IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 October 2023, accepted 16 October 2023, date of publication 18 October 2023, date of current version 24 October 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3325721*

RESEARCH ARTICLE

# Real-Time Clustering Based on Deep Embeddings for Threat Detection in 6G Networks

**EMILIO PAOLINI[1,2,3], (Student Member, IEEE),**
**LUCA VALCARENGHI[1], (Senior Member, IEEE), LUCA MAGGIANI[3],**
**AND NICOLA ANDRIOLLI[2], (Senior Member, IEEE)**
[1]TeCIP Institute, Scuola Superiore Sant'Anna, 56124 Pisa, Italy
[2]CNR-IEIIT, 56122 Pisa, Italy
[3]Sma-RTy Italia SRL, 20061 Carugate, Italy

Corresponding author: Emilio Paolini (emilio.paolini@santannapisa.it)

This work was supported in part by the Key Digital Technologies Joint Undertaking and its Members [including top-up funding by the Italian Ministry of Research and University (MUR)] through the Project CLEVER under Project 101097560; in part by the European Union through the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, a partnership on ''Telecommunications of the Future'' by the Program RESTART, under Grant PE00000001; and in part by the Project Smart Computing and Communication at the Edge (SMARTCOM) funded by Sma-RTy and CNR.

**ABSTRACT** Trials and deployments of sixth Generation (6G) wireless networks, delivering extreme capacity, reliability, and efficiency, are expected as early as 2030. Attempts from both industry and academia are trying to define the next generation network infrastructure. 6G will set in motion the fourth industrial revolution, delivering global, integrated intelligence. In this scenario, Artificial Intelligence (AI)-assisted architecture for 6G networks will realize knowledge discovery, automatic network adjustment and intelligent service provisioning. The long-term vision for implementing 6G security is to implement an autonomous, self-learning AI-assisted architecture that can perform threat mitigation without disrupting the normal use, enabling the resilience and reliability of the network and fulfilling security automation. This work proposes a first implementation of a proactive threat discovery service to be deployed at 6G base stations, paving the way for collective network intelligence in the context of cybersecurity mechanisms. Specifically, a fully unsupervised Deep Learning (DL) model is presented, able to recognize both Denial of Service (DoS) Hulk and DoS Goldeneye, with 97.0% and 92.2% mean F1-score respectively.

**INDEX TERMS** DoS attack detection, machine learning, autoencoder, 6G, real-time detection, autonomous networks, artificial intelligence.

## I. INTRODUCTION

The deployment of 5G network infrastructure has already begun, with a widespread growth expected in the coming years [1]. Hence academy and industries are now focusing towards 6G to fulfill the requirements of applications of the next decade. Indeed, various scenarios highlight the limitations of 5G networks in terms of data speed, latency, global coverage, and more [2]. Emerging applications like extended reality, holographic communications, and digital twin technologies will exploit the development of 6G network infrastructures to fully unlock their potentials [3].

6G networks will deliver extreme capacity, reliability, and efficiency. A key technology will be AI, enabling the transition from connected things to collective network intelligence [4], where DL is exploited to provide distributed autonomy. To enable the potentials of fully autonomous networks, AI systems need to be integrated in every aspect of network management. Although some works have proposed the exploitation of AI algorithms at the communication infrastructure level [5], none of the most recent advancements, such as AI-powered network functionalities using collective network intelligence, has been demonstrated in the context of 6G [2].

The emergence of AI-powered capabilities in 6G will unleash the potentials of proactive networks. Such networks

The associate editor coordinating the review of this manuscript and approving it for publication was Miguel López-Benítez.

can perform operations in an autonomous way, such as self-managing to maintain the desired network performance level, or self-protection to secure the network and deal with threats. Thus, security design exploiting AI systems will become more critical to autonomously detect and mitigate threats rather than current cryptographic methods [1].

In this context, the pivotal elements for enabling future networks, particularly in critical scenarios like military and banking applications, will be threat mitigation systems. Additionally, the massive device connections to 6G networks will also pose new challenges to DoS attack detection. These attacks represent a malicious effort to overwhelm a target's online services or infrastructure, making them inaccessible to legitimate users. Such attacks can also be distributed, involving a network of compromised devices, often referred to as a "botnet," coordinated to flood a target with an overwhelming volume of traffic or requests. Among most common DoS attacks, Hulk flood generates a unique pattern at each and every request, with the intention of increasing the load on the servers as well as evading any intrusion detection and prevention systems. In addition, DoS GoldenEye is also very common, exploiting HTTP Keep Alive and NoCache as flags in the HTTP header. By exploiting keep alive, an HTTP persistent connection is created, and thus a single TCP connection remains open for multiple HTTP requests/responses. Hence, multiple persistent HTTP connections with no cache lead to a situation in which the web service's resource pool gets exhausted, because there is no controlling mechanism to throttle down the requests.

Traditional methods for mitigating DoS attacks are becoming obsolete due to the rapid alteration and manipulation of attack patterns and mechanisms [6], [7]. Consequently, statistical and AI-driven methodologies can effectively address various forms of malicious traffic [8], by identifying, mitigating, and preventing these attacks. In addition, unsupervised learning methods can be exploited to perform threat mitigation without any prior knowledge of malicious traffic. This is of particular interest in the identification of new types of DoS attacks, for which data are not available during the training phase.

Hence, with the long-term vision of realizing an autonomous and self-learning network that can independently perform threat mitigation without causing disruption to normal use, in this work we propose an unsupervised method to perform DoS attacks detection based on DL. Specifically, the proposed solution relies on a particular type of unsupervised Neural Network (NN) models to perform feature learning on unlabelled traffic flows, namely autoencoder; on top of the autoencoder latent space a Gaussian Mixture Model (GMM) [9] is used to detect malicious packets. It is worth noting that the system works on flows collected directly at the base station level and not on statistics of traffic extracted offline, i.e., it provides a robust real-time protection for future autonomous networks. The real-time characteristic of the system is essential to alleviate the DoS attack damage as it is estimated that the average cost of service disruption can reach 5600$ per minute [10].

The remainder of this paper is structured as follows: in Sec. II, related works are surveyed, highlighting the difference with the proposed approach and their suitability for 6G networks. Sec. III presents the system architecture, providing an insight on how it can be efficiently implemented at the base station of 6G network infrastructure. Sec. IV reports the experiments and discusses the results. Finally, Sec. V concludes the paper.

## II. RELATED WORKS

Enhancing network security is paramount for the safe deployment of different 6G verticals [11], [12], [13].

To address specific security challenges, researchers have focused on innovative strategies tailored for 5G/6G networks. Notably, DL systems have exhibited encouraging outcomes in countering threats [14], owing to their adeptness in extracting high-level features.

As an example, [15] introduces an Intrusion Detection System (IDS) created using a Convolutional Neural Network (CNN), designed to classify traffic flows. A comparison is drawn between this proposed approach and an Recurrent Neural Network (RNN) model, revealing the benefits of the feed-forward architecture over the recurrent alternative. Although the architecture seems promising, several limitations might hamper its deployment in future network infrastructures. First of all, it requires a supervised learning phase, thus making it unsuitable for discovering new types of attacks, emerging every day [16]. Furthermore, the training of the AI model is performed on statistics extracted from traffic flows; this approach is not suited to work on real-time traffic due to the need to wait for full traffic flows at the base station. Similarly, in [17] an RNN with an autoencoder is proposed to detect DoS attacks in Software Defined Networking (SDN) environments. This approach suffers from the same supervised training issue mentioned before. Thus, even this solution is biased by the training set that cannot represent all the possible attacks to a 6G network. In addition, new attacks appear every day, thus making this approach less effective over time. In [18] the problem of real-time threat identification is tackled. To this aim, the authors propose to pre-process packets in matrices classified by a CNN. In particular, by defining both the length of the segments of traffic flows and the time window to collect the packets, the developed solution can produce traffic observations in the form of matrices that can be used for online attack detection. The proposed system is trained in a supervised way, again limiting the threat mitigation ability to known attacks.

Hence, all the aforementioned works have limitations that hamper their deployment in future 6G network infrastructures, either concerning the issues with real-time detection or the bias given by supervised approach used in the training phase. Instead, the solution proposed in this work can address both aspects, providing a real-time and completely unsupervised threat mitigation system for a fully autonomous 6G network.

Another recent work [19] proposes a novel feature selection technique for a DNN-based IDS, exploiting standard deviation and difference of mean and median. Experiments using a DNN showed a better performance compared to existing feature selection techniques for all considered intrusion detection datasets with a reduced execution time. However, this work still relies on statistics extracted from complete traffic flows, making it unsuitable for 6G networks.

In the context of real-time unsupervised learning approaches, authors in [20] propose time-based features over multiple time windows to detect anomalous DoS traffic. However, the authors tested a single-class problem, in which attacks are identified exploiting reconstruction error of an autoencoder. On the contrary, this work paves the way for explainability, relying on a latent representation that can give more insights and enable future works to recognize threats never seen before.

A summary of the reported related works is presented in Table 1, highlighting the main contribution and the open challenges for deployment in 6G networks. The aim of this paper is to overcome the open challenges, targeting a solution suited for a 6G network base station-level implementation.

## III. SYSTEM ARCHITECTURE

This section presents the proposed autonomous DoS attack detection system, with an overview of the possible integration of the proposed solution at the Radio Access Network (RAN) level in Sec. III-A and an insight on the DL algorithms highlighted in Sec. III-B.

### A. INTEGRATION WITHIN THE NETWORK INFRASTRUCTURE

5G base stations, also known as next generation eNBs (gNBs), are crucial components of the wireless network infrastructure, providing connectivity to 5G devices, namely User Equipments (UEs).

The solution proposed in this work complies with the current gNB architecture and it is expected to be suited for future 6G base stations even though the architecture may slightly change. As depicted in Fig. 1, the presented threat mitigation system is deployed directly at the base station level, moving security functionalities to the edge, to alleviate the load of the core network and providing intelligence to the 6G nodes. In particular, such an approach can be deployed by implementing some of the 5G core network functionalities on top of the base station. This choice goes in the direction of the CN-RAN convergence, as highlighted in [21] and [22]. For instance, implementing a local User Plane Function (UPF) at the base station will allow the gNB to perform computation on user packets. Moving security functionalities to the edge of the network is really important in the context of mobile networks, primarily because of the anticipated costs linked to service disruptions, as indicated in [10]. Hence, a solution placed as near as possible to where possible threats are generated is of paramount importance. As a result, the development of an on-site solution at the base station level capable of efficiently identifying threats in real-time becomes of primary importance for the evolution of NextG wireless networks.

The processing capabilities of the 6G base stations will be leveraged to collect packets coming from the same flow to form the matrix to be passed as input to the unsupervised DL model.

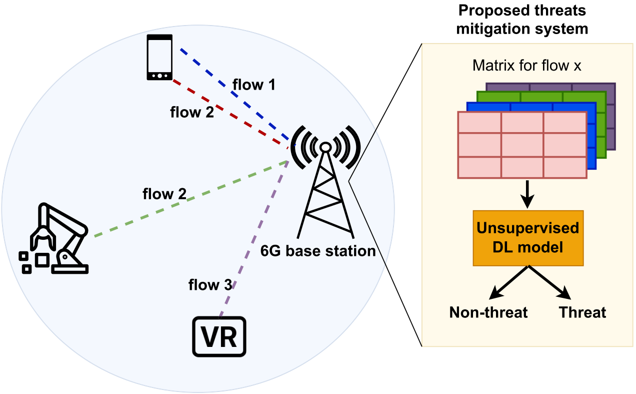


**FIGURE 1. Proposed integration of the model in a future 6G base station.**

The proposed implementation enables the evolution of the gNB, used so far with the main goal of providing connectivity to UEs. Indeed, to deploy fully-autonomous networks, base stations should be realized as intelligent nodes, each one with its own computational resources, able to make decisions on their own. Thus, the deployment of a system at the gNB-level, trainable in a totally unsupervised way and capable of recognizing threats, fits well within this vision.

### B. DL MODEL

In order to be amenable for a 6G architecture, in this work we address two aspects: (i) capability of the DL model of performing real-time detection and (ii) use an unsupervised DL model.

Concerning the first aspect, packet flows are inspected when they are relayed into the GPRS Tunnelling Protocol (GTP) tunnel at the gNB level. This is realized exploiting an UPF service deployed directly at the base station, forming a local core network. To perform real-time computations, instead of directly forwarding the incoming packets to the core network, these are pre-processed at the gNB to extract relevant features needed as inputs to the threat mitigation system. The method pre-processes packets belonging to the same flow to create a matrix of shape $n \times f$ as input to the DL algorithm, where $f$ represents the number of features and $n$ is the maximum number of packets for each flow within the time window $t$ [18]. Mathematically, supposing each packet is represented by a vector of $f$ features, i.e., $p \in \mathbb{R}^f$, the concatenation of $n$ packets corresponds to a matrix $m \in \mathbb{R}^{n \times f}$.

**TABLE 1.** A summary of considered related works.

| Work | Main contribution | Open challenges for deployment in 6G networks |
|---|---|---|
| [15] | IDS based on CNN | Supervised learning phase required and training of the CNN model performed on statistics extracted from traffic flows. |
| [17] | RNN with an autoencoder to detect DoS attacks in SDN environments | Supervised learning phase is required. |
| [18] | Techniques to pre-process packets in matrices and classification using a CNN | Supervised learning phase is required. |
| [19] | DNN with fusion of statistical importance for feature selection | Work with statistical feature. |
| [20] | Time-based features over multiple time windows to detect anomalous DDoS traffic | Single class problem relying on reconstruction error. Explainability missing. |

**Algorithm 1** Expectation-Maximization for Gaussian Mixture Model (GMM)

```
Initialize cluster means $\boldsymbol{\mu}^{(0)}$, covariances $\boldsymbol{\sigma}^{(0)}$, and mixing coefficients $\boldsymbol{\phi}^{(0)}$
Initialize convergence threshold $\epsilon$
Initialize iteration counter $t \leftarrow 0$
while not converged do
    E-step:
    for each data point $x_i$ do
        for each cluster $k$ do
            Compute the probability $\gamma_{i,k}^{(t)}$ using:
            $\gamma_{i,k}^{(t)} \leftarrow \frac{\phi_k^{(t)} \cdot \mathcal{N}(x_i|\boldsymbol{\mu}_k^{(t)},\boldsymbol{\sigma}_k^{(t)})}{\sum_{j=1}^{K} \phi_j^{(t)} \cdot \mathcal{N}(x_i|\boldsymbol{\mu}_j^{(t)},\boldsymbol{\sigma}_j^{(t)})}$
        end for
    end for
    M-step:
    for each cluster $k$ do
        Update cluster parameters:
        $\boldsymbol{\mu}_k^{(t+1)} \leftarrow \frac{\sum_{i=1}^{N} \gamma_{i,k}^{(t)} \cdot x_i}{\sum_{i=1}^{N} \gamma_{i,k}^{(t)}}$
        $\boldsymbol{\sigma}_k^{(t+1)} \leftarrow \frac{\sum_{i=1}^{N} \gamma_{i,k}^{(t)} \cdot (x_i - \boldsymbol{\mu}_k^{(t+1)}) \cdot (x_i - \boldsymbol{\mu}_k^{(t+1)})^T}{\sum_{i=1}^{N} \gamma_{i,k}^{(t)}}$
        $\phi_k^{(t+1)} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \gamma_{i,k}^{(t)}$
    end for
    Increment iteration counter $t \leftarrow t + 1$
    Check for convergence: If $\|\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}^{(t-1)}\| < \epsilon$ and $\|\boldsymbol{\sigma}^{(t)} - \boldsymbol{\sigma}^{(t-1)}\| < \epsilon$ and $\|\boldsymbol{\phi}^{(t)} - \boldsymbol{\phi}^{(t-1)}\| < \epsilon$, exit loop
end while
```

To enforce a real-time strategy, when less than $N$ packets are gathered within the specified time window, the matrix is augmented with zeros. This flexibility enables the method to accommodate scenarios characterized by extended intervals between packet arrivals. Ultimately, every attribute is scaled to fit within the range of [0,1].

Regarding the second aspect, i.e., unsupervised learning, we resort to autoencoders [23], i.e., data compression algorithms based on NNs. Autoencoders are particularly interesting for their capabilities to learn useful representations of data without the need for labels. These structures are composed of two parts: an encoder and a decoder. The encoder compresses the input data to lower dimensional features, while the decoder takes the compressed features as input and computes its output to be as close as possible to the original data. Working on the compressed feature space of the autoencoders, i.e., the latent space, can lead to good results since computations are performed on learned features. Mathematically, an autoencoder can be defined as two functions: (i) encoder $f(x) = h$, with $x$ representing the input data and $h$ the latent variable, and (ii) decoder $g(h) = x'$, with $x'$ representing the reconstructed input. The encoder part of the autoencoder can be represented as a non-linear mapping of inputs into the latent space. To train this architecture, a loss function is used to measure the difference between the original input data and the reconstructed input. The goal is to minimize this loss function so that the reconstructed input data is as close to the original input data as possible. A very common loss function is the Mean Squared Error (MSE), defined in Equation 1.

$$L = \frac{1}{N} \sum_{i=1}^{N} (x_i - x_i')^2 \tag{1}$$

On top of the latent space, a GMM is used for clustering. It assumes that the data is generated from a mixture of several Gaussian distributions, each representing a distinct cluster. Mathematically, a GMM with $K$ components is defined as:

$$q(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x|\mu_i, \sigma_i) \tag{2}$$

with $\mathcal{N}(x|\mu_i, \sigma_i)$ being the i-th gaussian component $C_i$ and $\phi_i$ representing the mixture component weights, with the constraint that $\sum_{i=1}^{K} \phi_i = 1$. These probabilistic models can be seen as a generalization of $k$-means clustering to incorporate information about the covariance structure of the data, as well as the centers of the latent Gaussians. GMM presents one advantage over $k$-means: it provides the probabilities of the data point belonging to each of the possible clusters, i.e., soft-clustering. Thus, for a given new data point, the algorithm finds its probability belonging to each cluster. Therefore, if for a particular cluster the probability is very low, this can be used to identify the data point as an outlier. Mathematically, a GMM is a linear superposition of $m$ Gaussian components, i.e., probability density functions with weight coefficients summing up to 1.

The expectation-maximization (EM) method is used for estimating the parameters, i.e., mean vector and covariance matrix, of a GMM. This algorithm has two steps:

- *E-step*: probability of each data point belonging to each of the components in the GMM, given the current estimates of the model parameters; assuming that $N$ inputs are available, then this step calculates $\forall i \in \{1, \dots, N\}, j \in \{1, \dots, K\}$
$$\gamma_{i,k} = \frac{\phi_k \mathcal{N}(x_i|\mu_k, \sigma_k)}{\sum_{j=1}^{K} \phi_j \mathcal{N}(x_i|\mu_j, \sigma_j)} \tag{3}$$
where $\gamma_{i,k}$ is the probability that input $x_i$ is generated by cluster $C_k$.
- *M-step*: the algorithm uses these probabilities to update the model parameters such that the likelihood of the data is maximized. This is done by setting the model parameters to the values that maximize the expected log-likelihood of the data, given the current estimates of the component membership probabilities. This step evaluates for every gaussian component $C_k$ the followings:
$$\phi_k = \sum_{i=1}^{N} \frac{\gamma_{i,k}}{N} \tag{4}$$
$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_{i,k} x_i}{\sum_{i=1}^{N} \gamma_{i,k}} \tag{5}$$
$$\sigma_k = \frac{\sum_{i=1}^{N} \gamma_{i,k} (x_i - \mu_k)^2}{\sum_{i=1}^{N} \gamma_{i,k}} \tag{6}$$

A pseudo-code for the expectation-maximization algorithm is reported in Algorithm 1.

Hence, the proposed approach consists in learning representations of normal traffic, fitting the GMM on deep embeddings obtained through the encoder and then identifying anomalies, i.e., attacks, as points with very low probability of belonging to normal traffic cluster. The pseudo-code for the clustering on deep embeddings is reported in Algorithm 2.

**Algorithm 2** Assign Points to Clusters Based on GMM

```
Input: Data point $x_i$, GMM model parameters: $\{\mu_k, \sigma_k, \phi_k\}_{k=i}^{K}$
Output: Cluster assignment for data point $x_i$
Initialize empty list of $r_i$ of probabilities for data point $x_i$
Compute deep embeddings for $x_i$ using encoder non-linear mapping. $h_i \leftarrow f(x_i)$
for each cluster $k$ do                ▷ Loop through each cluster component
    Evaluate probabilities for deep embeddings $h_i$ using GMM parameters:
    $r_{ik} \leftarrow \frac{\phi_k \mathcal{N}(h_i|\mu_k,\sigma_k)}{\sum_{j=1}^{K} \phi_j \mathcal{N}(h_i|\mu_j,\sigma_j)}$
end for
Assign data point $x_i$ to cluster with highest probability:
$c_i \leftarrow \arg\max_k r_{ik}$
Return cluster assignment $c_i$
```

Thus, leveraging these two methodologies, i.e., autoencoder and GMM, we implemented a system trained in a fully unsupervised way able to classify traffic flows in real-time.

## IV. EXPERIMENTS AND RESULTS

To validate the proposed approach, experiments have been carried out on the widely used CIC-IDS 2017 dataset [24], containing both benign traffic and malicious common attacks.

The exploited model architecture, obtained through a trial-and-error approach, is depicted in Fig. 2: a convolutional autoencoder is leveraged to learn features from traffic flows. The goal of this work is to give a demonstration of how the proposed approach can be used to enable unsupervised learning techniques at 6G base stations, while the model optimization is beyond the scope of this article. The encoder is composed of 5 convolutional layers, with 8, 16, 32, 64 and 192 filters, respectively. At the end of the convolutional part, 2 dense layers are used, composed of 512 and 16 neurons each, with the last layer representing the latent space encoding of the matrix traffic flows. The decoder part is mirrored with respect to the encoder structure. Rectified Linear Unit (ReLU) has been adopted as activation function. Finally, on top of the latent space a GMM is deployed to perform soft-clustering. During inference, the decoder part is discarded, since we are interested only in the soft-clustering and not in the reconstruction error.

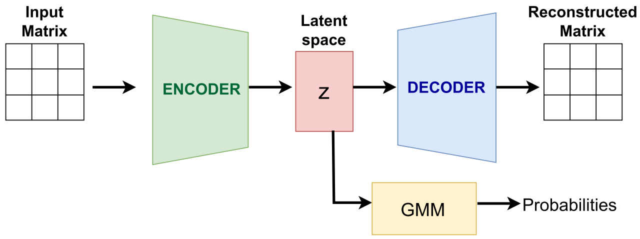


**FIGURE 2.** Autoencoder model with GMM layer stacked on top of the latent space. During inference, the decoder part is not considered.

Experiments have been carried out considering three possible values of the traffic flow length $N$: 10, 20, 50. To improve the statistical significance of the experiments, DL and GMM models have been trained and tested 5 times for each considered length $N$. The obtained results include the confidence intervals at 95% confidence level. Each training phase has been composed of 30 epochs, using Adam as optimizer and a batch size of 256.

The presented approach must be implemented in a totally unsupervised way and in a real-world scenario, where normal traffic appears with high likelihood with respect to malicious packets; indeed, most of the traffic received by base stations is related to normal activities of UEs. Instead, when an attack is performed towards the network, the traffic distribution is the opposite.

For this reason, the training set has been balanced to represent as much as possible a normal real use-cases, i.e., $99 - 1\%$ normal/attack traffic split. Furthermore, due to the unsupervised nature of the model and the ultimate goal of its deployment in a 6G base station, it would be impossible to balance such data. The proposed DL architecture, leveraging an unsupervised approach, can enable a scenario where learning is done directly on packets received at the base station, not needing human intervention in the training phase. For instance, it may be possible to rely on the learned representation of packets over time to identify new threats as points belonging to new clusters in the latent space.

To test the performance of the devised model, experiments where attacks are being performed towards the network are considered. Specifically, DoS Hulk attack types have been extracted to test the performance of the proposed model. These attacks generate unique requests bypassing caching engines, thus making the server present a unique page for each request, until resource exhaustion. In this scenario, with an attack currently carried out, the test distribution is the opposite of that of the training set.

Furthermore, another experiment has been carried out without any further retraining, considering a unknown DoS attack, namely Goldeneye. These kinds of attacks destroy the security in networks using 'HTTP Keep Alive + NoCache' as attack vector. Even in this scenario, we considered the same imbalance as in the Hulk experiment.

The accuracy metric is not ideally suited to compare the classification performance of the devised DLL model due to the different distribution among the two classes. Indeed, the accuracy is defined as:

$$\frac{TP + TN}{TP + TN + FN + FP} \tag{7}$$

with True Positive (TP) being the outcome where the model correctly predicts the positive class, True Negative (TN) being the outcome where the model correctly predicts the negative class, False Positive (FP) being the outcome where the model incorrectly predicts the positive class, and False Negative (FN) being the outcome where the model incorrectly predicts the negative class. Hence, the results might be significantly skewed due to the different number of samples belonging to different classes; thus, if a classifier is only able to label samples belonging to the majority class (i.e., it has a constant output) appearing the 99% of the time, the accuracy would be 99%. Still, the classifier would not work properly.

Thus, to address this aspect, the F1-score is adopted as performance metric, defined as:

$$F1 - score = \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{8}$$

where $P$ and $R$ represent the precision and the recall, respectively, defined as follows:

$$P = \frac{TP}{TP + FP}; \quad R = \frac{TP}{TP + FN} \tag{9}$$

Results of the DoS Hulk and DoS Goldeneye experiments are reported in Fig. 3 and Fig. 4, respectively.

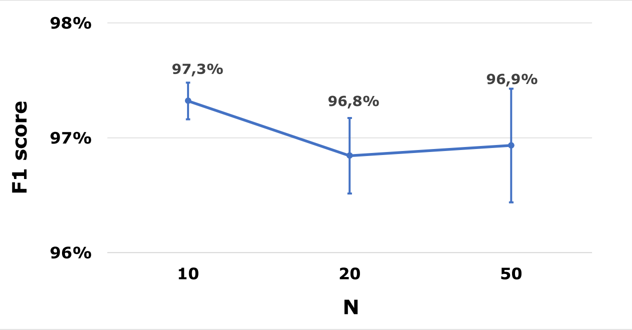


**FIGURE 3.** Results of the experiments on DoS Hulk attacks for varying *N*. The 95% confidence interval is also shown.

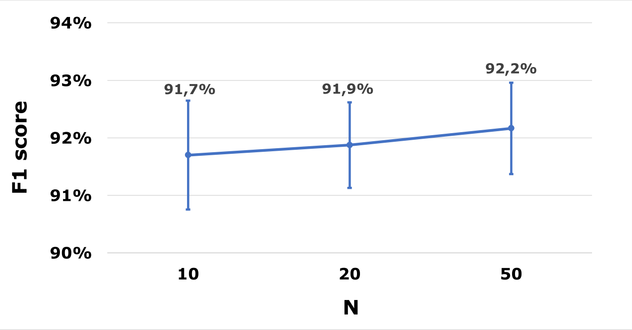


**FIGURE 4.** Results of the experiments on DoS Goldeneye attacks for varying *N*. The 95% confidence interval is also shown.

Regarding DoS Hulk, all three configurations show good F1-score values. This can be compared with the best performing model in [24], that shows an F1-score of 98%, albeit working on statistics extracted on complete traffic flows which makes it unsuitable for real-time detection. In particular, the $N = 10$ configuration reaches the highest score, i.e., 97.3%; increasing $N$ does not lead to better results for this type of DoS attack. The motivation can be traced back to the structure of DoS Hulk attacks, where a unique pattern is generated at each request, thus making $N = 10$ enough to detect these kinds of attacks.

Furthermore, although both the $N = 20$ and $N = 50$ scenarios show promising performance, i.e., 96.8% and 96.9% F1-scores respectively, the growth of $N$ increases the computational burden for the base station, leading in the worst case to a scenario where the base station cannot handle the full load.

In Table 2 we report results of other works in the literature exploiting the CIC-IDS 2017 dataset, comparing them with the results in this paper. Their F1-score is slightly higher when compared to our results. However, these implementations are not practically amenable to a 6G deployment for the reasons mentioned in Sec. II.

Concerning DoS Goldeneye, the configuration that reaches the highest F1-score is $N = 50$, i.e., 92.2%, slightly outperforming both $N = 10$ and $N = 20$ counterparts by 0.5% and 0.3%, respectively. In this experiment, increasing $N$

**TABLE 2.** Obtained F1-score and comparison with other works.

| Model | F1-score | | |
|---|---|---|---|
| ID3 [24] | 0.98 | | (all-classes) |
| DNN with fusion of statistical importance [19] | 0.9989 | | (all-classes) |
| DNN with genetic algorithm as feature selection [25] | 0.9832 | | (all-classes) |
| | N=10 | N=20 | N=50 |
| Autoencoder +GMM | 0.973 | 0.968 | 0.969 |

leads to an increasing F1-score. These attacks leverage both Keep Alive and NoCache vectors and they can be harder to detect. Hence, with DoS Goldeneye the growth of $N$ can help the DL model to better generalize on the attack structure and consequently improve the detection effectiveness.

However, this scenario still suffers from the aforementioned problems due to the larger amount of packets that need to be collected and pre-processed; furthermore, the slight F1 increase with respect to the two other configurations is not enough to justify its deployment. Thus, we can conclude that both $N = 10$ and $N = 20$ configurations give the best trade-off between F1-score and computational burden. Additionally, since the model was not trained on Goldeneye, a retraining mechanism could also be implemented to further improve the accuracy on unknown attacks when the proportion of data outliers exceed a certain threshold.

We also compared these results with a supervised approach based on an NN whose structure is equal to the encoder part of the autoencoder and that works on input matrices composed of 50 packets. The NN is trained on DoS Hulk and reaches an F1-score of 96.7%. This result is slightly below the performance of the unsupervised approach for $N = 50$, i.e., 0.2%. However, an interesting result is obtained in the following scenario: when testing the supervised approach on a previously unknown DoS attack, i.e., Goldeneye, the performance drops to a mean F1-score of 54.46%. Compared to the unsupervised approach, reaching 92.2%, this highlights the need for unsupervised approaches for detecting new types of threats that were not included in the training set.

Concerning the computational complexity of the architectures, a comparison among the number of Multiply-Accumulate (MAC) operations of the encoder part is carried out for varying $N$. Results are sketched in Fig. 5. The computation complexity of GMM, given by $O(N \times K)$, with $N$ representing the number of data points and $K$ the number of components of the GMM, is indeed negligible when compared to the one of NNs. As $N$ increases, the number of MAC operations increases, reaching more than $30M$ MACs for $N = 50$. Mean inference time for varying $N$ is also computed for 10 samples, exploiting an 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz CPU, resulting in 0.0288 s, 0.0291 s, and 0.0299 s for $N = 10$, $N = 20$, and $N = 50$, respectively. As expected, increasing the number of rows of the matrix produces an increase in the inference time, however the difference is negligible among the different values of $N$.

Finally, we can visualize the autoencoder latent space to give a glimpse on encoding data distribution. The Goldeneye test set with $N = 50$ is used as an example. The latent space is in $R^{16}$, hence it is necessary to resort to data dimensionality reduction technique to visualize it. In particular, the t-SNE technique [26] has been exploited using 2 dimensions. As sketched in Fig. 6, most of the normal traffic flows are encoded in a dense area (red dots on the right hand side of the plot) with a limited presence of malicious packets (blue dots mainly located away from normal traffic). The embedding latent space of the traffic flows further confirms the capability of the autoencoder to distinguish between the threat and non-threat flows.

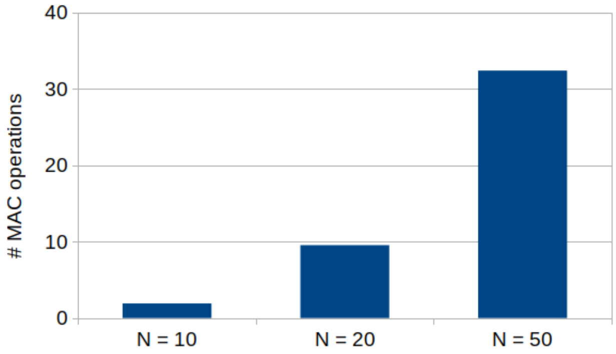


**FIGURE 5.** Number of MAC operations for varying *N*.

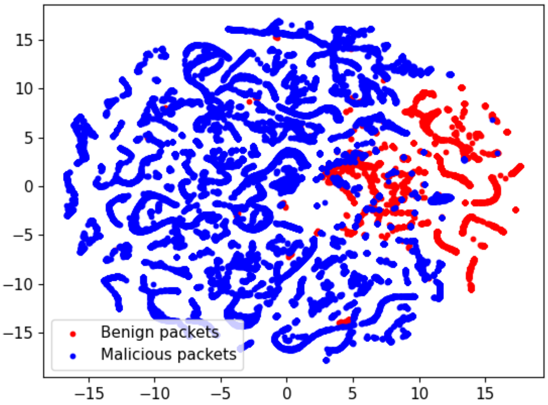


**FIGURE 6.** Visualization of the autoencoder latent space for the Goldeneye test set with $N = 50$, using t-SNE.

## V. CONCLUSION

AI will play an important role in future 6G networks, providing intelligent behaviors in any aspects of the network management. In this context, AI-based 6G security can provide smart and reliable security solutions [1]. Hence, this work shows how a threat mitigation system exploiting convolutional autoencoder and GMM can perform malicious flow detection in a totally unsupervised scenario. To validate our system, we used the CIC-IDS 2017 dataset, containing both benign and malicious traffic flows. Two possible DoS attacks, namely DoS Hulk and DoS Goldeneye, have been studied. We showed that in the first experiment, increasing $N$ does not lead to an increasing F1-score. Instead, in the other experiment, as $N$ increases, an increasing F1-score is observed. The two different behaviours in the two experiments can be traced back to the different structures of the attacks, with Hulk exploiting only the NoCache mechanism, while Goldeneye leveraging both Keep Alive and NoCache. We also compared the performance of a supervised approach to detect previously unknown threats. Results highlighted the need for an unsupervised approach to deal with such scenarios.

Summarizing, the demonstration provided in this work can pave the way for future collective network intelligence moved at the edge. Indeed, the proposed method can be considered as a first step towards a first 6G intelligent node, capable of continuously learning over flows collected at the base stations. The exploitation of the latent space in addition to the probabilities of each flow to belong to different classes can be a helpful insights to determine also new types of attacks. Indeed, if one point does not have high probabilities to belong to the known classes and it is indeed very far from all the other points, it may be an outlier corresponding to a new attack category, thus triggering a retraining phase to improve threat detection. In addition, this work can be applied as a decision support system for future IDS. Indeed, the obtained F1-score results do not highly differ from recent works related to DoS in the context of 5G networks, typically ranging from 90% to 98% [27], [28]. Moreover, a mixed approach can be devised, where the NN identifies threats and a human intervention is needed to confirm the blacklisting of that specific UE.

Furthermore, models can be collaboratively learned by multiple instances of nodes, leveraging distributed learning techniques. Towards this direction, future works will study the effectiveness of federated learning techniques to improve the proposed solution. In addition to this, a distributed attack could also be collaboratively detected by different base stations, exploiting for instance temporal information: if two or more base stations recognize the same attacks in a certain time slot, the threat may be due to a distributed attack to the network. Additionally, methods to find the best performing architecture for DoS problems, such as Neural Architecture Search (NAS), can be utilized.

## REFERENCES

[1] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "AI and 6G security: Opportunities and challenges," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit (EuCNC/6G Summit)*, Jun. 2021, pp. 616–621.

[2] C. D. Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6G frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 836–886, 2021.

[3] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021.

[4] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May 2020.

[5] W. Jiang, S. D. Anton, and H. D. Schotten, "Intelligence slicing: A unified framework to integrate artificial intelligence into 5G networks," in *Proc. 12th IFIP Wireless Mobile Netw. Conf. (WMNC)*, Sep. 2019, pp. 227–232.

[6] Y. Ma, X. Chen, W. Feng, and N. Ge, "DDoS detection for 6G Internet of Things: Spatial–temporal trust model and new architecture," *China Commun.*, vol. 19, no. 5, pp. 141–149, May 2022.

[7] B. A. Khalaf, S. A. Mostafa, A. Mustapha, M. A. Mohammed, and W. M. Abduallah, "Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods," *IEEE Access*, vol. 7, pp. 51691–51713, 2019.

[8] S.-N. Nguyen, V.-Q. Nguyen, J. Choi, and K. Kim, "Design and implementation of intrusion detection system using convolutional neural network for DoS detection," in *Proc. 2nd Int. Conf. Mach. Learn. Soft Comput.*, Feb. 2018, pp. 34–38.

[9] D. A. Reynolds, "Gaussian mixture models," *Encyclopedia Biometrics*, vol. 741, nos. 659–663, Jul. 2009.

[10] A. B. de Neira, B. Kantarci, and M. Nogueira, "Distributed denial of service attack prediction: Challenges, open issues and opportunities," *Comput. Netw.*, vol. 222, Feb. 2023, Art. no. 109553.

[11] H. Moudoud, L. Khoukhi, and S. Cherkaoui, "Prediction and detection of FDIA and DDoS attacks in 5G enabled IoT," *IEEE Netw.*, vol. 35, no. 2, pp. 194–201, Mar. 2021.

[12] M. Liyanage, I. Ahmad, A. B. Abro, A. Gurtov, and M. Ylianttila, *A Comprehensive Guide to 5G Security*. Hoboken, NJ, USA: Wiley, 2018.

[13] S. A. Abdel Hakeem, H. H. Hussein, and H. Kim, "Security requirements and challenges of 6G technologies and applications," *Sensors*, vol. 22, no. 5, p. 1969, Mar. 2022.

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

[15] J. Kim, Y. Shin, and E. Choi, "An intrusion detection model based on a convolutional neural network," *J. Multimedia Inf. Syst.*, vol. 6, no. 4, pp. 165–172, Dec. 2019.

[16] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, "Security for 5G and beyond," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3682–3722, 4th Quart., 2019.

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

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

[19] A. Thakkar and R. Lohiya, "Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system," *Inf. Fusion*, vol. 90, pp. 353–363, Feb. 2023.

[20] M. A. Salahuddin, V. Pourahmadi, H. A. Alameddine, Md. F. Bari, and R. Boutaba, "Chronos: DDoS attack detection using time-based autoencoder," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 1, pp. 627–641, Mar. 2022.

[21] J. Cha, Y. Moon, S. Cho, D. Kim, J. Choi, J. Jung, J. Lee, and S. Choi, "RAN-CN converged user-plane for 6G cellular networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 2843–2848.

[22] J. Choi, N. Sharma, S. S. Gantha, V. Mandawaria, J. Cha, D. Kim, J. Jung, J. Lee, and S. Choi, "RAN-CN converged control-plane for 6G cellular networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 1253–1258.

[23] W. H. L. Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Autoencoders," in *Machine learning*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 193–208.

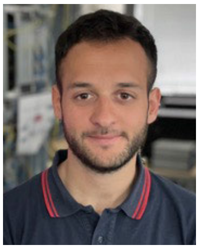
[24] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108–116, Jan. 2018.

[25] Z. Liu and Y. Shi, "A hybrid IDS using GA-based feature selection method and random forest," *Int. J. Mach. Learn. Comput.*, vol. 12, no. 2, pp. 43–50, 2022.

[26] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[27] K. Sood, M. R. Nosouhi, D. D. N. Nguyen, F. Jiang, M. Chowdhury, and R. Doss, "Intrusion detection scheme with dimensionality reduction in next generation networks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 965–979, 2023.

[28] B. Sousa, N. Magaia, and S. Silva, "An intelligent intrusion detection system for 5G-enabled Internet of Vehicles," *Electronics*, vol. 12, no. 8, p. 1757, Apr. 2023.



**EMILIO PAOLINI** (Student Member, IEEE) received the B.S. degree in computer engineering and the M.S. degree in artificial intelligence and data engineering from the University of Pisa, Italy, in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree with Scuola Superiore Sant'Anna, with a scholarship co-funded by the National Research Council (CNR) and Sma-RTy Italia SRL. His current research interest includes the integration and acceleration of artificial intelligence (AI) technologies in NextG wireless networks.

**LUCA VALCARENGHI** (Senior Member, IEEE) has been an Associate Professor with Scuola Superiore Sant'Anna, Pisa, Italy, since 2014. He has published more than 100 papers in international journals and conference proceedings and actively participated in the TPC of several IEEE conferences, such as GLOBECOM and ICC. His current research interests include optical network design, analysis, optimization, artificial intelligence optimization techniques, communication network reliability, fixed and mobile network integration, fixed network backhauling for mobile networks, and energy efficiency in communications networks.

He received the Fulbright Research Scholar Fellowship, in 2009, and the JSPS "Invitation Fellowship Program for Research in Japan (Long Term)," in 2013.

**LUCA MAGGIANI** received the joint Ph.D. degree in embedded systems from Scuola Superiore Sant'Anna, Pisa, and Universite Clermont Auvergne, Clermont-Ferrand. Since 2015, he has been running highly innovative start-ups focused on distributed artificial intelligence targeting the next internet revolution. He is currently the CEO of Sma-RTy Italia SRL. He has coauthored more than 40 international contributions to computer science and intelligence communities.

**NICOLA ANDRIOLLI** (Senior Member, IEEE) received the Laurea degree in telecommunications engineering from the University of Pisa, in 2002, and the Diploma and Ph.D. degrees from Scuola Superiore Sant'Anna, Pisa, in 2003 and 2006, respectively. He was a Visiting Student with DTU, Copenhagen, Denmark, and a Guest Researcher with NICT, Tokyo, Japan. From 2007 to 2019, he was an Assistant Professor with Scuola Superiore Sant'Anna. Since 2019, he has been a Researcher with CNR-IEIIT. He has a background in the design and performance analysis of optical circuit-switched and packet-switched networks and nodes. He has authored more than 200 publications in international journals and conferences, contributed to one IETF RFC, and filed 11 patents. His current research interests include photonic integration technologies for telecom, datacom, and computing applications, working in the field of optical communications, processing, and computing.

• • •

Open Access funding provided by 'Scuola Superiore "S.Anna" di Studi Universitari e di Perfezionamento' within the CRUI CARE Agreement