graduate Program on Electrical Engineering and supervises the Decision Technologies Laboratory (LATITUDE), University of Brasília. His research interests include distributed systems and network management and security.