Intrusion Detection Scheme With Dimensionality Reduction in Next Generation Networks

Keshav Sood, Mohammad Reza Nosouhi, *Member, IEEE*, Dinh Duc Nha Nguyen, Frank Jiang, *Senior Member, IEEE*, Morshed Chowdhury, and Robin Doss, *Senior Member, IEEE*

***Abstract*— Due to millions of heterogeneous physical nodes, multiple-vendor and multi-tenant domains, and technologies etc., 5G has greatly expanded the threat landscape. Particularly from the high rate of traffic and ultra-low latency requirement of applications in 5G networks, the detection of the network traffic anomalies in real-time is critical. The conventional security approaches lack compatibility with modern network designs and are not much effective in 5G settings. We propose a two-stage network traffic anomaly detection system compatible with ETSI-NFV standard 5G architecture. Our architecture consists of two modules, i.e., (a) Dimensionality Reduction to compress the sample size at the edge of 5G networks and (b) Deep Neural Network classifier (DNN) that detects traffic anomalies. We have conducted our experiments using OMNET++ and ETSI-NFV (OSM MANO) 5G orchestration real platform deployed on AWS cloud systems. We have used the UNSW-NB15 data set and have shown that at dimensionality reduction factor of 81% the detection accuracy obtained is 98%. The proposal is compared with other recent approaches to show the overall merit of the architecture.**

***Index Terms*— 5G security, network security, next generation networks, anomaly detection, dimensionality reduction.**

I. Introduction

THE integration of 5G, Software-defined Networking (SDN), Network Function Virtualization (NFV), Internet of Things (IoT) and other modern technologies has created the concept of Next Generation Networking (NGN). In NGNs service-related functions are independent from underlying transport-related technologies [1]. Unfortunately, in NGNs we have seen unique security challenges different from the conventional networking. 1) The NGN applications (for example 5G-IoT industrial network applications) have ultra-low latency requirements which means the response time of security applications should be very low. Higher response time (to detect anomalies) is an opportunity for attackers or hackers as it gives reasonable time to them to access the network and perform malicious activities [2]. 2) Some nodes (for example IoT devices) are cheap and designed for a specific application and do not support resource and power-hungry encryption algorithms for security purposes [3]. 3) With the emergence of SDN and NFV the core components in 5G can be deployed on the edge networks which creates a new threat surface due to the difficulty of manage distributed packet core networks [4]. 4) 5G networks are heterogeneous and security process are expected to ensure a secure connectivity between 3G, LTE, and 5G network elements [5].

Overall, all these unique characteristics of NGNs have increased the threat landscape. Particularly, collecting, processing, and analyzing large amounts of heterogeneous data streams and huge number of network connections for network traffic anomaly detection in real-time and autonomic way is a great challenge [6]. Given all this the security is an open issue and Intrusion Detection Systems (IDSs) are an effective solution [7]. Although encryption techniques can be used to mitigate threats as it increases privacy level however hinders network operators' visibility of traffic analyses and hence the administrator can lose the ability to distinguish malicious or benign traffic [5]. The existing solutions work using Deep Packet Inspection (DPI) tools (Snort [8] and Suricata [9]) that have poor performance when encrypted traffic exists. For example, Snort [8] is a popular Deep Packet Inspection (DPI) tool that can work effectively on transmission rates up to 1 Gbps (it starts to discard packets from 1.5 Gbps [10] while 5G delivers transmission rates up to 10 Gbps.)

Recently, the Deep Learning (DL) models are being used in different cyber security applications in IDSs. The anomaly detection using DL is not a new area of research and a variety of methods have been explored [11], [12]. However, the existing methods have some common issues such as low accuracy levels, sub-optimal model design, unrealistically high accuracy level due to lack of model generalization, use of decade old and simplest data sets, etc. Further, the traditional IDSs need to decrypt the traffic and then analyze it, and again re-encrypt it. This is a resource and time-consuming task, especially for analyzing large scale networks that generate huge amount of heterogeneous data [13]. Also it is not feasible for applications that provide end-to-end encryption (e.g., WhatsApp) as the network provider cannot decrypt their traffic.

Therefore, the conventional security approaches lack compatibility with modern network designs and are not effective in NGNs settings, hence a novel deep learning-based IDS suitable for high-speed NGNs (such as 5G applications) is required. As the threat landscape is increasing the existing

Manuscript received 11 April 2022; revised 13 September 2022, 31 October 2022, and 28 November 2022; accepted 13 December 2022. Date of publication 2 January 2023; date of current version 6 January 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. George Loukas. *(Corresponding author: Keshav Sood.)*

The authors are with the Centre of Cyber Security Research and Innovation (CSRI), School of Information Technology, Deakin University, Geelong, VIC 3220, Australia (e-mail: keshav.sood@deakin.edu.au; m.nosouhi@deakin.edu.au; nguyendinh@deakin.edu.au; frank.jiang@deakin.edu.au; morshed.chowdhury@deakin.edu.au; robin.doss@deakin.edu.au).

Digital Object Identifier 10.1109/TIFS.2022.3233777

DL based IDS solutions are forced to evolve continuously to adopt new ways of network traffic anomaly detection. New methodologies are urgently required to detect anomalies eventually efficiently, automatically, and seamlessly for better and secure network management [6], [14]. To show the current state-of-the-art, firstly, we present, in Table I, the key existing research works in IDS domain those make use of deep learning for anomaly detection both in 5G and non-5G sector.

We note some common gaps in the recent works which are shown in Table I and discussed in detail in Section II. These gaps are; a) authors have used two decade old data sets which lacks the 5G network features as well as the modern footprint attack style, lack of modern 5G traffic features/scenarios, and the training and testing sets have different distributions, b) the existing schemes are not rigorously tested on real 5G test beds using real 5G data sets, nor they have fully proposed the methods to reduce the feature vectors size as it is vital to alleviate the computational load and to improve the performance of the IDSs, and c) the existing approaches may lack the compatibility with ETSI-NFV (European Telecommunications Standards Institute-NFV) standards architecture as none of the proposed design is evaluated to capture the difficulties of deploying their proposals in real-time. Motivated from this, our proposal fully accounts these gaps/requirements into consideration. We propose an intrusion detection scheme with Dimensionality Reduction (DR) for NGNs.

The contributions of this paper are as follows.

1) We propose a two–stage Deep Learning–based IDS architecture with dimensionality reduction DR) for network traffic anomaly detection (AD) at the edge of 5G networks. We use Deep Autoencoder to reduce the size of feature vectors, and DNN model to facilitate the anomaly detection process. It can be argued that after feature extraction by Anomaly Detection module there is no need of Dimensionality Reduction at the edge of 5G networks. Therefore, to justify the need of DR at the edge (and this contribution) the proposal is evaluated with and without DR module to prove that DR module is critical since it is a trade-off between the accuracy and the processing time.
2) To evaluate the performance of the proposal we have used OMNET++ 5G simulator. We have used the UNSW-NB15 dataset for evaluation. This data set is built using a standard wireless computer network; hence it can be used to simulate the device-to-device (D2D) connections that 5G supports.
3) Our approach is uniquely designed to be compatible with the standard ETSI-NFV architecture. The scheme (or module) can be implemented as the pluggable Virtual Network Functions (VNFs) on any 5G network slice having its own independent network orchestration module. Both the DR and AD modules are integrated into the ETSI-NFV architecture as separate VNFs that are controlled by standard VNF managers as an independent management orchestration of each slice (Fig. 1). This design and implementation show a great step towards providing an efficient and effective way to empower networks to detect attacks in 5G.
4) A Proof of Concept (PoC) using ETSI OSM-MANO test bed is given to show how the proposed approach would fit in with a real-life. This is to capture the difficulties of deploying the VNFs involved in the process and give a sense of realism to the solution. This contribution presents relevant aspects of deployment of the approach and evaluation results show its performance.

We have compared our proposal with other approaches of anomaly detection in 5G networks. The results from our approach are considerably better than the recent existing approaches. This confirms the need of new IDSs with DR in NGNs, as we have presented and evaluated in this paper. Also, it testifies that the proposal is contributing a new knowledge in this domain. The rest of the paper is organized as follows. The related work is discussed in Section II. The proposed architecture is discussed in Section III. The performance evaluation results are shown in Section IV. In Section V we have analyzed the complexity and security of the proposal. The PoC of the integration of this with ETSI-NFV is given in Section VI. Finally, Section VII summarizes the paper.

II. Related Work

In NGNs the threats can be categorized based on the network part (core network, access network, transport network, and inter-connected network). For example, core network is vulnerable to IP based attacks, access network is vulnerable to man-in-middle attack, Software-defined networks (SDNs) involvement has triggered the concept of authentication of user node operating between multiple network slices, DDoS can be performed on SDN enabled 5G networks to gain control over network elements and so on. Furthermore, due to the ultra-low latency requirement of applications in heterogeneous NGNs enabled networks, the anomaly detection in real-time is a key challenge [11], [12]. Recently, DL methods have become a popular solution for anomaly detection in 5G networks. This is because of their ability to learn nonlinear and complex patterns in data that makes them an ideal tool to distinguish anomalies (e.g., malicious traffic) from normal traffic flow. In this section, we synthesize the previous research done in the field of traffic anomaly detection in 5G networks.

Authors in [15] used a feedforward neural network (FNN) for multi-class anomaly classification of DDoS attack data. In this multi-class classification, authors have considered four categories of traffic. Firstly, we note that the proposed approach yielded sub-par performance for classifying traffic in certain attack subcategories (e.g., DoS over HTTP and DDoS over HTTP). It means the approach cannot handle large volume of attacks or large-scale attack traffic is not reflected in the feature set they used for their experiments. Secondly, the SVM classifier they have used process data in batches, means the classier needs substantial improvements to operate in real-time anomaly detection. With these two reservations we observe that the approach is not fully suitable in real-time and at very high speed of traffic for anomaly detection.

In [16] authors used autoencoder for anomaly detection for edge 3GPP networks. However, the training of the model is conducted offline which means detecting anomalies in time series data is not feasible. This is simply due to the limited

TABLE I
SUMMARY AND A HIGH LEVEL COMPARISON OF THE KEY EXISTING WORKS IN INTRUSION DETECTION SYSTEM

Year/Ref.	Domain: 5G (✓), Non-5G (X), Others specified	Model Used	Data Set	Edge Network	Dimensionality Reduction	ETSI-NFV Compatibility	OSM MANO Orchestration: Independent to each slice (✓), Centralized (X)
2021, [15]	X, IoT	Feed-forward neural networks + SVM	BoT-IoT	X	X	X	X
2021, [16]	3GPP mobile network, IoT	Autoencoder	Self-generated synthetic data	X	X	X	X
2021, [17]	X, IoT	Grid LSTM	Power, loop, and climate data set	✓	X	X	X
2021, [18]	X, cellular networks	SVM, SVR, LSTM	Milan, Dakar	X	X	X	X
2020, [19]	X, cloud systems	autoencoder + SVM	NSL-KDD, KDD Cup 99	X	X	X	X
2018, [20]	X	autoencoder + SVM	NSL-KDD, KDD99	X	X	X	X
2019, [6]	✓	DNN, LSTM	CTU-13	✓	X	X	X
2019, [21]	✓	Random Forest	-	X	X	X	X
2019, [22]	✓	DNN	AWID-CLS-R	X	X	X	X
2020, [5]	✓	CNN	CIC-IDS2018	X	X	X	X
2021, [23]	✓	CNN	KDD'99, DARPA'98	✓	X	X	X
Our Scheme	✓	Autoencoder + DNN Classifier	UNSW-NB15	✓	✓	✓	✓

computational and storage capacities of edge devices. Unfortunately, the data set used in this work is non cellular/non-5G for the validation of the proposal. In [17] a unified data driven networking system is proposed to analyze large volume time series data. The anomaly detection accuracy is high, but the used dataset provides no information on the specific types of anomalies and therefore cannot label the training data set by their anomaly types. The used data set is not collected from 5G nodes. The work in [18] used a spatio-temporal mechanism for anomaly detection. The proposed framework works in two stages and learns temporal and spatial contexts separately. Following this the learned representations are used to detect spatio-temporal anomalies. They used one class SVM classifier to extract the spatial context and used the SVR models to extract temporal anomalies. We note that the approach works well with IoT sensor data that arrives in time series in order to find anomalies. However, instead of standard convolutional Neural Networks authors have used the supervised machine learning models. The authors have not calculated the overhead and the complexity of the approach which we have shown in our work in Section V.

Further, the authors of [19] have explored the benefits of stacked contractive autoencoder and SVM classifier approaches to realize IDS's performance. The model is compared with the baseline autoencoder models using well known, but two decades old, security datasets. We note that the trade-off between model training time and feature reduction is not explored. In this regard, in [20], self-taught learning based SVM model is used for anomaly detection using KDD99 data set. Following this, in [6], authors proposed a mobile edge computing architecture for anomaly detection using policies. The centralized network orchestration handles the complete workflow and control of the proposed framework. Authors in [21] proposed a new architecture to detect flooding DDoS attacks at sources side. The computation is shifted to control plane which may add additional end-to-end latency. Moreover, loss precision metric highly affected when a large variety of attacks are launched. The work in [22] proposed a deep learning-based IDS. The accuracy of the model needs significant improvement as the approach is limited to detect certain attacks only.

The authors of [5] have proposed a Software Defined Security (SDS) framework as a tool to provide flexible and scalable network security solutions. The proposed framework considers the design of Convolutional Neural Networks (CNNs) using Neural Architecture Search (NAS) for traffic anomaly detection in 5G networks. The authors have tested their assumption using normal and anomalous network flows collected from a simulated environment. However, the proposed approach still needs to discuss and address the issues such as speed or training time of model, large volumes and high dimensional traffic flows vs. model training time, and the high level of traffic heterogeneity in 5G networks. Moreover, they have not discussed how the proposed approach can be implemented in real-world settings. Nevertheless, this is an early work in 5G security and needs significant evaluation as the impact of the proposal on real 5G settings and deployment issues during feature extraction and processing are not shown.

Very recently in [23], authors focus on enhancing network security in 5G by integrating SDN/NFV-based Service Function Chaining and Machine Learning. Authors have proposed a framework for efficient provisioning of value-added services at mobile edge computing clouds. We note that the proof of concept is not evaluated on any 5G test bed. This does not show any challenges of the prototype that this would have while deployment to give a sense of realism of the solution. The authors of [24] introduce a self-adaptive anomaly detection and cyber-defense system for 5G mobile networks in which traffic fluctuations are expected. The proposed system works based on a two-level Deep Learning model which makes the detection process more efficient and optimized. This is because the system adapts network computational resources based on traffic fluctuations. However, the two levels of DL models in the proposed system have not been trained using a real dataset and the detection accuracy is not evaluated.

In [25], the authors investigate the security issues of the 5G network and introduce a DL–based model to build a secure and reliable Network Slicing framework. Their target is to predict, detect, and eliminate security threats by analyzing incoming traffic before they can reach and damage the core of a 5G network. For performance evaluation, the authors have used volume–based flooding and spoofing attacks in two different scenarios. However, in the proposed system, the training process is not performed in real–time, which prevents the model from achieving high levels of detection accuracy for scenarios in which very dynamic traffic is expected [25].

We have also synthesized some recently published works. We consider [26] in which authors propose a Robust Transformer-based Intrusion Detection System (RTIDS) utilizes positional embedding technique to associate sequential information between features. Following this, authors have used a variant stacked encoder-decoder neural network for model training evaluation. They have evaluated the approach, using CICDDoS2019 dataset, with the baselines machine learning models, i.e., support vector machines (SVM), and deep learning algorithms that include recurrent neural network (RNN), fuzzy neural network (FNN), and long short-term memory (LSTM). These models are also used in the works listed in Table I. Although authors in [26] obtained the highest accuracy 98.5%, we emphasize that using the CICDDoS2019 dataset to validate the proposed system may not be appropriate since the dataset is not related to the 5G network. The CICDDoS2019 dataset is collected from a small group of victim devices which are all under the same LAN [27], [28]. Also, according to the CICDDoS2019 dataset,[1] the extracted features are 80 numerical values, such as duration, packet length, bit rates, of which sizes are negligible when compared with the payloads. Not to mention that the single sample containing 80 features is extracted from plenty of packets in a period. Finally, the training time of the algorithms for intrusion detection is high which is not effective to significantly reduce the damage may caused by anomalous events.

We have presented a good overview in the introduction section also about the recent works done in this domain. Other than these works, in this regard, a comprehensive survey has been done in [29] that presents a literature review of the applications of DL–based methods in network security domain. The authors have studied different DL models such as Deep Autoencoders, DBNs, Restricted Boltzmann Machines (RBMs), RNNs, CNNs, etc. For each DL method, they present a brief tutorial–style description along with the discussion about how each method can be used in cyber security applications. They also cover a broad range of attacks such as false data injection, spams, network intrusions, malware, and malicious domain names used by botnets. Another comprehensive survey has been done in [30] which studies the applications of machine learning models in securing software-defined networks (SDN) enabled networks. The authors have classified the relevant research works into machine learning–based techniques and IDS frameworks for SDN.

[1] https://github.com/ahlashkari/CICFlowMeter/blob/master/ReadMe.txt

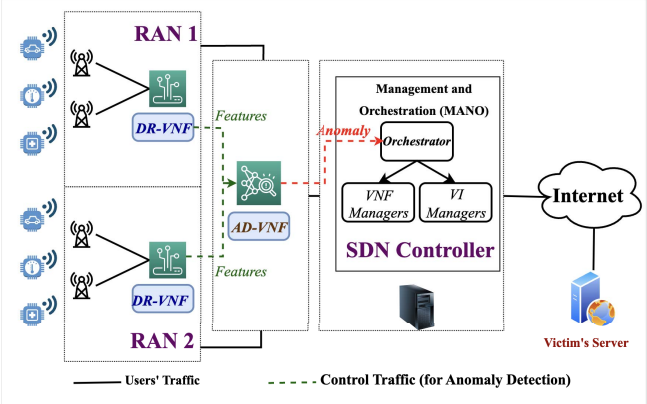

Fig. 1. The proposed architecture. The Dimensionality Reduction (DR) and Anomaly Detection (AD) modules are integrated into the standard architecture (ETSI-NFV) of a 5G enabled SDN network as two pluggable VNFs that are shown as DR–VNF and AD–VNF. They are managed and orchestrated through standard VNF managers.

Recently, in [7] authors proposed a federated learning based IDS. They used the CICIDS2017 dataset. Note that the proposed architecture is evaluated on dataset which is not related to the cellular networks. The testbed architecture has only 4 clients. A promising way to aggregate more information is at 5G slice which has not been considered. In contrast to the existing works, we emphasize that our proposed approach is compatible with the standard ETSI architecture and integrally accounts the accuracy and real time anomaly detection constraints into consideration. The existing works have taken these issues but separately. The results obtained from our approach are approximate to real scenarios. The comparison of our approach with existing works validates our contribution.

III. The Proposed Architecture

In this section, we present and discuss the proposed architecture and its functionality. Fig.1 shows the proposed architecture which consists of two separate modules that are implemented at Radio Access Network (RAN) and edge segments of 5G networks. These modules are defined in the form of two pluggable Virtual Network Functions (VNFs), i.e., DR–VNF and AD–VNF, that can be easily integrated into the standard architecture of ETSI NFV, as shown in Fig. 2. In this model, these VNFs are controlled and orchestrated by VNF managers. This makes the proposed approach fully compatible with the standard 5G network architecture shown in Fig. 2. We assume that the proposed IDS module is an integral part of 5G Authentication Server Function (AUSF), as seen in Fig. 3. Here each slice has dedicated virtual resources (VMs) as well as its own network management orchestration module. End nodes in each distributed underlying networks (divided in zones) are only connected to one dedicated network slice at one time. Therefore, the proposed strategy is a part of each orchestration module.

In Fig. 1, the proposed approach consists of two separate modules. In the first module, i.e., DR, we employ a Deep Autoencoder (DAE) to reduce the size of traffic feature vectors. Generally, there may exist more than 100 different

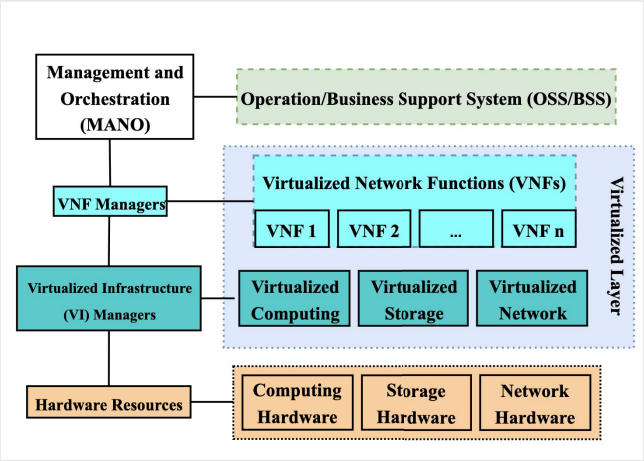

Fig. 2. ETSI NFV standard 5G architecture.

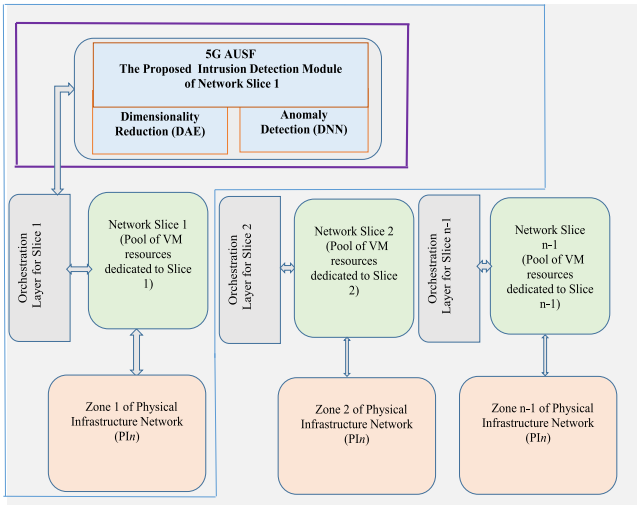

Fig. 3. A high-level framework of the proposed slice-specific IDS approach's integration with ETSI-NFV MANO architecture.

features to be analyzed by a traffic anomaly detection module before making any classification decision on a specific traffic flow [10], [31]. Using a DAE, the size of feature vectors can be significantly reduced without any considerable loss in accuracy [31], [32]. This results in a significant increase in the efficiency of the anomaly detection module in terms of processing speed, anomaly detection time, online model (for anomaly detection) training time, CPU utilization, and memory usage. In the second module, i.e. AD, we employ a Deep Neural Network classifier (DNN) [31], [33] that takes the reduced sized feature vectors from the DR module and detects anomalies in the traffic flows.

We employ the DR and AD modules in RAN segment and Edge segments of the network, respectively as seen in Fig. 1. This enables the DR module to function in real–time with the practical transmission rates delivered by a 5G network. This is because there is always a lower number of traffic flows in the RAN segment compared to the other segments of the network. However, the AD module at the edge level

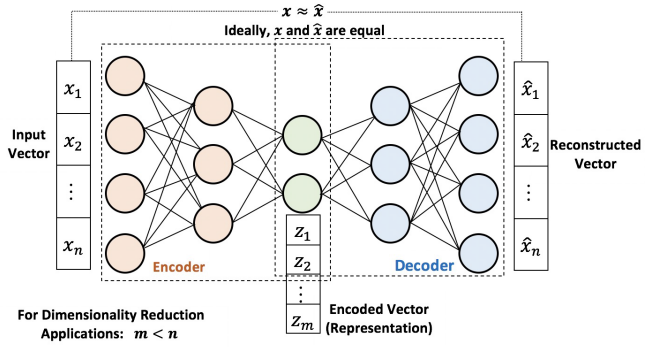

Fig. 4. General architecture of a deep autoencoder (DAE).

has access to the traffic flows of multiple RANs. This creates a high level of diversity in the data since it can analyze the traffic flows of a wide range of users which results in making more accurate classification decisions. Moreover, employing the detection mechanism at the edge segment of the network results in early detection of malicious traffic, i.e., before the rest of the network infrastructure and resources are affected.

In our experiments, we obtain a reduction factor of 61% which results in a significant reduction in the size of feature vectors. This is obtained at the cost of a 6.5% loss in similarity. We show that this level of loss in similarity does not have any considerable effect in the accuracy of the anomaly detection process. Finally, regarding accuracy, the proposed approach detects malicious traffic very accurately. In our experiments, we obtain a detection accuracy of 98.7%. In the rest of this section, we present the functionality of DR–VNF and AD–VNF modules separately.

A. DR–VNF Module

We employ the DR–VNF module to reduce the size of feature vectors before they are applied to the AD–VNF module for detection of malicious traffic. This significantly increases efficiency of the anomaly detection process (performed by AD–VNF) in terms of speed, storage, and processing overheads. For example, the feature vectors may typically have 100 different features such as source and destination IP addresses and ports, timestamps, flow duration, protocol, TCP flags, total length, flow bytes/s, flow packet/sec, etc. It would be perfect if the vectors could be converted to their equivalent vectors of size 16 without any considerable loss in the similarity. Indeed, it is much more efficient for AD–VNF to analyze 16×1 vectors than 100×1. Deep Autoencoders (DAE) are an effective tool to attain this target, i.e., to reduce the size of feature vectors without causing any considerable damage to the characteristics and statistics features of data.

DAEs are a type of Artificial Neural Networks (ANN) that are used to perform efficient data encodings in an unsupervised manner. This is mostly done for dimensionality reduction purposes and performed by training the network of neurons to remove noise from the input data. To do this, a DAE employs two separate networks of neurons, i.e. an encoder and a decoder as shown in Fig. 4. The encoder part performs the dimensionality reduction process while the decoder is only

employed in the training (learning) phase to reconstruct the input vector from the encoded vector. In fact, the reconstructed vector is only used to minimize the error (dissimilarity) between the input and the encoded vectors during the training phase. In other words, a DAE learns the optimum values for the weights and biases of neurons such that the error is minimized. The number of neurons at the output layer of the decoder network is equal to the number of neurons at the input layer of the encoder because the reconstructed vector should have the same dimension as the input vector. After training the network of neurons, the encoder part can be used to obtain the encoded vector (representation) for any real–time application [32], [34]. In the following, we present the mathematical model of the DR–VNF module.

Assume $\mathbf{X} \in \phi$ is the set of training vectors $\mathbf{x}^i = \{x_1, x_2, \ldots, x_n\}$ where ϕ is an N–dimensional feature spaces. If $f_\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $f'_\beta : \mathbb{R}^m \rightarrow \mathbb{R}^n$ are the encoding and decoding functions, respectively, we have

$$\mathbf{z}^i = f_\alpha(\mathbf{x}^i) \tag{1}$$

$$\hat{\mathbf{x}}^i = f'_\beta(\mathbf{z}^i) = f'_\beta \circ f_\alpha(\mathbf{x}^i) \quad , \tag{2}$$

where $\mathbf{z}^i = \{z_1, z_2, \ldots, z_m\}$ and $\hat{\mathbf{x}}^i = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$ are the representation (encoded) and reconstructed (decoded) vectors, respectively, associated with the input vector $\mathbf{x}^i$. Note that $\mathbf{Z} \in \theta$ shows the set of encoded vectors where θ is an m–dimensional feature spaces ($m < n$). $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are parameters of the encoder and decoder functions, respectively.

Now, assume there is an unknown probability distribution ρ defined over ϕ. Given Δ as a dissimilarity (error) function (such as Mean–Absolute Error (MAE), Mean–Squared Error (MSE), Euclidean Distance, etc.), the relevant *autoencoder problem* is to find $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ such that the expected value of the dissimilarity function Δ is minimized, i.e.

$$\underset{(\boldsymbol{\alpha},\boldsymbol{\beta})}{min}\, \mathbb{E}_{(\mathbf{x}^i,\hat{\mathbf{x}}^i)\sim\rho}(\Delta(\mathbf{x}^i, \hat{\mathbf{x}}^i)) = \underset{(\boldsymbol{\alpha},\boldsymbol{\beta})}{min}\, \mathbb{E}(\Delta(\mathbf{x}_i, f'_\theta \circ f_\phi(\mathbf{x}_i))) \tag{3}$$

Because the probability distribution ρ is unknown, it is not feasible to obtain the expected value of the dissimilarity function. Thus, we limit the autoencoder problem to the space of the training vectors, i.e.

$$\underset{(\boldsymbol{\alpha},\boldsymbol{\beta})}{min}\, \Delta(\mathbf{x}^i, \hat{\mathbf{x}}^i) = \underset{(\boldsymbol{\alpha},\boldsymbol{\beta})}{min}\, \sum_{j=1}^{n} \Delta(\mathbf{x}_j, f'_\theta \circ f_\phi(\mathbf{x}_j)), \tag{4}$$

The above autoencoder problem is solved for every $\mathbf{x}^i$ and $\hat{\mathbf{x}}^i$ in $\mathbf{X}$ and $\hat{\mathbf{X}}$ (respectively), i.e., all the vectors in the training dataset are learnt. Different types of autoencoders can be derived from this general model depending on the choice of functions f_α, f'_β, and the dissimilarity function Δ. Moreover, applying additional constraints such as regularization can change the type of autoencoder. For example, if MSE is selected as the dissimilarity function, for the autoencoder problem we have

$$\underset{(\boldsymbol{\alpha},\boldsymbol{\beta})}{min}\, \frac{1}{n}\sum_{j=1}^{n}(\mathbf{x}_j - \hat{\mathbf{x}}^j)^2 = \underset{(\boldsymbol{\alpha},\boldsymbol{\beta})}{min}\, \frac{1}{n}\sum_{j=1}^{n}(\mathbf{x}_j - f'_\theta \circ f_\phi(\mathbf{x}_j))^2 \tag{5}$$

To solve the above autoencoder problem, gradient–based optimization approach is a popular and effective method to choose [35], [36], [37]. There exists several versions of gradient–based optimization algorithms. For example, in Batch Gradient Descent (BGD), the gradients of all samples are calculated at first. Then, based on the obtained gradients, the neural network parameters are updated [35]. However, it is used in offline training applications in which the whole set of the training dataset is available. However, in online (real–time) applications, training samples may become available after the model is employed. On the other hand, Stochastic Gradient Descent (SGD) can be used in online training applications. Each time, it updates the parameters using an instant training sample [34], [35]. In other words, in BGD, all the training samples must be learnt before a single update is done on the network parameters. However, in SGD, one or a subset of the training samples can be learnt in order to update the network parameters. This makes SGD an efficient optimization algorithm. Specifically, in high–dimensional optimization problems, SGD performs very efficient in terms of speed and computational overhead [35].

Since we build the autoencoder for an online (real–time) application with a huge number of data points, we solve the autoencoder problem using the SGD approach. Therefore, we have

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} - \epsilon^{(k)} \nabla_{\boldsymbol{\alpha}} \Delta_i(\boldsymbol{\alpha}^{(k)}) \tag{6}$$

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} - \epsilon^{(k)} \nabla_{\boldsymbol{\beta}} \Delta_i(\boldsymbol{\alpha}^{(k)}) \tag{7}$$

where $\nabla_{\boldsymbol{\alpha}} \Delta_i(\boldsymbol{\alpha}^{(k)})$ and $\nabla_{\boldsymbol{\beta}} \Delta_i(\boldsymbol{\alpha}^{(k)})$ are the gradients taken using $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively (considering a training sample $\mathbf{x}^i$). ϵ is the learning rate that is used to adjust the speed of convergence. It determines the size of steps that are taken to reach the optimum parameters. Using larger values for ϵ results in faster training but at the risk of missing the optimum values (loss in accuracy). On the other hand, a smaller ϵ makes the convergence of algorithm slower. When the optimization problem is solved, parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are obtained. This means that the autoencoder model has been built and vectors $\mathbf{z}^i = f_{\boldsymbol{\alpha}}(\mathbf{x}^i)$ can be obtained as the encoded (representation) vectors.

B. AD–VNF Module

In our proposed approach, the AD–VNF module is employed at the edge segment of the 5G network. Thus, it covers all the RANs in that edge segment. This module performs the final detection of malicious traffic. It receives the encoded feature vectors from the DR–VNF modules in that segment of the network. Because the feature vectors have been already downsized by the DR–VNFs at the RAN segment, the AD–VNF module needs to analyze vectors with smaller dimensions. Thus, we expect to see a significant improvement in the performance of the AD–VNF module in terms of speed, computation overhead, and memory usage.

We employ a Deep Neural Network (DNN) in this module to perform the anomaly detection. Recently, Deep Learning–based models have been effectively being used in many tasks and applications such as image processing,

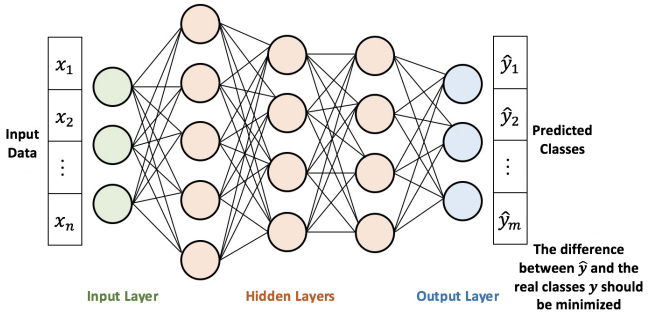

Fig. 5. General architecture of a deep neural network (DNN). The parameters of neurons (weights and biases) are calculated in such a way that the error between predicted classes and real classes is minimized.

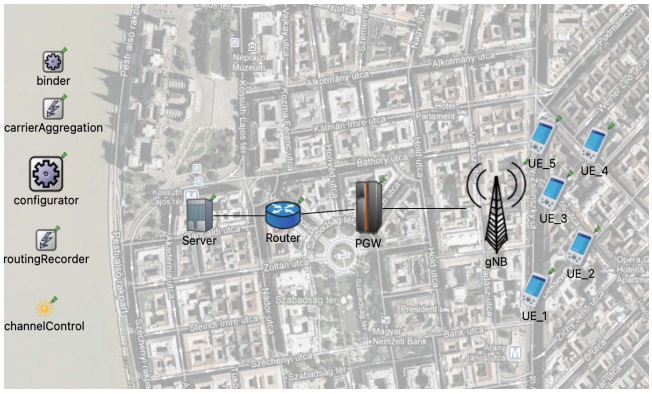

Fig. 6. The network topology implemented in OMNET++.

speech recognition, anomaly detection, etc. They can automatically learn complex nonlinear patterns of data and make highly–accurate classifications and decisions. A typical DNN consists of three different parts as shown in Fig. 5. The first part is called the input layer which is an interface to the input vector. The size of this layer should be the same as the dimension of input vector. The second part is called hidden layers. This includes all the layers between the input layer and the last part which is called the output layer. The size of output layer is selected based on the number of target classes. For example, in case of a binary classification task, the size of output layer is 2. Each layer consists of a number of neurons which indicates the size of that layer. A neuron in a specific layer receives several signals from the neurons in the previous layer (the output of previous layer) and performs a typically simple calculation on the signals, e.g. a weighted sum of the input signals followed by a nonlinear activation. Thus, as a whole network, the neurons cooperatively perform a complex nonlinear mapping of the input vector to the output vector. This is learned through the error back-propagation technique in which the optimum values for weights of each neuron are calculated.

In the development of a DNN, it is very important to select the right number of hidden layers. Unlike Shallow Neural Networks that have one hidden layer, DNNs can have several hidden layers. For example, Google LeNet model developed for image recognition has 22 hidden layers [38]. Every layer indicates a deeper level of knowledge. As a result, a DNN with four layers, for example, can learn more complex patterns and features of data than a DNN with two layers. Thus, selecting a high number of hidden layers may improve the accuracy of detection. However, if the number of hidden layers increases, more neurons in the network will be created. This increases the processing time, memory usage, and computation overhead of the model since the network parameters (weights and biases) should be computed for a higher number of neurons (equivalently, a more complex optimization problem should be solved). In fact, this indicates the trade–off between accuracy and speed of the model (and other performance metrics such as memory usage and CPU utilization). Moreover, the size of hidden layers is another important parameter that directly affects the DNN performance in terms of accuracy, speed, etc.

To fully develop a DNN, an optimization problem must be solved. Consider a DNN that maps the input vectors $\mathbf{x}^i = \{x_1, x_2, \ldots, x_n\}$ to a set of target classes $\mathbf{y}^i = \{y_1, y_2, \ldots, y_m\}$. The optimization problem is solved by obtaining the network parameters $\mathbf{w}$ and $\mathbf{b}$ (wights and biases of neurons) such that the following cost function is minimized, i.e. $\underset{(\boldsymbol{w},\boldsymbol{b})}{min}\ \mathbb{C}(\mathbf{y}^i, \hat{\mathbf{y}^i})$ in which

$$\hat{\mathbf{y}^i} = \sum_{j=1}^{n} \rho(\mathbf{w}_j^T \mathbf{x}_j + \mathbf{b}_j) \tag{8}$$

Similar to the autoencoder problem that was discussed in the previous subsection, this optimization problem can be solved using the SGD approach. Once the optimization problem is solved, the network parameters $\mathbf{w}$ and $\mathbf{b}$ are determined for each layer of the DNN model. In our experiments, we used several activation functions such as ReLu, Sigmoid, and tanh to evaluate the effect of activation functions on the detection accuracy. Regarding the cost function, we used the MAE and MSE approaches in the SGD algorithm (see the results in next section).

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed approach. *A)* We first present the experimental set up details as well as discuss the features of the dataset used in the experiments. *B)* Then, we present the experimental results for the dimensionality reduction and anomaly reduction modules in two separate subsections. *C)* To justify the need of DR at the edge, the proposal is evaluated with and without DR module. *D)* Following this, we show the comparison results of our proposal with recent key approaches. *E)* Finally, a PoC using ETSI OSM MANO test bed is given to show how the proposed approach would fit in with a real-life MANO to capture the difficulties of deploying the VNFs involved in process and give a sense of realism to the solution. This is to present the relevant aspects of deployment of the approach and evaluation results show its performance.

A. Setup

For our experiments we use Intel Core i5, 2.3GHz CPU with 8GB of RAM. We performed a proof–of–concept

TABLE II
DESCRIPTION OF THE MAIN FEATURES OF THE UNSW–NB15 DATASET

Feature	Data Type	Description	Category	Feature	Data Type	Description	Category
srcip	Nominal	Source IP address	Flow	*dbytes*	Integer	Destination to source bytes	Basic
sport	Integer	Source port number	Flow	*sintpkt*	Float	Source inter-packet arrival time	Time
dstip	Nominal	Destination IP address	Flow	*dintpkt*	Float	Destination inter-packet arrival time	Time
dsport	Integer	Destination port number	Flow	*smeansz*	Integer	Source flow packet size	Content
sbytes	Integer	Source to destination bytes	Basic	*dmeansz*	Integer	Destination flow packet size	Content

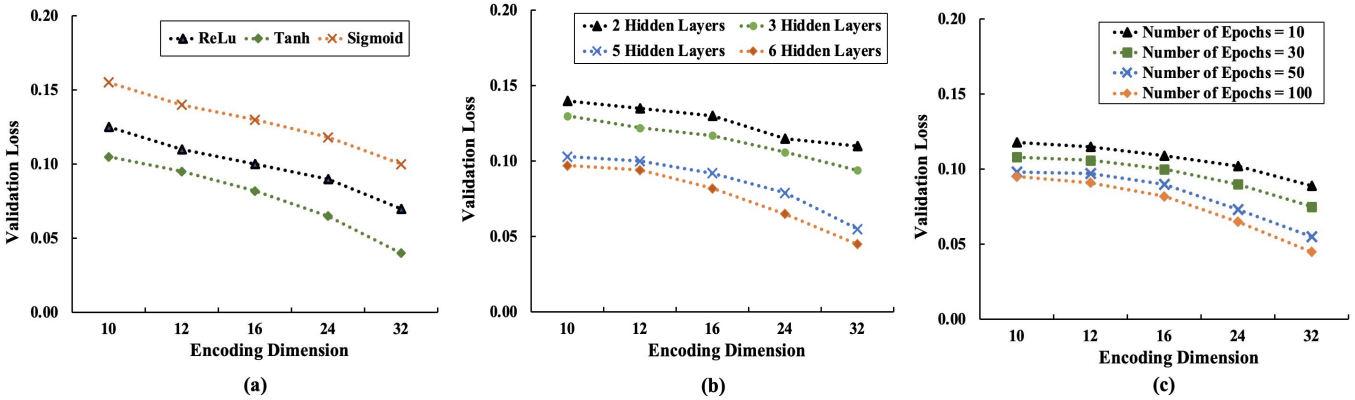

Fig. 7. Validation loss of the dimensionality reduction module (DR–VNF). This metric indicates the level of dissimilarity between the input feature vector and the associated reduced–size vector (the encoded vector). Note, the vectors become more dissimilar as the dimension of the encoded vector decreases (a) The effect of activation functions employed in the autoencoder on dissimilarity. Tanh showed better performance than ReLu and Sigmoid in terms of validation loss (b) A higher number of hidden layers employed in the autoencoder results in more similarity between the input and the encoded vectors. (c) The effect of number of epochs on validation loss.

implementation of the proposed architecture in OMNET++ [39]. To do this, we used Simu5G [40] (which is a simulator of 5G RAN and core network) and performed it in the OMNET++ environment integrated with INET Framework 4.3.5. Fig. 6 shows the implemented network topology. In addition, to evaluate performance of the DR and AD modules, we used UNSW–NB15 dataset [41] for our experiments. This dataset has been generated by capturing 100 GB of raw traffic (using tcpdump tool). It includes different types of attacks such as Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. The training and test datasets consisted of 175,341 and 82,332 records, respectively. Each data entry has 49 different traffic features (see Table II for a list of selected features). In our experiments, we targeted on the detection of Analysis, Exploits, Backdoor, and DoS attacks that contain 2000, 33393, 1746, and 12264, respectively with 56,000 number of normal observations. We performed some data pre–processing tasks on the dataset to prepare it for analysis. For example, features like source and destination IP addresses contain non–numeric characters that are not accepted by the learning models in Python. Thus, these features should be converted to numeric–only fields. Moreover, we converted the categorical features to numeric using the one–hot encoding technique. The data pre–processing tasks were performed using the pandas and numpy libraries in Python.

B. Dimensionality Reduction

In this subsection, we present the results of our experiments for the dimensionality reduction module. Fig. 7 shows the effect of encoding dimension (the size of the encoded vectors) on the validation loss. This metric indicates the level of dissimilarity between the input feature vector and the associated reduced—size vector (the encoded vector). As the figures show, if we reduce the size of encoded vectors (encoding dimension), the level of dissimilarity increases. For example, at encoding dimension 32 (which indicates 48% dimensionality reduction regarding 62 as the size of input vectors) we obtain the similarity level of 94%. However, for the encoding dimension 16 (reduction factor of 74%), the vectors became 90.5% similar. We performed the experiments using three different activation functions, i.e. ReLu, Tanh, and Sigmoid. The best results were obtained using the Tanh activation function. Fig. 7(a) shows the results.

We also changed the number of hidden layers employed in the autoencoder. As we expected, the autoencoder with 6 hidden layers showed better performance (in terms of dissimilarity metric) than the autoencoder with 5, 3, and 2 hidden layers (Fig. 7(b)). This is because each layer provides a deeper level of knowledge. However, this resulted in more time taken for the training of the network shown in Fig. 8(a). Regarding the effect of number of epochs on validation loss, the autoencoder performed better when we increased the number of epochs in the training phase (Fig. 7(c)). The reason is that by applying a higher number of epochs, the model has more chance to learn from the training data. However, applying a higher number of epochs may result in over-fitting of the model. To avoid over-fitting, in our implementation, we enabled the *EarlyStopping()* feature of *Keras* library in Python. Using this feature, the training procedure will be automatically stopped as soon as the model becomes over-fitted. This is checked through the metric of validation loss.

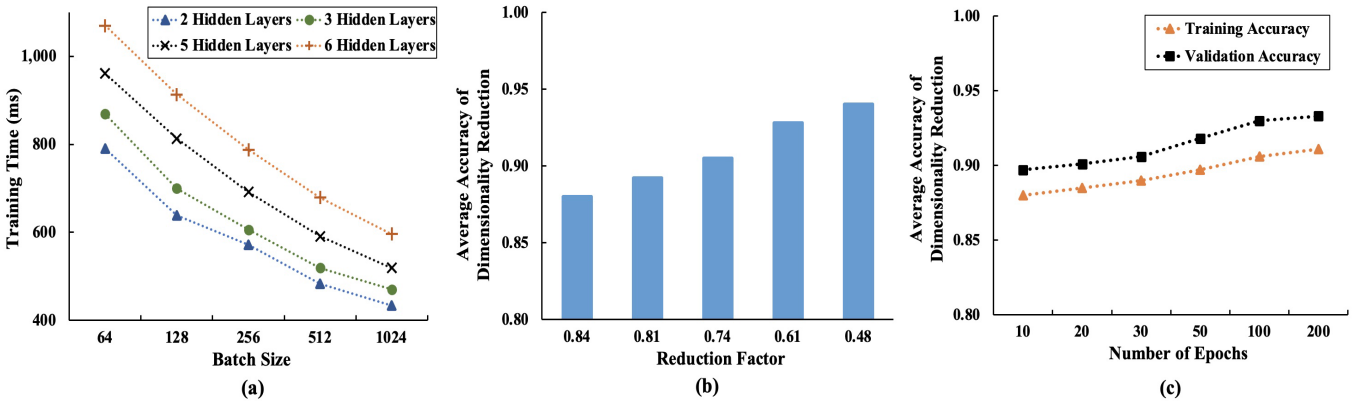

Fig. 8. Training time and accuracy of the autoencoder employed in the DR-VNF module. (a) Employing a higher number of hidden layers results in more time taken for the training phase. However, selecting a larger batch size shortens the training phase. (b) The effect of the reduction factor on accuracy of the autoencoder. (c) Applying a higher number of epochs increases the accuracy, however, overfitting should be taken into consideration.

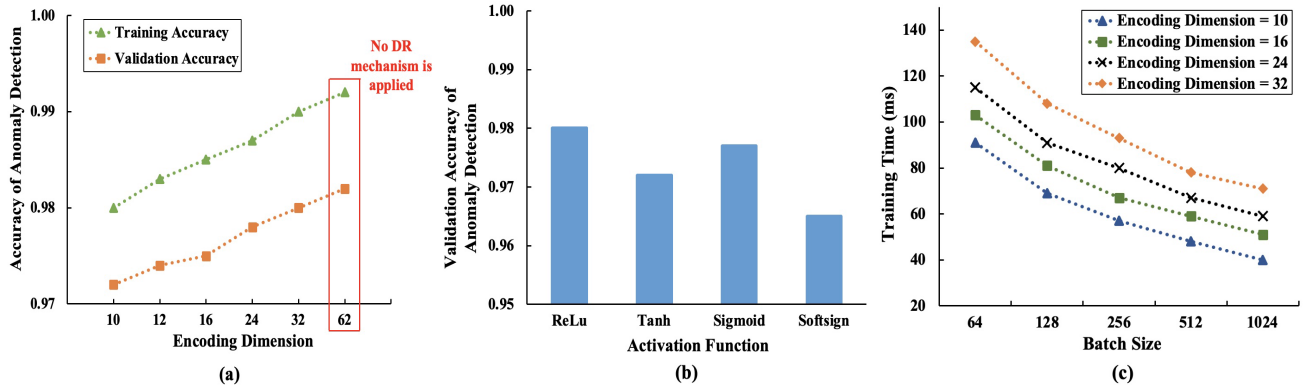

Fig. 9. (a) Detection accuracy of the proposed AD–VNF module for different encoding dimensions. (b) The effect of activation function on detection accuracy of the model. (c) Batch size has a direct impact on the training time.

We performed the experiments using different batch sizes on the training data. As we expected, changing the batch size has no considerable effect on the validation loss, however, it significantly affects the training time of the model. In fact, the model learns all the data in a batch (and updates the parameters of neurons) before the next batch is learned. Thus, a small batch size increases the number of times that model learns and updates its parameters which results in much longer training phase. Fig. 7(c) and Fig. 8(c) shows the results (per epoch). Moreover, as we discussed before, employing a higher number of hidden layers has the disadvantage of higher latencies. Fig. 8(b) shows that reducing the reduction factor results in more accurate dimensionality reduction, i.e. higher level of similarity between the input feature vectors and the associated encoded vectors. As discussed before, applying a higher number of epochs provides more opportunities for the model to learn from the training data. This enhances the model performance in terms of accuracy. However, as we clarified before in this section, over-fitting is a disadvantage of applying a high number of epochs that should be taken into consideration. The results have been shown in Fig. 8(c).

C. Anomaly Detection

In this subsection, we present the experiment results for the anomaly detection (AD–VNF) module. We have used the elements of confusion matrix, i.e., *True Positive* (TP), *False Positive* (FP), *True Negative* (TN) and *False Negative* (FN) to calculate the following metrics for performance evaluation. Among the above–mentioned metrics, the accuracy and F–Score metrics are widely used for performance evaluation of classification models. The accuracy metric is used to evaluate the overall performance of the model while F–Score combines the precision and recall into a single metric that reflects the properties of both metrics.

Fig. 9(a) shows the detection accuracy of the anomaly detection (AD–VNF) module for different encoding dimensions. We obtain the highest detection accuracy 98% for the encoding dimension 32 (reduction factor 48%). As we expected, lower encoding dimensions result in less accurate detection. In Fig. 9(b), the effect of activation functions on the detection accuracy has been illustrated. This time, the ReLu activation function performs better than Tanh, Sigmoid, and Softsign. Regarding the batch size, similar to the results that we presented for the dimensionality reduction module, increasing the batch size during the training phase results in a lower number of times that the model learns and updates its parameters. This makes the training phase faster which enables architecture to work approximately near real-time, as seen in Fig. 9(c). Also the training time decreases at lower encoding dimensions because the model needs to learn vectors

TABLE III

ACCURACY METRICS OF THE PROPOSED APPROACH FOR THE DETECTION OF DIFFERENT ATTACKS

	Attack Type			
	Analysis	**Exploits**	**Backdoor**	**DoS**
Accuracy (%)	98	97.2	97.6	96.8
F–Score (%)	97.4	96.1	96.8	96.2
Recall (%)	98.6	98.1	97.9	97.6

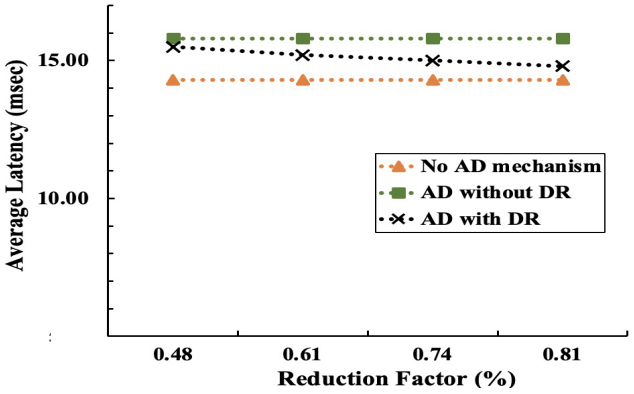

Fig. 10. The average latency created by the AD mechanism.

with smaller size when a lower encoding dimension has been applied. In other words, the volume of training data has been reduced in this case. This is indeed one of our targets in employing the DR–VNF module.

Moreover, we obtain the detection accuracy for several types of attacks available in the UNSW–NB15 dataset. Table III presents the accuracy metrics of the anomaly detection (AD–VNF) module for each type of attack. As the table shows, we obtain the highest detection accuracy 98% for the Analysis attack. In addition, based on our proof–of–concept implementation, the latency created by the proposed AD mechanism is not significant. Thus, it allows us to perform real–time traffic anomaly detection. We also observed that increasing the reduction factor results in a slight decrease in the average latency. Fig. 10 shows the average latency in three different scenarios.

D. Comparison

In this sub-section we present a comparison of our proposal with seven recent approaches. Before reading the Table IV we encourage readers to revisit Table I for a high level comparison of our work with these works. Table IV presents the average values of accuracy, F-Score, and Recall metrics regarding the detection of different types of attacks.

In [6], a security solution based on mobile edge computing has been proposed to detect network anomalies in 5G mobile networks. The authors have used deep learning techniques (DNN) to analyze network flows for detection of anomalies. Moreover, a policy–based approach has been proposed to provide a dynamic management system of the computing resources used in the anomaly detection process. However, their proposed approach needs further enhancements to offer a better detection accuracy. We reiterate that the

TABLE IV

COMPARISON OF OUR APPROACH WITH OTHER RESEARCH WORKS

Method	**Accuracy**	**F–Score**	**Recall**
[6]	95.4	97.4	99.5
[19]	87.33	87.37	87.33
[21]	92.29	94.64	99.99
[16]	68.26	80.12	93.96
[20]	93.52	95.36	98.10
[22]	95.58	96.81	98.21
[23]	93.64	95.39	98.84
Our Approach	97.4	96.6	98

authors have assumed that the centralized network orchestration handles the complete workflow and control of the proposed framework but the proposal is not tested on any 5G test bed and limited results are given to justify the necessity of novel IDS. ETSI compatibility is also not shown.

In the most recent work by Wang et al. proposed an stacked contractive autoencoder (SCAE) model for unsupervised feature extraction from raw network traffic [19]. The feature vectors generated by the SCAE model is then applied to a support vector machine (SVM) classification model to develop a cloud intrusion detection system. The authors have performed their experiments on two different datasets, namely KDD Cup 99 and NSL-KDD. However, the proposed SVM classifier needs further improvement in terms of detection accuracy, specifically, in regard to the effective recognition of some new attacks existed in the testing dataset used in the experiments. Further, in [21], a traffic inspection approach has been proposed to detect if a protected network device is participating in flooding–based DDoS attacks. The proposed approach assumes the non–stationarity and heterogeneity inherent in the next generation communication environments. For performance evaluation, the authors have used a dataset obtained through traffic observation of 61 real devices that includes different types of DDoS attacks based on TCP, UDP, and HTTP protocols. As observed in Table IV, the proposed approach needs significant improvement regarding the detection accuracy of DDoS attacks.

From our experiments we have shown optimal and trade-off results with the accuracy 97.4% in comparison to all other approaches listed in Table IV. There are some research works, even at the edge of the networks, achieving a detection accuracy of 99.99% using random forest and other models. Note that effective model design and then architectural evaluation requires a significant degree of architectural engineering [42]. Further those existing works/models used non-5G dataset (example KDD, CIDDS-001 etc.) which are almost 20 years old and do not represent current dynamic network environments. Also, we observe that those results are unlikely to represent real world detection levels and therefore gives the impression that the model is not well generalized [3]. Further, network anomaly detection is more difficult when these models are applied on UNSW-NB15 dataset [6]. Contrary we used real 5G data set and test bed for the evaluation of our proposed architecture. Our work is aligned with standard 5G ETSI-NFV architecture (evaluated in the next section).

We emphasize that from Fig. 8, 9, 10, and from Table III and IV, it is reasonable to say that our experiments are robust enough and justify the choice of works we have used for our comparison. The proposed approach is simple, feasible, effective, and practically possible as well as suitable for NGNs. Also, the comparison of our approach with the seven existing recent approaches indicates the proposed approach in this paper is promising. However, as said, to gain a deeper understanding and trust in this approach, there are still certain aspects that need to be rigorously explored which we have discussed in the next sections. In the following sections, firstly we have shown the security and complexity analysis of the proposed approach and following this we have shown a PoC of the deployment of our approach.

V. Security and Complexity of the Proposal

In this section, we have discussed a threat model to highlight the capabilities of an intruder. Following this a security analysis of the proposal is discussed. The proposed security analysis also highlights the complexity or the overhead that the proposed solution generates.

A. Threat Model

With the emergence of billions IoT devices, the IoTs adoption in various daily-life applications, combined with the lack of proper patching and securing, has made IoT devices an easy target for malicious actors [43]. 5G is becoming a promising infrastructure for IoT networks as 5G has key features such as segmented networks (network slices), enhanced privacy etc., which boost security and privacy [7]. Nevertheless, 5G success also attracts malicious actors (attackers and hackers) to look for vulnerabilities and exploit network capabilities. As said that with 5G-IoT integration, more devices are connected to networks means it creates more target and large attack surface. The attack on IoT devices (in 5G) could become more chaotic. Particularly in 5G we can categories threats into two dimensions, a) the attacks or threats on the user equipment's (or IoT) itself and b) threats related to the VNFs in the cloud (i.e., issues related to the hosting of VNFs). We consider a simple scenario in which the end nodes in the network can be a primitive IoT device. As IoT devices are cheap and easy to tamper, the attackers can inject the malicious data while remotely updating the device configuration to gain control over the end device [7]. To generalize this aspect, we note that a large number of IoT devices can be compromised and form IoT-Botnet, and a novel flood attack from this botnet on the 5G networks can be mounted. Although no such attack has yet been reported because 5G is still in its early deployment stages. However, this novel attack is one of the biggest threats on 5G infrastructure and it is vital to model, analyze and find measures to mitigate it [4], [44].

This flood attack means the compromised devices will generate exponential amount of data towards server which may impact the ability of IDS systems, for example multi-dimensional time series data faces critical problems such as dimensional explosion and data sparseness, as well as complex pattern features such as periods and trends. Such characteristics lead to rule-based anomaly detection methods (IDSs) suffer from poor detection effects, high utilization of resources and high model training and anomaly detection time [45]. To mitigate this issue, we propose DAE for dimensionality reduction to improve the security and minimize the computing burden of 5G networks. The following subsection shows that our approach is lightweight and to some extent provides immunity from flooding attack.

B. Security and Complexity Analysis: The Immunity of the Proposed Solution

Note that the proposed scheme resides as a module at each network slice, as shown in Figure 3, which therefore carries the computation burden (see DAE and DNN sub modules in Figure 3). We note that the DAE model consists of N layers where layer 1 and N are the input and output layers, respectively. We consider that in the worst use case in which we assume that the DAE model is a fully connected network of neurons (see Fig. 4), this means the DAE will impose high level of computational complexity and will generate overhead. We refer to the size of i^{th} layer as $n^{(i)}$ ($i \epsilon$ 1,2,…, N) which represents the number of neurons in that layer. Consider we have $n^{(1)} = n^{(l)} = n$, here n represents the size of feature vectors. Therefore, the output (O) of ith layer is calculated as below.

$$O^{(i)} = [O_1^i, O_2^i \ldots .. O_n^i(i)], \tag{9}$$

here, $O^{(i)} = g(\sum_{k=1}^{n^{(i-1)}} w_{jk}^{(i)} O_k^{(i-1)})$ in which $w_{jk}^{(i)}$ represents the weight of jth neuron in layer i applied to the $O_k^{(i-1)}$, and g is the activation function deployed in layer i. Now the output (number of multiplications and additions required) of fully connected DAE to compute $o^{(N)}$is obtained as below.

At each layer i ($i \epsilon$ 1,2,…, N), $n^{(i)}$ values must be computed i.e., $(O_1^i, O_2^i \ldots .. O_n^i(i))$. The computation of each value of O_j^i requires $n^{(i-1)}$ multiplications and $n^{(i-1)} - 1$ additions. Therefore, the number of multiplications (M_N) and number of additions (A_N) DAE requires to give the final output is computed as below.

$$M_N = \sum_{i=2}^{N} n^{(i)} n^{(i-1)} \tag{10}$$

$$A_N = \sum_{i=2}^{N} n^{(i)} (n^{(i-1)} - 1) \tag{11}$$

To simplify the above equations ((M_N) and (A_N)), it is reasonable to assume that the DAE's encoder and decoder networks have the same number of layers. So, the size of each layer in the encoder/decoder network is a half/twice of the size of its previous layer. Therefore, in such scenarios, for (M_N) we have;

$$M_N = 2\left[(\frac{n}{2} \times n) + (\frac{n}{4} \times \frac{n}{2}) + (\frac{n}{2^{\frac{N}{2-1}}} \times \frac{n}{2^{\frac{N}{2-2}}}) \right] \tag{12}$$

$$M_N = 2n^2 \sum_{i=1}^{\frac{N}{4-1}} \frac{1}{2^{(2i-1)}} \tag{13}$$

here, n is the size of the feature vectors. Similarly, A_N is computed as below.

$$A_N = 2\left[n^2 \sum_{i=1}^{\frac{N}{4-1}} \frac{1}{2^{2i-1}} - n \sum_{i=1}^{\frac{N}{2-1}} \frac{1}{2^i}\right] \tag{14}$$

For any specific DAE network, the n and N parameters are fixed. Therefore, irrespective to the data size (also the number of intruder nodes in the network), M_N and A_N can be accurately computed. Since n is typically a limited number (e.g., 128), M_N and A_N be small enough to be handled by currently used servers (considering their enormous computational capabilities). It means that the DAE (which is used for DR) reduces the computational complexities and enhances network's capacity to mitigate flooding attack to a great extent since it is suitable to handle large data sets with dimensions (generates by flooding attack). Also, in the following section, we have given a Proof of Concept (PoC) to show how the proposed approach can be deployed in the real scenarios as well as how this would fit in with a real-life MANO architecture. This is to capture the complexities of deploying the proposal in real practice.

VI. PoC: Integration of DR-VNF and AD-VNF With ETSI-NFV Architecture and Performance Evaluation

Note that our approach is uniquely designed to be compatible with the standard ETSI-NFV architecture and is positioned at the network edge layer. In Fig. 1 and 3, we have shown that the module can be implemented as the pluggable virtual network function on any network slice having its own independent network orchestration module. Both the Dimensionality Reduction and Anomaly Detection modules are integrated into the ETSI-NFV architecture as separate VNFs, as sub-modules of Authentication Server Function (AUSF) that are controlled by standard VNF managers as an independent management orchestration of each slice (Fig. 1 and Fig. 3). Also, note that the Access and Mobility Management Function (AMF) is a control plane function in 5G core networks. This is responsible for device ((or User Equipment (UE)) registration, mobility, reachability, and connection management.[2] Whereas to facilitate security processes in 5G, AUSF is responsible. AUSF Authenticates the UE for the requester network function (NF) (as it provides keying material to the requester NF, protects the steering information list for the requester NF, etc.). This AUSF is a NF entity in 5G core networks. It acts a network function service producer, and the NF consumer is the AMF.[3]

With these references, in our proposal, we assume that the proposed authentication scheme is a pluggable VNF, i.e., set as a separate network function. This can be an integrated part of AUSF (shown in Fig. 3). Hence, it is valid to say that the proposed approach, in the real-scenarios, is a new pluggable sub-module of the 5G-AUSF module. In this section, we have given a Proof-of-Concept, of its actual deployment, using ETSI OSM-MANO test bed to show how the proposed approach would fit in with a real-life MANO. This is to capture the difficulties of deploying the VNFs involved in process and give a sense of realism to the solution. We reiterate that this contribution presents relevant aspects of deployment of the approach and evaluation results show its performance, as below. Fig. 11 shows the structural diagram of our proposal using ETSI-VNF architecture in which we use OSM MANO for the deployment of AD-VNF and DR-VNF as pluggable VNFs module. We use Amazon Cloud Service (AWS) cloud platform for the deployment and conducting evaluation of our proposal.

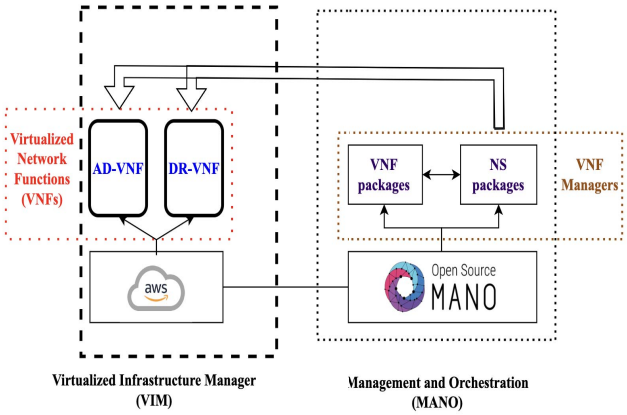

Fig. 11. Structural diagram of the integration of our proposal with ETSI-MANO architecture.

The Open-Source MANO (OSM) version 10 is setup on an Amazon Elastic Compute Cloud (Amazon EC2). The EC2 features 8GB of memory and 4 vCPU (Intel Xeon processors). The OSM connects to the Virtualized Infrastructure (VI) which is represented by the AWS. The system is aligned with ETSI-NFV architecture in 5G. AD-VNF and DR-VNF are deployed by the OSM with each VNF has 16GB of memory and 8 vCPU (Intel Xeon processors). These VNFs use Python 3.6.9 and Tensorflow version 1.14.0. We present the results of two scenarios (with or without DR module) in different Reduction Factors (48%, 61%, 74% and 81%). The dataset has 9740 observed samples with 70% of the dataset for training and the rest for validation and test procedures (10% and 20%, respectively). The batch size and epoch are set at 1024 and 100. We note that the result is the average of 10 times execution.

Fig. 12(a) illustrates the impact of Dimensionality reduction by DR-VNF on CPU Utilization. The experimental results show two scenarios; the CPU utilization(%) w.r.t varying reduction factor both with and without DR-VNF module. The CPU utilization without DR-VNF is 7.133. As we reduce the data dimensions to 0.48 (encoding 32), 0.61 (encoding 24), 0.74 (encoding 16), and 0.81 (encoding 12), the CPU utilization reduces to 7.0, 7.02, 6.808, and 6.733, respectively. This shows that the DR has a clear impact on CPU utilization. We note that the impact will be higher on larger datasets.

[2]http://www.techtrained.com/network-function-access-mobility-management-function-amf-in-5g-core-network-5g-system-5gs/ (accessed on 01/09/2022)

[3]https://www.etsi.org/deliver/etsi_TS/129500_129599/129509 /15.01.00_60/ts_129509v150100p.pdf (accessed on 01/09/2022)

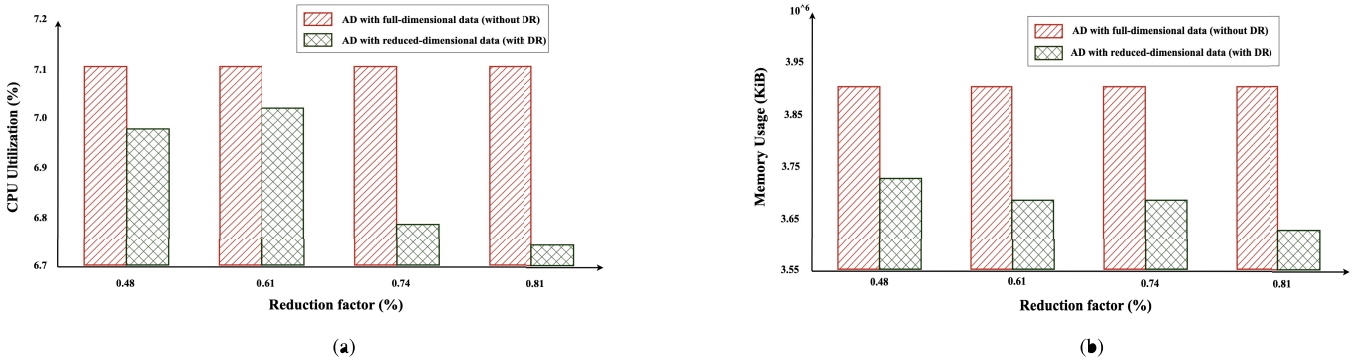

Fig. 12. (a) VNF CPU utilization with and without DR, (b) Memory usage with and without DR.

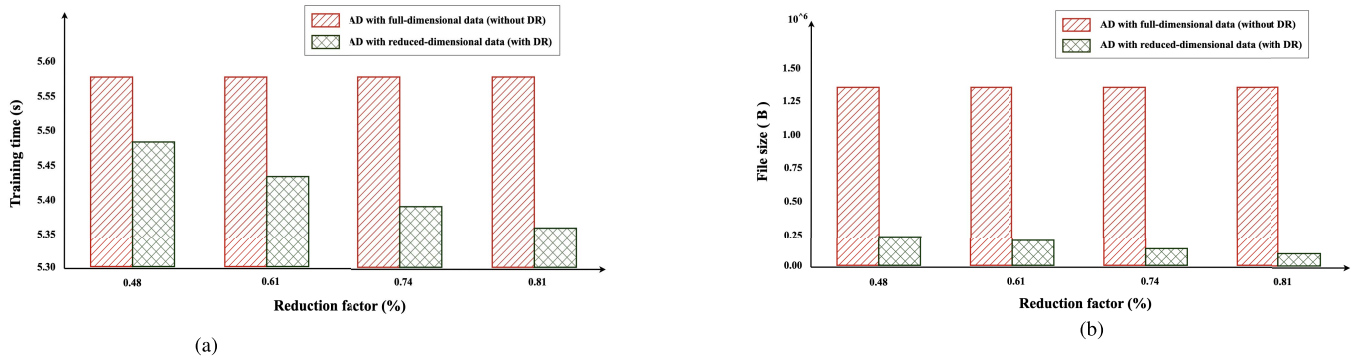

Fig. 13. (a) The impact of DR-VNF on processing script size/data file, b) The impact of DR-VNF on training time of the model.

Comparing Fig. 12(a) with Fig. 9(a), we see that higher is the dimensionality reduction factor (lower encoding dimensions), the accuracy drops but it is not that significant. On the other side it clearly gives advantage in terms of CPU utilization (Fig. 12(a)) and memory usage (Fig. 12(b)). The higher DR factor has greatly improved the CPU Utilization and memory usage. Further it impacts on the training time of the model and reduction in the script size for computation, as seen in Fig. 13(a) and Fig. 13(b), respectively.

Overall, we emphasize that the proposed IDS model is easy to deploy as VNFs as it does not introduce any great deployment challenges. Further it is useful in any 5G network to reduce computational processing infrastructure without compromising with the other KPIs (accuracy and end-to-end latency, Fig 9(a), and Fig. 10) of intrusion detection, respectively. More importantly it is highly beneficial for any cloud system (or in any network where heavy use of virtualization is seen) means an effective utilization of resources and operational costs will be reduced.

We emphasize that in this study we have presented limited but a significant early investigation into the aspects surrounding real-time anomaly detection in 5G networks context. The proposed solution has an architecture compatible to the standard 5G architecture. We have discussed in detail how the choice of parameters of DAE and DNN will affect the performance and carried out several experiments to demonstrate the impact. The performance of the proposed solution is evaluated on a 5G simulator with a comprehensive dataset. We address the two major limitations of conventional security approaches, i.e., poor compatibility with modern network designs and lack of efficient traffic processing.

VII. Summary and Future Works

In this paper, we have proposed an effective and feasible architecture which has leveraged the Deep Learning concept for intrusion detection in 5G networks. The proposed architecture is fully compatible, as validated by experiments, with the standard ETSI-NFV architecture. We have concluded that by reducing the dimensionality of the traffic feature vectors we are able to detect anomalies in real-time, which we have achieved by reducing the training time of model. We have validated that the architecture is helpful to provide the early detection of anomalies at the edge of the network before the other segments are affected. The results of our experiments show that for the reduction factor of 81% (achieved using the DR module) malicious traffic can be detected at 98.7% success rate by the AD module.

Further, the performance of the architecture using OSM MANO (5G orchestration platform) has been examined which shows that the proposed method reduces CPU Utilization, online model training time, and memory usage. This emphasizes the need to reduce the feature spaces at the edge of the network where the resources do not have much computational capacities. The deeper comparison with previous approaches emphasized the overall merit of the work.

In future, we plan to evaluate how feature extraction in real-time (time series data) will affect the performance of VNFs. Also at large dataset size we aim to optimize the trade-off

between the accuracy and the processing time. Finally, in this work the feature extraction is solely conducted at the edge level, in future, the work will be extended to analyze the impact in case if the nodes interact directly in, for example, device to device (D2D) communication.

In the real world, due to cyber-attacks on sensors the reported data can be compromised and/or blocked. Consequently, the received data can be incorrect, incomplete, and sparse presenting a significant challenge for real-time decision making and network security analytics [46]. To a great extent, the existing IDSs using standard AI models struggle to make explainable and intelligent network/application management and control decisions with compromised data. Therefore, firstly, the existing solutions are unable to recover missed or sparse data values and not fully capable to make accurate network security decisions. Secondly, in real-world data sets, data imbalance is a significant problem in that directly affects the AI models' performance [47]. Overall, it is challenging to develop a single IDS that jointly addresses both issues. We aim to extend our current work in this direction.

REFERENCES

[1] *A few words on Next Generation Networks*. Accessed: Aug. 28, 2022. [Online]. Available: https://www.itu.int/osg/csd/wtpf/wtpf2009/ngn.html

[2] K. Sood, K. K. Karmakar, V. Varadharajan, U. Tupakula, and S. Yu, "Analysis of policy-based security management system in software-defined networks," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 612–615, Apr. 2019.

[3] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," *IEEE Sensors Lett.*, vol. 3, no. 1, pp. 1–4, Jan. 2019.

[4] S. P. Rao, S. Holtmanns, and T. Aura, "Threat modeling framework for mobile communication systems," 2020, *arXiv:2005.05110*.

[5] J. Lam and R. Abbas, "Machine learning based anomaly detection for 5G networks," 2020, *arXiv:2003.03474*.

[6] L. F. Maimó, A. H. Celdrán, M. G. Pérez, F. J. G. Clemente, and G. M. Pérez, "Dynamic management of a deep learning-based anomaly detection system for 5G networks," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 8, pp. 3083–3097, Aug. 2019.

[7] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "IoTDefender: A federated transfer learning intrusion detection framework for 5G IoT," in *Proc. IEEE 14th Int. Conf. Big Data Sci. Eng. (BigDataSE)*, Dec. 2020, pp. 88–95.

[8] *Snort: An Open Source Network Intrusion Detection and Prevention System*. Accessed: Dec. 28, 2021. [Online]. Available: http://www.snort.org

[9] *Suricata: Open source IDS/IPS/NSM Engine*. Accessed: Dec. 28, 2021. [Online]. Available: https://suricata-ids.org

[10] L. F. Maimó, F. J. G. Clemente, M. G. Pérez, and G. M. Pérez, "On the performance of a deep learning-based anomaly detection system for 5G networks," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2017, pp. 1–8.

[11] J. Cao et al., "A survey on security aspects for 3GPP 5G networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 170–195, 1st Quart., 2020.

[12] S. Zhang, Y. Wang, and W. Zhou, "Towards secure 5G networks: A survey," *Comput. Netw.*, vol. 162, Oct. 2019, Art. no. 106871.

[13] K. Sood, S. Yu, D. D. N. Nguyen, Y. Xiang, B. Feng, and X. Zhang, "A tutorial on next generation heterogeneous IoT networks and node authentication," *IEEE Internet Things Mag.*, vol. 4, no. 4, pp. 120–126, Dec. 2021.

[14] K. Sood, K. K. Karmakar, V. Varadharajen, N. Kumar, Y. Xiang, and S. Yu, "Plug-in over plug-in evaluation in heterogeneous 5G enabled networks and beyond," *IEEE Netw.*, vol. 35, no. 2, pp. 34–39, Mar. 2021.

[15] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for Internet of Things," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107784.

[16] M. Savic et al., "Deep learning anomaly detection for cellular IoT with applications in smart logistics," *IEEE Access*, vol. 9, pp. 59406–59419, 2021.

[17] D. Wu, H. Xu, Z. Jiang, W. Yu, X. Wei, and J. Lu, "EdgeLSTM: Towards deep and sequential edge computing for IoT applications," *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, pp. 1895–1908, Aug. 2021.

[18] A. Dridi, C. Boucetta, S. E. Hammami, H. Afifi, and H. Moungla, "STAD: Spatio-temporal anomaly detection mechanism for mobile network management," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 894–906, Mar. 2021.

[19] W. Wang, X. Du, D. Shan, R. Qin, and N. Wang, "Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1634–1646, Jul. 2022.

[20] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018.

[21] M. A. S. Monge, A. H. González, B. L. Fernández, D. M. Vidal, G. R. García, and J. M. Vidal, "Traffic-flow analysis for source-side DDoS recognition on 5G environments," *J. Netw. Comput. Appl.*, vol. 136, pp. 114–131, Jun. 2019.

[22] S. Rezvy, Y. Luo, M. Petridis, A. Lasebae, and T. Zebin, "An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks," in *Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2019, pp. 1–6.

[23] B. Feng, H. Zhou, G. Li, Y. Zhang, K. Sood, and S. Yu, "Enabling machine learning with service function chaining for security enhancement at 5G edges," *IEEE Netw.*, vol. 35, no. 5, pp. 196–201, Sep. 2021.

[24] L. F. Maimó, Á. L. P. Gómez, F. J. G. Clemente, M. G. Pérez, and G. M. Pérez, "A self-adaptive deep learning-based system for anomaly detection in 5G networks," *IEEE Access*, vol. 6, pp. 7700–7712, 2018.

[25] A. Thantharate, R. Paropkari, V. Walunj, C. Beard, and P. Kankariya, "Secure5G: A deep learning framework towards a secure network slicing in 5G and beyond," in *Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2020, pp. 852–857.

[26] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "RTIDS: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, pp. 64375–64387, 2022.

[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. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8.

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

[29] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 122, pp. 1–35, 2019.

[30] J. A. Herrera and J. E. Camargo, "A survey on machine learning applications for software defined network security," in *Proc. Appl. Cryptogr. Netw. Secur. Workshops (ACNS)*, vol. 11605, no. 1, 2019, pp. 70–93.

[31] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020.

[32] Y. Pan, F. He, and H. Yu, "Learning social representations with deep autoencoder for recommender system," in *Proc. World Wide Web Conf.*, 2020, pp. 2259–2279.

[33] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[34] F. Ye, C. Chen, and Z. Zheng, "Deep autoencoder-like nonnegative matrix factorization for community detection," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1393–1402.

[35] G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider, "A theoretical analysis of deep neural networks and parametric PDEs," 2019, *arXiv:1904.00377*.

[36] L. Bottou, "Stochastic gradient learning in neural networks," *Neuro-Nimes*, vol. 91, pp. 687–696, Nov. 1991.

[37] V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Learning rate adaptation in stochastic gradient descent," *Nonconvex Optim. Appl.*, vol. 54, pp. 433–444, Dec. 2001.

[38] C. Szegedy et al., "Going deeper with convolutions," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9. [Online]. Available: https://research.google/pubs/pub43022/
[39] *Omnet++*. Accessed: Jan. 3, 2022. [Online]. Available: https://omnetpp.org
[40] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G–An OMNeT++ library for end-to-end performance evaluation of 5G networks," *IEEE Access*, vol. 8, pp. 181176–181191, 2020.
[41] 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)*, Nov. 2015, pp. 1–6.
[42] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim, "An empirical study on network anomaly detection using convolutional neural networks," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1595–1598.
[43] M. M. Alani, "IoTProtect: A machine-learning based IoT intrusion detection system," in *Proc. 6th Int. Conf. Cryptogr., Secur. Privacy (CSP)*, Jan. 2022, pp. 61–65.
[44] B. Santos et al., "Threat modelling for 5G networks," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, May 2022, pp. 611–616.
[45] Z. Chen, Z. Peng, X. Zou, and H. Sun, "Deep learning based anomaly detection for muti-dimensional time series: A survey," in *Proc. China Cyber Secur. Annu. Conf.* Singapore: Springer, 2021, pp. 71–92.
[46] S. Liu, J. Zhang, Y. Xiang, and W. Zhou, "Fuzzy-based information decomposition for incomplete and imbalanced data learning," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1476–1490, Dec. 2017.
[47] Y. Song, M. Li, Z. Zhu, G. Yang, and X. Luo, "Nonnegative latent factor analysis-incorporated and feature-weighted fuzzy double *c*-means clustering for incomplete data," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 10, pp. 4165–4176, Oct. 2022.

Keshav Sood received the Ph.D. degree from Deakin University, Melbourne, VIC, Australia, in 2018. Following his Ph.D. degree, he worked as a Research Fellow with the Advanced Cyber Security Engineering Research Centre (ACSRC), The University of Newcastle, NSW, Australia. He is currently a Lecturer at Deakin University. He worked on the project funded by the Defence Science and Technology Group. Some of his work is funded by the Department of Defense, Australia, and the Cyber Security Cooperative Research Centre (CSCRC), Australia. He is a Reviewer of IEEE journals and IEEE transactions, including *IEEE Network* magazine, *IEEE Wireless Communication* magazine, IEEE TRANSACTIONS ON COMPUTATIONAL SYSTEMS, IEEE COMMUNICATION LETTERS, and IEEE ACCESS.

Mohammad Reza Nosouhi (Member, IEEE) received the master's degree in telecommunications engineering from the Isfahan University of Technology, Isfahan, Iran, in 2007, and the Ph.D. degree from the University of Technology Sydney, Ultimo, NSW, Australia, in 2020. He is currently working as a Research Fellow with the Centre for Cyber Security Research and Innovation (CSRI), Deakin University, Australia. He has more than ten years of industry experience in ICT field. His research interests include post-quantum cryptography, next generation authentication systems, applied cryptography, and blockchain systems.

Dinh Duc Nha Nguyen received the bachelor's degree from the Posts and Telecommunications Institute of Technology, Vietnam, and the master's degree from the Queensland University of Technology, Australia. He is currently pursuing the Ph.D. degree with Deakin University, Melbourne, VIC, Australia. He has been awarded a place on the Dean's list of excellent academic performance two consecutive times in 2020. He has more than six years of industry experience, mainly as a network analyst and a software engineer.

Frank Jiang (Senior Member, IEEE) received the Ph.D. degree from the University of Technology Sydney and the master's degree in computer science from The University of New South Wales (UNSW), Australia. He gained the three and half years of post-doctoral research experiences at UNSW. He has published over 120 highly reputed SCI/EI indexed journals/conferences articles. His main research interests include data-driven cyber security, predictive analytics, and biologically-inspired learning mechanism and its application in the complex information security systems.

Morshed Chowdhury received the Ph.D. degree from Monash University, Australia, in 1999. He is currently an Academic Staff Member at the School of Information Technology, Deakin University, Australia. Prior to joining Deakin University, he was an Academic Staff with the Gippsland School of Computing and Information Technology, Monash University. He has more than 12 years of industry experience in Bangladesh and Australia. He was a fellow of the International Atomic Energy Agency (IAEA), where he has visited a number of international laboratory/centers, such as the Bhaba Atomic Research Centre, India, Brookhaven National Laboratory, Upton, NY, USA, and the International Centre for Theoretical Physics (ICTP), Italy. His current research interests include security of the Internet of Things, wireless network security, health data analytics, and documentation security.

Robin Doss (Senior Member, IEEE) is currently the Research Director of the Centre for Cyber Security Research and Innovation (CSRI), Deakin University. In addition, he also leads the "Next Generation Authentication Technologies" theme within the National Cyber Security Cooperative Research Centre (CSCRC). His research program has been funded by the Australian Research Council (ARC), government agencies such as the Defence Signals Directorate (DSD), Department of Industry, Innovation and Science (DIIS), and industry partners. He has an extensive research publication portfolio. His research interests include system security, protocol design, and security analysis with a focus on smart, cyber-physical, and critical infrastructures. He is a member of the Executive Council of the IoT Alliance Australia (IoTAA). He was a recipient of the "Cyber Security Researcher of the Year Award" from the Australian Information Security Association (AISA) in 2019.