IEEE Open Journal of the Communications Society

Received 28 November 2023; revised 17 January 2024; accepted 4 February 2024. Date of publication 6 February 2024; date of current version 21 February 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3363082

Real-Time Network Packet Classification Exploiting Computer Vision Architectures

EMILIO PAOLINI [1,2,3] (Student Member, IEEE), LUCA VALCARENGHI [1] (Senior Member, IEEE), LUCA MAGGIANI [3], AND NICOLA ANDRIOLLI [4] (Senior Member, IEEE)

[1] Scuola Superiore Sant'Anna, TeCIP Institute, 56124 Pisa, Italy

[2] CNR, Istituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni, 56122 Pisa, Italy

[3] Sma-RTy Italia, 20061 Carugate, Italy

[4] Department of Information Engineering, University of Pisa, 56122 Pisa, Italy

CORRESPONDING AUTHOR: E. PAOLINI (e-mail: emilio.paolini@santannapisa.it).

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

ABSTRACT Forthcoming 6G/NextG networks highlight the need for advanced Artificial Intelligence (AI)-based security mechanisms to identify malicious activities and adapt to emerging threats. In this context, the integration of computer vision techniques into the cybersecurity field is promising due to their potential for sophisticated pattern recognition. In this paper we introduce a computationally efficient classification scheme acting directly on the raw packets collected at base stations and enforcing real-time conversion of packets into images. The innovative points of the proposed solution are the lightweight implementation, aligning well with the demands of future 6G networks, and the operation at network edge, enabling early threat identification as close as possible to the packet origin. We investigate the performance of this approach both in terms of F1-score and prediction time using state-of-the-art computer vision architectures and a customized Convolutional Neural Network (CNN) in an intrusion detection problem using a 5G dataset. Experimental results show the superiority of the CNN architecture over complex models. Across multiple packet window sizes N (i.e., 10, 50, 100 packets), the CNN consistently outperforms the other state-of-the-art computer vision models, achieving very high F1-scores (namely, 0.99593, 0.99860, 0.99895). A scalability analysis highlights a trade-off between CNN scalability and performance, where larger N values lead to increased prediction time. On the other hand, the other computer vision models exhibit better scalability, enabling an optimal model selection without trade-offs.

INDEX TERMS DoS, computer vision, artificial intelligence, 6G networks, packet classification, convolutional neural networks.

I. INTRODUCTION

As 5G network infrastructures are being deployed, with a more pervasive growth expected in the next few years [1], both academy and industry are now focusing on 6G/NextG to fulfill the requirements of applications of the next decade. Indeed, in many scenarios the limitations of 5G networks are evident in terms of data rate, latency, global coverage, etc. [2]. Applications such as extended reality, holographic communications, and digital twin will leverage the deployment of 6G network infrastructures to fully achieve their potentials [3].

Among the many benefits, 6G networks will provide extreme capacity, reliability, and efficiency. To achieve these challenging performance targets, it is expected that 6G networks deploy intelligent operations in both network orchestration and management [4]. Hence, along with

network simplification exploiting Radio Access Network (RAN)-Core Network (CN) convergence, a key technology will be Artificial Intelligence (AI), enabling the transition from connected things to collective network intelligence [5], [6]. The advent of AI-driven functionalities in 6G will enable the deployment of proactive networks. These networks can perform operations in an autonomous way, such as self-management to maintain the desired network performance level, or self-protection to secure the network and deal with threats. Hence, 6G security vision has a tight integration with AI, leading to the paradigm of security automation [4]. Security design exploiting AI systems will become pivotal to autonomously detect and mitigate threats rather than current cryptographic methods [1].

Threat mitigation system, i.e., proactively recognizing and addressing potential dangers, thereby safeguarding protecting against unforeseen risks and vulnerabilities, will be the key element for enabling future networks in critical scenarios, such as military and banking applications. Additionally, the massive device connections to 6G networks will also pose new challenges to Denial of Service (DoS) attack detection, resulting in traditional DoS mitigation methods outdated [7], [8]. Subsequently, statistical and AI-based methods can cope with different types of malicious traffic [9], identifying, mitigating, and preventing these attacks.

Therefore, many works in recent years have focused on the possibility to build AI-based systems for defending wireless networks [10]. However, future networks will be characterized by heterogeneous devices and traffic, demanding more advanced classifiers. In the rapidly evolving landscape of network security, the integration of computer vision techniques for cybersecurity applications represents an opportunity, with the promise to enable sophisticated pattern recognition strategies. Indeed, similarities between DoS and computer vision technique lie in their shared purpose of complex pattern recognition. In computer vision, algorithms process visual information to recognize intricate patterns within images and video. This process involves many layers of abstraction, where lower layers detect basic features like edges, while higher layers mix these features to identify complex objects or scenes. Similarly, in the context DoS attack detection, network traffic analysis involves identifying anomalous patterns. This recognition of patterns aims at distinguishing normal network behavior from malicious activities. As in computer vision, effective DoS attack detection often requires the extraction of meaningful features from the network traffic, followed by classification or anomaly detection techniques to discern malicious behavior [11]. Hence, algorithms such as image retrieval and object shape recognition adapted from computer vision techniques can offer an effective solution to the threat identification challenge [12].

By converting network traffic data into matrix representations, computer vision techniques can be leveraged to extract meaningful patterns and features. Each network flow can be mapped into a pixel grid, where various attributes like source and destination addresses, ports are encoded. This transformation allows viewing network traffic as visual patterns, enabling the application of Convolutional Neural Networks (CNNs) and other image-based algorithms for analysis. As discussed in [13], [14], [15], leveraging the transformation of network traffic to images not only facilitates efficient and real-time data processing, but also enables the use of pre-existing image analysis tools, opening up new possibilities for enhancing network security.

Hence, in this work we firstly describe how packets, exploiting their temporal relationships, can be transformed to images ready to be used as inputs to computer vision algorithms. Then, we study the performance of this approach exploiting both well-known CNN architectures and a purpose-built CNN architecture, called afterwards customized-CNN in an intrusion detection problem, exploiting a 5G dataset. Differently from most implementations to date, the transformation of network traffic to images is done directly on raw packets, which can be directly collected at the base station, enabling a truly real-time system protection. This features is essential in a system amenable to future networks, complying with 6G requirements on latency and alleviating the DoS attack damage, since a threat can be identified quickly and as close as possible to where it is generated. Moreover, immediate detection holds great importance in this scenario due to the projected expenses associated with service interruptions [16]. Consequently, an on-site solution at the base station level that can effectively detects threats in real-time becomes of essential importance for the future of 6G/NextG wireless networks.

II. RELATED WORKS

Network security has been one of the prime concerns in 5G networks to provide increased user privacy, new trust and service models and enable the support for Internet of Things (IoT) and mission-critical applications [17], [18]. Network protection must be strengthened and enhanced for the safe deployment of different 6G verticals [19]. To overcome some of the additional security challenges imposed by novel network architectures, researchers have focused on novel approaches suitable for 6G networks. Deep Learning (DL) systems have been showing promising results in threat mitigation [20] thanks to their capability of extracting high-level features.

For example, in [21] an Intrusion Detection System (IDS) is developed based on CNN, capable of performing classification on statistics extracted from complete traffic flows of the CIC-IDS 2018 dataset [22]. The proposed solution is compared with a Recurrent Neural Network (RNN) model, showing the advantages of the feed-forward model over its recurrent counterpart. Although the architecture seems promising, an important limitation hampers its deployment in future network infrastructures: 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 complete traffic flows at the base station.

Another work that exploits Deep Neural Networks (DNNs) for an IDS is proposed in [23]. The authors carried out a comparative study of IoT IDS with three DL models: DNN, Long Short-Term Memory (LSTM), and CNN. It is shown that DL models outperform the other methods applied in IoT IDS environment. The study only focuses on the CIC-IDS 2017 dataset [22], which cannot be considered as a good benchmark for a 5G/6G scenario because the dataset has not been collected in a real 5G network and thus the packet characteristics, e.g., packet inter-arrival time, can be very different with respect to the ones of a mobile network. Furthermore, the authors use the csv format of the dataset, i.e., statistics extracted from complete traffic flows, again hampering the possibility to deploy such systems in a real-time environment.

Tailored to specific 5G datasets, both works in [24], [25], deal with traffic classification. The first works on features extracted from complete traffic flows, hampering its exploitation on 5G/6G scenarios. Concerning the latter, a PCAP-to-Embeddings techniques is proposed, where Long Short-Term Memory Autoencoders are used for embeddings generation followed by a Fully-Connected network for classification purposes.

At the border between computer vision techniques and DoS traffic detection, authors in [26] propose to exploit ResNet architecture to detect malicious packets. Results are obtained on the CICDDoS2019 dataset [27], which, although being recent, does not resemble 5G/6G traffic characteristics, such as packet inter-arrival times. Furthermore, the authors consider only ResNet as a benchmark, not exploiting other computer vision architectures.

Another interesting work in the context of computer vision techniques applied to network traffic is [11], in which the authors discuss a multivariate correlation analysis technique to accurately represent the network traffic records and convert them into corresponding images. The detection system is developed based on Earth Mover's Distance (EMD), a widely used dissimilarity measure. EMD considers cross-bin matching, resulting in a more precise evaluation of the dissimilarity between distributions compared to other dissimilarity measures like Minkowski-form distance L_p and X^2 statistics. The experiments are conducted using two old datasets [28], [29] that do not contain recent DoS threats; in addition the proposed methodology works by building normal traffic profiles, hence not being able to distinguish among different types of attacks. In the same context, in [14] the authors describe a way to capture network traffic using pcap files and then convert these into a 2D image using a visual representation tool, i.e., binvis. For efficiency, the packets are divided into multiple chunks before this conversion process. The proposed approach is limited by the exploitation of binvis tool, that might slow down when dealing with substantial volumes of data, as in 6G networks [30]. Finally, the work in [15] describes a way to transform packets into images considering both header and payload. The authors include features like source/destination host, source/destination port number, that may hinder the generalization capabilities of the ML model. Furthermore, exploiting payload data, as proposed by the authors, at the 5G/6G base stations level is unfeasible due to encrypted packets.

In contrast to all the aforementioned works, the research proposed in this paper differs in many aspects. First of all, we investigate multiple computer vision architectures, allowing us to explore a broader spectrum of possibilities in our investigation. Through the exploitation of preprocessing techniques, we discuss how a real-time transformation of network packets into images is practically possible. Furthermore, the utilization of a very recent dataset [24] collected within a 5G environment and never exploited with computer vision techniques allows us to set a first benchmark for future studies.

III. PROPOSED ARCHITECTURE

In this section, we first describe how network packets can be transformed into images, with a focus on the used features and the corresponding preprocessing techniques. Then, we give an insight on how the proposed method can be implemented in a next generation eNB (gNB), showing its amenability to future 6G/NextG wireless infrastructure.

A. FROM NETWORK TRAFFIC TO IMAGES

The packet represents the basic unit of data transferred over a computer network. Each packet contains a part of the complete message and embeds information that helps identifying the traffic flow. The latter can be identified by a 5-tuple composed of source and destination IPs, source and destination ports, and protocol used.

In this work, relying on the concept of network traffic flow, an encoding scheme to translate packet attributes into a structured format, i.e., matrices, is proposed. By structuring the input as packet matrices, we create a spatial data representation. This representation enables the Neural Network (NN) to learn the traits of both DoS attacks and benign traffic by employing convolutional filters that slide across the input, identifying crucial patterns. Network traffic classification leveraging CNNs allows us to exploit one of their main advantages: the ability to identify DoS patterns irrespective of their temporal occurrence in the data. This intrinsic quality, i.e., producing consistent outputs despite the location of patterns in the input, is one of the paramount features of CNN architectures [31].

Specifically, the approach consists in (i) identifying F features, e.g., Time-to-live and packet length, that can be extracted from packets belonging to a given flow, and (ii) defining a maximum number of packets N for each flow within the time window T [32]. Hence, the maximum size of the input matrices will be $N \times F$. To have a real-time approach, if N packets are not collected within the time window, the matrix is padded with 0s. This allows the method to adapt to situations with long packet inter-arrival time. Finally, each attribute is normalized to the interval $[0, 1]$.

TABLE 1. Features used to create input images and the corresponding preprocessing techniques.

Features Name	Preprocessing Technique
Timestamp	Interarrival time, considering timestamp=0 for first packet of the matrix
IP length	-
Time-To-Live	-
Highest layer	Encoded as a number using SHA-256 hash algorithm
IP flags	Encoded as a number using base-16 representation
Protocols	Dense vector and then from binary vector to integer representation
TCP features, including length, ACK, flags and window size (0 if not a TCP packet)	-
UDP length (0 if not an UDP packet)	-
ICMP type (0 if not an ICMP packet)	-

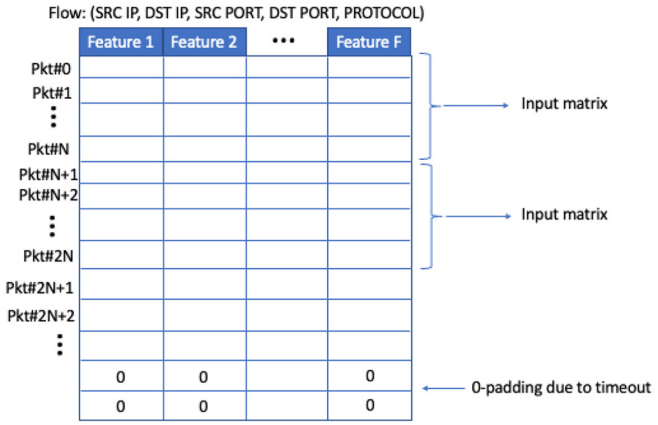


FIGURE 1. From network packets to images. From each network flow several matrices can be obtained when N packets are collected. In addition, 0-padding is also used when *N* packets do not arrive within the time window.

A summary of the proposed method, capable of transforming packet flows into images, is depicted in Fig. 1.

Concerning the features, those that are deterministic or similar, and hence can hamper the generalization of the NN models, have been excluded, such as IP addresses and TCP ports. A list of the exploited features with the corresponding preprocessing techniques is reported in Table 1.

B. INTEGRATION IN BASE STATIONS

In this section, we detail how the proposed architecture can be implemented in a future 6G base station.

In 5G networks, RAN and CN functions are strictly separated, due to the diverse protocols, interfaces, and management mechanisms. Consequently, achieving a unified, simplified network architecture integrating these components into a converged network proved challenging for 5G architectures. However, with the advent of evolving technologies and the transition to 6G networks, there is an unprecedented opportunity to rethink network architectures. The shift towards a converged RAN-CN architecture will enable the creation of a simpler, more efficient network infrastructure [6].

Hence, in future 6G networks, a new approach will be exploited providing more flexibility in network deployment, where the RAN and the CN functions can be converged in the same platform and optimized together according to the use-case requirements [6], [33], [34]. With a less strict separation between RAN and CN, each 6G base station can be equipped with functionalities coming from both CN and RAN, ultimately deploying a local CN on top of each node.

Among the novel Network Functions (NFs), the Network Data Analytics Function (NWDAF) [35] will assume a more prominent role within 6G networks, serving as a foundation for distributed network intelligence. Hence, each future base station can be equipped to host the NWDAF, offering on-demand data analytics to other NFs [36].

The NWDAF can be exploited for intelligent threat mitigation involving user data. It has the potential to gather User Plane Function (UPF) data emanating from the User Equipments (UEs) and feed this information into a DL system for the identification of malicious traffic. For instance, a threat identification and mitigation system can be implemented at the NWDAF by identifying and automatically dropping packets marked as malicious. This architecture allows the direct identification of potential threats at the base station level, alleviating the need to disseminate them throughout the network. This approach complies with the vision of placing security mechanisms as close as possible to the potential sources of threats. Furthermore, real-time detection is pivotal in this context, given the estimated cost of service disruption [16]. Thus, a base station-level solution capable of real-time threat mitigation is of significant importance for future NextG wireless networks.

The architecture of the proposed system is illustrated in Fig. 2. For conciseness, we report only the principal NFs used for this solution, i.e., UPF responsible for data forwarding, routing, and quality of service (QoS) enforcement, Session Management Function (SMF) involved in the establishment and management of the UPF and the session of the UE and NWDAF.

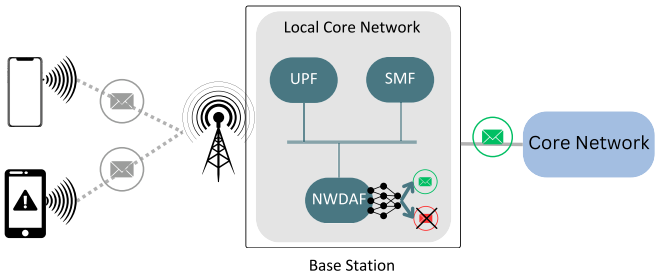


FIGURE 2. Architecture of the proposed system, reporting the main NFs used. The NextG base station is deployed along with a local CN. The proposed technique utilizes NN implemented inside the NWDAF to process packets acquired from the UPF. Malicious Packets can be identified and dropped directly at the local NWDAF.

In detail, the NWDAF will perform the following tasks: (i) *Data collection*: the NWDAF collects all network traffic flows coming from the UEs connected to the base station. (ii) *Data preprocessing*: for each packet, the NWDAF extracts the features and normalizes them, as described in Table 1. If N packets are not collected within T seconds, then it pads the matrix with 0s. (iii) *Classification*: Once the matrix is ready, the NWDAF is responsible for passing the sample to the NN, deployed along the local CN. The NN architecture can be both user-defined or rely on well-known computer vision models, as discussed in the next section.

IV. METHODOLOGY

In this section, we first describe the dataset used for the experiments, highlighting its amenability to 6G/NextG wireless networks. Then, we briefly review the state-of-the-art computer vision architectures that have been exploited. Finally, the experiments carried out are introduced and results are presented.

A. NETWORK INTRUSION DETECTION

The accuracy and the efficiency of an Machine Learning (ML)-based cybersecurity system heavily depends on the quality of the dataset and how close the behavior of the data is to the behavior in a real network scenario. One of the problem in AI-based security research is the lack of a comprehensive dataset that resembles complex 5G/6G network behaviors.

The majority of the datasets available online are outdated for modern networks as they have been compiled before some critical technological evolutions, e.g., UNSW-NB 15 [37], CTU-13 [38]. Other recent dataset available on the Web, such as the CIC-DDoS2019 [27], presents limitations in terms of many redundant records/high class unbalance. Additionally, as mentioned in Section II, the behavior of 5G/6G networks is far from the testbeds or the simulation platforms used to create this dataset.

To overcome this problem, authors in [24] recently proposed 5G-NIDD, a network intrusion detection dataset generated from a real 5G test network. The dataset is collected using the 5G Test Network (5GTN) in Oulu, Finland. 5G-NIDD presents a combination of attack traffic and benign traffic under different attack scenarios. Real mobile devices attached to the 5GTN was used to generate traffic.

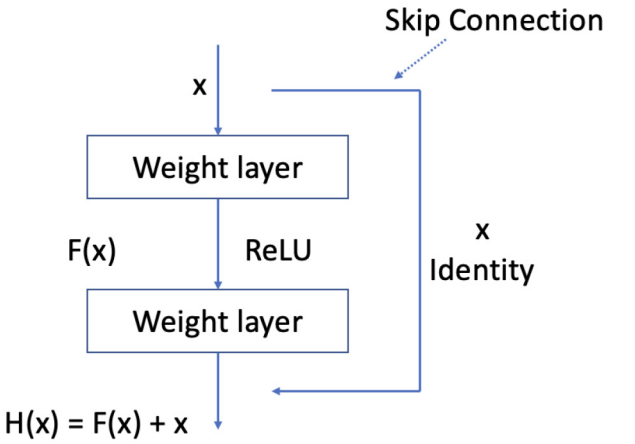


FIGURE 3. Residual block skipping two layers exploiting skip connections.

Data is extracted from two base stations, each connected to an attacker node and several benign 5G UEs. The attack scenarios include DoS attacks and port scans. Under DoS attacks, the dataset contains ICMP Flood, UDP Flood, SYN Flood, HTTP Flood, and Slowrate DoS. Under port scans, the dataset contains SYN Scan, TCP Connect Scan, and UDP Scan. The dataset is publicly available in both pcapng and csv formats. The pcapng format contains full packet payloads, while the csv files are a collection of statistics extracted for each traffic flow.

Hence, in this work, we exploit the 5G-NIDD to test the proposed architectures. A list of the attacks included in the dataset, with the corresponding description, is reported in Table 2. In the experiments, the attack type ICMP flood has not been considered since the number of samples for this class, once the dataset has been preprocessed into matrices, was very low. However, 9 classes are still present since the HTTP flood was performed using two different tools, Slowloris and Torshammer respectively.

B. NEURAL NETWORK ARCHITECTURES

In this section, we report the computer vision models that have been tested on matrices of traffic packets. In addition to state-of-the-art computer vision models, we have also designed and tested a customized-CNN, specifically aimed at recognizing threats.

One of the major innovative architecture used in computer vision is the Residual Network (ResNet) [39]. In order to solve the problem of the vanishing/exploding gradient, this architecture introduces the concept of Residual Blocks. As depicted in Fig. 3, instead of simply learning $F(x)$, the network fits $H(x) = F(x) + x$, where x is an input to the residual block and output from the previous layer.

The key concept of residual blocks relies on skip connections, as shown in Fig. 3, allowing smoother gradient flow and ensure that important features are carried until the final

TABLE 2. Attack types contained in the 5G-NIDD dataset [24] and corresponding description.

Category	Type	Description
DoS	ICMP	ICMP flood attacks exploit ICMP echo requests to inundate a target with an extremely high frequency. A subsequent ICMP echo reply is expected to be returned to the same IP address following the echo request. In this attack variant, the targeted system responds with echo replies to an extensive volume of ICMP echo requests. The heightened rate of echo requests further results in an increased frequency of response, ultimately overpowering the network and rendering services inaccessible to regular users.
	UDP Flood	In a UDP flood attack, the assailant dispatches UDP packets at an elevated pace. Because UDP operates without establishing a connection, it permits the transmission of a substantial volume of data. Upon reaching the intended destination, the UDP packets trigger a response in the form of a "Destination Unreachable" packet if no corresponding application is detected. Over the course of time, as an increasing influx of UDP packets is processed and answered, the system progressively loses responsiveness until it becomes unresponsive.
	SYN Flood	A SYN flood attack exploits the TCP three-way handshake used in communication between nodes. In this attack, the attacker sends SYN packets to initiate connections but omits the final ACK step, leaving connections half open. High-frequency transmission of SYN packets keeps numerous ports half open, overwhelming the receiver and blocking legitimate user access.
	HTTP Flood	Application layer HTTP flood attacks are aimed at the application level. They are widely used for carrying out DoS/DDoS attacks because of their demonstrated ability to emulate regular human actions, enabling them to evade detection effectively.
	Slowrate DoS	Slow-rate DoS is a cyber attack that subtly disrupts a target system by sending a controlled and gradual stream of malicious traffic, mimicking normal user behavior. This approach makes detection and mitigation challenging, gradually consuming the target's resources and causing unresponsiveness over time.
Port Scans	SYN Scan	The SYN scan is a widely used port scanning technique that leverages part of the TCP three-way handshake to discover open ports. The attacker sends a TCP packet with the SYN flag to the target; if a port is open, the target responds with a packet containing SYN and ACK flags. Closed ports send a RST packet. By analyzing these responses, the attacker identifies open, or closed ports.
	TCP Connect Scan	The TCP connect scan, similar to SYN scan, utilizes the TCP handshake for scanning, yet completes the three-way handshake, leading to a longer scanning time compared to SYN scan. However, attackers might not require the same privileges as for SYN scan. The process involves initiating a full handshake, including the completion step, which occurs after the initial SYN and SYN ACK exchanges. Open ports result in a full three-way handshake and potential data exchange, but the attacker's system terminates the connection promptly.
	UDP Scan	UDP scan involves transmitting UDP datagrams to specific ports on a target host. The attacker sends payload-lacking UDP packets to non-UDP ports, leading to minimal response likelihood. Such ports may show an open—filtered status, suggesting firewall blocking or packet forwarding. Identifying open status is challenging for non-UDP ports. For UDP ports, a response indicates openness, while an ICMP port unreachable error signifies a closed port..

layers without adding computational load to the network. In our experiments, we rely on ResNet50V2, composed of 48 convolutional layers, one max pooling layer, and one average pooling layer.

Starting from residual connections, MobileNetV2 [40] exploits an inverted residual structure where the residual connections are between the bottleneck layers. This model, well suited to mobile devices, also exploits lightweight

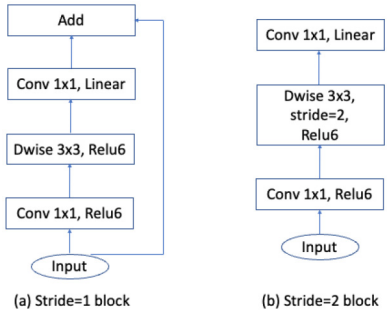


FIGURE 4. Two types of blocks for MobileNetV2: (a) residual block with stride=1 and (b) downsizing block with stride=2.

depthwise convolutions to filter features as a source of non-linearity in the intermediate layers. MobileNetV2 presents two types of blocks, a residual block with stride 1 and another block for downsizing with stride 2. These blocks are depicted in Fig. 4.

Aiming at increasing computational efficiency, authors in [41] propose EfficientNet, a more systematic method for enhancing accuracy and efficiency by scaling depth/width/resolution of CNN models. This is in contrast with conventional scaling methods that are based on random approaches, demanding manual tuning and significant effort. Specifically, the technique is based on a compound scaling method, that relies on a constant ratio to perform a balanced scaling of width, depth, and resolution.

Opposed to the computational efficiency of MobileNet and EfficientNet, DenseNet [42] connects each layer to every other layer in a feed-forward fashion, resulting in a high-demanding architecture in terms of computational resources.

An important milestone in the CNN architectures was the Inception Net [43]. The main idea behind this architecture is the Inception layer, a combination of layers with their output filters concatenated into a single output vector forming the input for the next layer.

Taking the principles of Inception to extreme, the Xception architecture is introduced [44]. In Inception, 1×1 convolutions compressed the input before applying different filters to various depth spaces. Xception reverses this process, first applying filters to depth maps and then compressing the input with 1×1 convolutions across depth, resembling a depthwise separable convolution.

In addition to these NN models, a customized-CNN has been specifically developed aiming at an accurate network packet classification, whose structure is reported in Fig. 5.

The CNN is made of 3 convolutional layers, with 8, 64 and 128 filters respectively. 3 Fully-Connected (FC) layers have been added, with 512, 128 and 9 units, respectively. As activation function, ReLU has been adopted for all layers, except for the last one that employs the SoftMax to perform classification. A summary of the state-of-the-art computer vision architectures that have been studied in the experiments along with the customized-CNN, is reported in Table 3. We can observe a large increase in parameters of the customized-CNN, mainly due to the exploitation of FC layers at the end on the convolutional section of the architecture.

TABLE 3. Architectures studied in the experiments.

Architecture Name	Version	Size(MiB)/Parameters(M)		
		N=10	N=50	N=100
customized-CNN	-	6.56/1.72	66.56/17.4	141.56/37.1
ResNet	V2	89.77/23.5	89.84/23.5	89.98/23.6
MobileNet	V2	8.53/2.23	8.57/2.25	8.66/2.27
EfficientNet	V2S	77.01/20.2	77.06/20.2	77.15/20.2
DenseNet	201	69.06/18.1	69.06/18.1	69.06/18.1
Inception	V3	83.38/21.9	83.38/21.9	83.38/21.9
Xception	-	79.44/20.8	79.51/20.8	79.58/20.9

The deployment of these models at the base-station can surely increase the power and computational resource consumption. However, advancements in hardware acceleration (like specialized chips or GPUs) [45] and optimization techniques [46] have significantly improved their efficiency. These advancements hence can enable quicker inference times and reduced energy consumption.

C. EXPERIMENTAL SETUP

In this section, the experimental setup is highlighted, describing how classification experiments have been carried out.

Once the raw packets have been obtained, a script to transform pcapng files into suited input matrices is developed, as highlighted in Section III-A. The source code is publicly available at [47]. The script allows to define the maximum number of packets for each matrix, N, thus enabling experiments for varying matrix length. In the experiments a maximum number of packets per time window $N = \{10, 50, 100\}$ has been considered. Furthermore, for DenseNet and Inception models an input resizing has been applied. In particular, for DenseNet inputs have been resized to 32×32, while for Inception the resizing has resulted in input matrices of 128×128. This is due to the fact that these models do not support small input matrices, resulting in negative dimensions of feature maps. Specifically, the resizing has been obtained with the bilinear interpolation of the input matrices.

Concerning the time window T, it has been kept fixed to $10s$ for all experiments, due to two reasons: (i) experimentally validated results have shown that 10 seconds is a good choice [32]; (ii) the proposed architecture must be amenable to a 5G/6G implementation, thus it must perform real-time detection, which cannot be achieved with a longer time window. Additionally, shortening the time window would result in most of the packet matrices being 0-padded, thus hampering the classification performance.

Once the input matrices have been created, a normalization and padding phase has been performed. Data normalization is performed to scale values to a predefined uniform range. This prevents larger values from overwhelming smaller ones

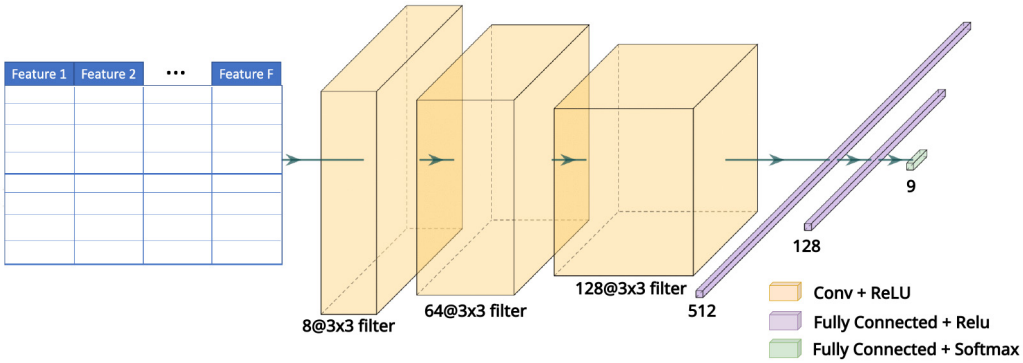


FIGURE 5. Customized CNN developed for the experiments.

during the training process. A Min-Max scaling has been adopted: the min-max values for each feature have been searched through the entire dataset, and a rescaling has been carried out, resulting in all values belonging to the range [0, 1]. Then, for input matrices with less than N rows a 0-padding strategy is adopted. Furthermore, each flow has been mapped to a specific label.

Finally, a $80-20\%$ training-test split has been performed, while keeping the original balance of the dataset. To have a fair comparison, all the models have been trained using the same training parameters, i.e., batch size, epochs, optimizer and learning rate.

Since the training set is not balanced, a class weighting technique has been adopted. Leveraging this technique, it is possible to assign higher weights to minority classes, allowing the model to pay more attention to their patterns and reducing the bias towards majority classes. The weight for each class is given by:

$$w_j = \frac{\#\text{samples}}{\#\text{classes} \times \#\text{samples\_}j}$$

where w_j is the weight for class j, #samples is the total number of samples in the dataset, #classes is the total number of unique classes in the dataset, and #samples_j is the total number of samples belonging to class j.

To evaluate the results of the experiments, we used the common evaluation metrics, such as the confusion matrix and the F1-score. Since the test set is kept unbalanced to resemble as much as possible real world data, accuracy metric can lead to skewed results and thus it is not considered. Instead, being the F1-score defined as the harmonic mean, or weighted average, of precision P and recall R values:

$$F1\text{-}score = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$
$$P = \frac{TP}{TP + FP}; \quad R = \frac{TP}{TP + FN}$$

it accounts for instances where precision or recall values are exceptionally low, resulting in a diminished score even in the case of imbalanced classes.

TABLE 4. F1-score obtained with different architectures.

Model	F1-score		
Embeddings & FC [25] (binary)	0.9989		
Embeddings & FC [25] (multi-class)	0.98666		
RF Sehan [24] (binary)	0.9989		
RF Sehan [24] (multi-class)	0.99017		
MAGNETO [48] (binary)	0.9913		
MAGNETO [48] (multi-class)	0.9985		
	N=10	N=50	N=100
Customized CNN	**0.99593**	**0.99860**	**0.99895**
ResNet	0.92266	0.92167	0.79025
MobileNet	0.78308	0.56924	0.64268
EfficientNet	0.81165	0.66327	0.79159
DenseNet	0.88585	0.94067	0.76900
Inception	0.99447	0.87925	0.69674
Xception	0.81352	0.79042	0.76556

V. RESULTS AND DISCUSSION

In this section, the obtained results are reported and discussed. Results for the different considered N values are reported in Table 4. In the results, we included the performance (in terms of F1-score) of the considered architectures, i.e., both state-of-the-art computer vision architectures and the customized-CNN, and we compare the obtained values with the best performing model for both binary and multi-class experiments of Multi Layer Perceptron (MLP), obtained by [24] and the technique introduced in [25].

Since the proposed system must be compliant with real-time requirements of future wireless networks, in the experiments we also evaluated the prediction time of the tested architectures for the entire test set exploiting an 11th Gen Intel Core i7-11700K @ 3.60GHz. Results are depicted in Fig. 6.

As reported in Table 4, the best performing model is the customized-CNN for all the values of N. Notably, this model slightly outperforms also the models proposed in [24] and [25] by 0,00878 and 0,01229, respectively. When increasing N, the F1-score of the model improves. When compared to both binary and multi-class MAGNETO [48],

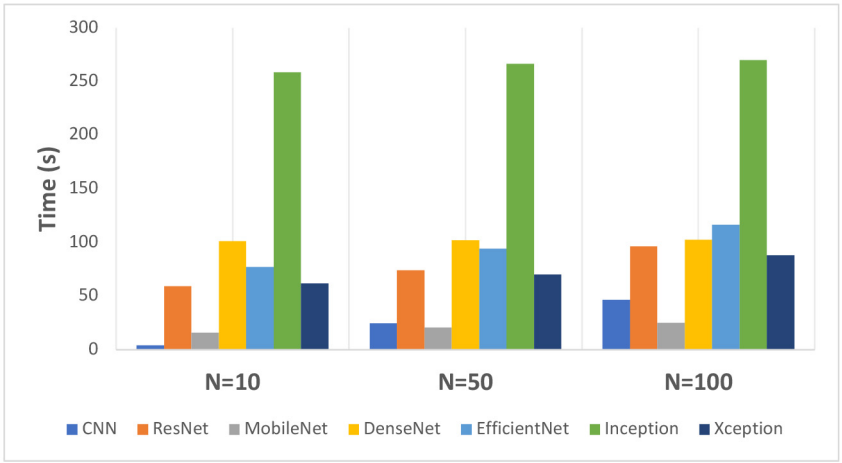


FIGURE 6. Prediction time of the tested architectures.

a work proposing the translation of network traffic into images while maintaining the retention of semantic data about the relationships between features, the customized-CNN achieves a slightly higher F1-score, of 0.00765 and 0.00045 respectively.

Concerning the other tested state-of-the-art computer vision architectures, only the ResNet and Inception models reach an F1-score above 90% for $N = 10$. However, these architectures do not increase their performance for increasing N: indeed a drop can be observed for $N = 100$ for both of them, while for $N = 50$ ResNet keeps a similar performance while Inception degrades. The behavior of Xception architecture is opposite as the CNN architecture: for increasing N, a decrease in F1-score is noticed. DenseNet obtains the highest F1-score for $N = 50$, while both MobileNet and EfficientNet have their best performing scenario for $N = 10$. The reported results have noticeably different behavior among architectures for varying N; for instance, we have a decrease in F1-score for increasing N with ResNet, Inception, and Xception, while for DenseNet the F1-score increases for $N = 50$ when compared with $N = 10$ and then it decreases for $N = 100$. These different behaviors are due to the fact that these architectures differ in many aspects, e.g., in terms of connections among neurons, number of parameters. Finally, these results highlight one important outcome: for the tasks of recognizing threats, complex models do not work better than simpler models. Indeed, the customized-CNN, composed of few layers, outperforms all state-of-the-art computer vision architectures.

A look at the confusion matrix of the best performing model, i.e., CNN, can give a better insight on the obtained results. The confusion matrices for $N = 10$, $N = 50$, and $N = 100$ are reported in Fig. 7, Fig. 8, and Fig. 9 respectively.

We can observe that for $N = 10$ the customized-CNN incorrectly classifies as belonging to class 1 (i.e., HTTP flood) almost 10% samples actually belonging to class 2 (i.e., Slowrate DoS). While for the other classes, only a small number of misclassifications can be observed. In particular, the model misclassifies samples belonging to class 6 (i.e., HTTP flood - Torshammer) as samples of class 1 and 2.

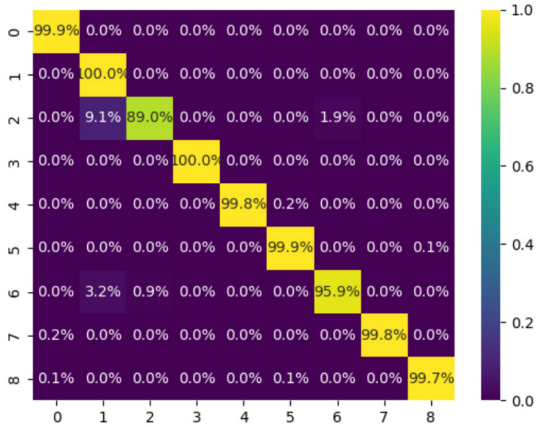


FIGURE 7. Confusion matrix for CNN model with $N = 10$.

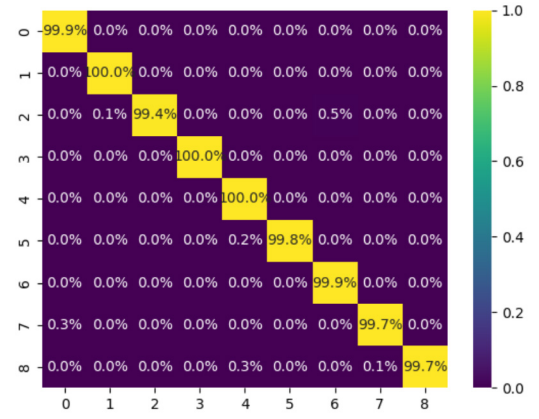


FIGURE 8. Confusion matrix for CNN model with $N = 50$.

If N is increased, instead, as depicted in Fig. 8 and 9, the issue with class 2 is solved. Indeed, for $N = 50$, the

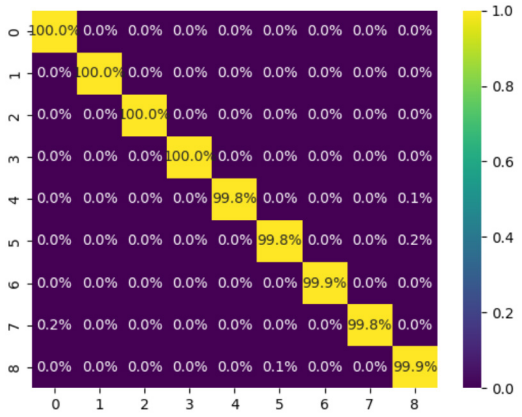


FIGURE 9. Confusion matrix for CNN model with $N = 100$.

CNN reaches an TP rate of 99.4, while for $N = 100$ the TP rate increases to 100%. This confirms that for this model, increasing the number of samples for each matrix, is helpful and leads to a better generalization.

However, increasing N leads to an increase in prediction time, as sketched in Fig. 6. Especially for the customized-CNN architecture, the prediction time goes from $\approx 4s$ for $N = 10$ to $\approx 46s$ for $N = 100$. Indeed, as reported in Table 3, the customized-CNN has a substantial increase in the number of parameters when N increases. This highlights a trade-off between the classification performance and the speed at which the classification is performed for this model. Concerning the state-of-the-art computer vision architectures, a different behavior can be noticed. Indeed, while Inception has very high prediction time even for $N = 10$, this architecture scales well, resulting in an increase of just $10s$ between the $N = 10$ and $N = 100$ scenarios. This can be traced back to the almost constant size and number of parameters for this architecture among different values of N, as reported in Table 3. Similar considerations can be derived for the other models: for instance, DenseNet shows an almost constant prediction time for different values of N, with just $\approx 1s$ increase. Hence, for these models the prediction time does not influence the choice of N and the N that leads to the best performing model can be freely chosen.

VI. CONCLUSION

6G/NextG networks will require intelligent threat mitigation systems to cope with different types of malicious traffic [9] and able to adapt to newly discovered threats. Hence, in this paper, we have explored the innovative approach of transforming network traffic packets into image representations and leveraging state-of-the-art computer vision architectures for classification.

In the experiments, an intrusion detection problem has been investigated with the goal of classifying normal and malicious behaviors. We have firstly discussed how raw packets can be converted in a real-time manner to input matrices ready to be fed into state-of-the-art computer vision architectures and the customized CNN.

Results have shown that complex models do not perform better than the developed CNN architecture. Indeed, for all the considered values of N, i.e., 10, 50, and 100, the CNN outperforms all the state-of-the-art computer vision techniques, reaching F1-scores of 0.99593, 0.99860, and 0.99895 respectively. Moreover, the scalability of our approach has been tested. Results indicate a trade-off between the scalability of the CNN and the obtained performance. Indeed, for increasing N, a non-negligible increase in prediction time is observed. On the other hand, state-of-the-art computer vision architectures, even when starting with very high prediction times, scale much better. This enables the possibility to choose the best performing model with respect to N without any trade-off.

The proposed system is just a first step on the application of computer vision techniques to network traffic analysis. Indeed, leveraging the exploitation of convolution-based models, more complex patterns can be discovered in packet matrices. For instance, a proactive approach, capable of identifying new types of attacks can be implemented. Additionally, a distributed learning approach, relying on federated/split learning techniques, will be considered in future works to enhance data privacy and model performance. Finally, hardware acceleration techniques could be studied to deploy these models on dedicated platforms, i.e., FPGA, offloading the computational workload from the gNB without compromising its performance.

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)*, 2021, pp. 616–621.

[2] C. De Alwis et al., "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] P. Porambage, G. Gür, D. P. M. Osorio, M. Liyanage, A. Gurtov, and M. Ylianttila, "The roadmap to 6G security and privacy," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1094–1122, 2021.

[5] 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/Jun. 2020.

[6] V. Ziegler, H. Viswanathan, H. Flinck, M. Hoffmann, V. Räisänen, and K. Hätönen, "6G architecture to connect the worlds," *IEEE Access*, vol. 8, pp. 173508–173520, 2020.

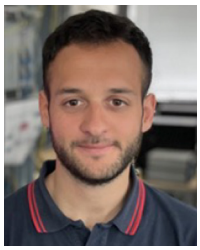
[7] Y. Ma, X. Chen, W. Feng, and N. Ge, "DDoS detection for 6G Internet of Things: Spatial-temporal trust model and new architecture," *China Commun.*, vol. 19, no. 5, pp. 141–149, May 2022.

[8] 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.

[9] 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.*, 2018, pp. 34–38.

[10] A. Suhag and A. Daniel, "Study of statistical techniques and artificial intelligence methods in distributed denial of service (DDOS) assault and defense," *J. Cyber Secur. Technol.*, vol. 7, no. 1, pp. 21–51, 2023.

[11] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of denial-of-service attacks based on computer vision techniques," *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2519–2533, Sep. 2015.
[12] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, and G. A. Shah, "IoT DoS and DDoS attack detection using ResNet," in *Proc. IEEE 23rd Int. Multitopic Conf. (INMIC)*, 2020, pp. 1–6.
[13] S. S. Kim and A. L. N. Reddy, "A study of analyzing network traffic as images in real-time," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2005, pp. 2056–2067.
[14] G. Bendiab, S. Shiaeles, A. Alruban, and N. Kolokotronis, "IoT malware network traffic classification using visual representation and deep learning," in *Proc. 6th IEEE Conf. Netw. Softw. (NetSoft)*, 2020, pp. 444–449.
[15] R. Moreira, L. F. Rodrigues, P. F. Rosa, R. L. Aguiar, and F. de Oliveira Silva, "Packet vision: A convolutional neural network approach for network traffic classification," in *Proc. 33rd Conf. Graph., Patterns Images (SIBGRAPI)*, 2020, pp. 256–263.
[16] 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.
[17] 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./Apr. 2021.
[18] M. Liyanage, I. Ahmad, A. B. Abro, A. Gurtov, and M. Ylianttila, *A Comprehensive Guide to 5G Security*. Hoboken, NJ, USA: Wiley, 2018.
[19] S. A. A. Hakeem, H. H. Hussein, and H. Kim, "Security requirements and challenges of 6G technologies and applications," *Sensors*, vol. 22, no. 5, p. 1969, 2022.
[20] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, 2017, pp. 1–8.
[21] 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, 2019.
[22] 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.
[23] J. Jose and D. V. Jose, "Deep learning algorithms for intrusion detection systems in Internet of Things using CIC-IDS 2017 dataset," *Int. J. Elect. Comput. Eng. (IJECE)*, vol. 13, no. 1, pp. 1134–1141, 2023.
[24] S. Samarakoon et al., "5G-NIDD: A comprehensive network intrusion detection dataset generated over 5G wireless network," 2022, *arXiv:2212.01298*.
[25] G. Agrafiotis, E. Makri, A. Lalas, K. Votis, D. Tzovaras, and N. Tsampieris, "A deep learning-based malware traffic classifier for 5G networks employing protocol-agnostic and PCAP-to-embeddings techniques," in *Proc. Eur. Interdiscipl. Cybersecurity. Conf.*, 2023, pp. 193–194.
[26] Z. Liu, "Detecting DDoS issues under 5G with ResNet," in *Proc. Int. Conf. Statist., Data Sci., Comput. Intell. (CSDSCI)*, 2023, pp. 257–264.
[27] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. Int. Carnahan Conf. Security Technol. (ICCST)*, 2019, pp. 1–8.
[28] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, 2012.
[29] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the JAM project," in *Proc. DARPA Inf. Survivabil. Conf. Expo. DISCEX*, 2000, pp. 130–144.
[30] S. K. Gupta, B. Pattnaik, V. Agrawal, R. S. K. Boddu, A. Srivastava, and B. Hazela, "Malware detection using genetic cascaded support vector machine classifier in Internet of Things," in *Proc. 2nd Int. Conf. Comput. Sci., Eng. Appl. (ICCSEA)*, 2022, pp. 1–6.
[31] A. Mazari and H. Sahbi, "Deep temporal pyramid design for action recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 2077–2081.
[32] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del Rincon, and D. Siracusa, "LUCID: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 876–889, Jun. 2020.
[33] J. Cha et al., "RAN-CN converged user-plane for 6G cellular networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2022, pp. 2843–2848.
[34] J. Choi et al., "RAN-CN converged control-plane for 6G cellular networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2022, pp. 1253–1258.
[35] A. Chouman, D. M. Manias, and A. Shami, "Towards supporting intelligence in 5G/6G core networks: NWDAF implementation and initial analysis," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2022, pp. 324–329.
[36] S. Sevgican, M. Turan, K. Gökarslan, H. B. Yilmaz, and T. Tugcu, "Intelligent network data analytics function in 5G cellular networks using machine learning," *J. Commun. Netw.*, vol. 22, no. 3, pp. 269–280, Jun. 2020.
[37] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.
[38] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, Sep. 2014.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
[41] M. Tan and Q. Le, "EfficientNetv2: Smaller models and faster training," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 10096–10106.
[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.
[43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
[44] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.
[45] C. Latotzke and T. Gemmeke, "Efficiency versus accuracy: A review of design techniques for DNN hardware accelerators," *IEEE Access*, vol. 9, pp. 9785–9799, 2021.
[46] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proc. IEEE*, vol. 108, no. 4, pp. 485–532, Apr. 2020.
[47] E. Paolini. "Source code for the experiments." 2023. [Online]. Available: https://github.com/emiliopaolini/5g_ddos
[48] A. Dunmore, A. Dunning, J. Jang-Jaccard, F. Sabrina, and J. Kwak, "MAGNETO and DeepInsight: Extended image translation with semantic relationships for classifying attack data with machine learning models," *Electronics*, vol. 12, no. 16, p. 3463, 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 the Scuola Superiore Sant'Anna, with a scholarship co-funded by the National Research Council and Sma-RTy Italia SRL.

His research focuses on the integration and acceleration of artificial intelligence 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. He received a Fulbright Research Scholar Fellowship, in 2009, and a JSPS "Invitation Fellowship Program for Research in Japan (Long Term)," in 2013. His research interests include optical networks design, analysis, optimization, artificial intelligence optimization techniques, communication networks reliability, fixed and mobile network integration, fixed network backhauling for mobile networks, and energy efficiency in communications networks.

LUCA MAGGIANI received the Ph.D. degree from Scuola Superiore Sant'Anna, Pisa, and the Université Clermont Auvergne, Clermont Ferrand, in 2017. He is the Co-Founder and the CEO of SmaRTy SAS, and CEO of Sma-RTy Italia SRL. He manages a Research and Development Team for advanced artificial intelligence applications and at the same time, develops computer vision application for automotive and surveillance. During his research activity, he has coauthored over 15 international reviews on embedded smart video sensors, bio-inspired systems, and their processing architectures.

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. From 2019 to 2023, he was a Researcher and then a Senior Researcher with CNR-IEIIT. Since 2024, he has been an Associate Professor with the University of Pisa. He has a background in the design and the performance analysis of optical circuit-switched and packet-switched networks and nodes. He authored more than 200 publications in international journals and conferences, contributed to one IETF RFC, and filed 11 patents. His 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