2024 International Conference on Smart Applications, Communications and Networking (SmartNets) | 979-8-3503-8532-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/SMARTNETS61466.2024.10577748

Optimizing VNF Migration in B5G Core Networks: A Machine Learning Approach

Brahma Reddy Tanuboddi*[1], Gad Gad*[2], Zubair Md Fadlullah*[3], and Mostafa M. Fouda‡§[4]
*Department of Computer Science, Western University, London, ON, Canada.
‡Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA.
§Center for Advanced Energy Studies (CAES), Idaho Falls, ID, USA.
Emails: [1]btanubod@uwo.ca, [2]ggad@uwo.ca, [3]zfadlullah@ieee.org, [4]mfouda@ieee.org

***Abstract*—Software-Defined Networking (SDN) is an innovative networking paradigm that fundamentally changes the way network management and operation are approached. Network Functions Virtualization (NFV) is a network architecture concept that leverages standard IT virtualization technology to virtualize entire classes of network node functions into building blocks that can be connected or chained together to create communication services. NFV is part of the broader trend towards the virtualization of IT services and infrastructure. When combined with SDN, it provides a complete solution for a fully virtualized network, offering unprecedented levels of agility and efficiency. The concept of Virtual Network Function (VNF) migration has introduced the need for optimized algorithms to minimize migration time and costs, addressing a critical aspect of resource utilization. This paper explores the utilization of machine learning methods, especially neural networks to enhance the migration of VNFs, and presents a framework leveraging Convolution Neural Networks (CNN) and Artificial Neural Networks (ANN) to predict optimal migration paths for VNFs, aiming to minimize migration time and cost. The proposed solution analyzes network conditions, workload patterns, and resource availability, enabling dynamic and efficient VNF re-locations. This approach significantly improves network performance and reliability, making it a vital contribution to the field of network function virtualization.**

***Index Terms*—Virtual Network Function (VNF), Deep Learning, Software Defined Network (SDN), Virtual Network Function (VNF), Convolution Neural Networks (CNNs), Artificial Neural Networks (ANNs).**

I. Introduction

Software-Defined Networking (SDN) emerged as a programmable computing-integrated network systems solution by separating the control and data planes [1]–[8], simplifying network orchestration. Despite its advantages, SDN faces challenges like the Single Point of Failure (SPoF) in single-controller scenarios. SDN's centralized control approach, using protocols like OpenFlow, reduces complexity and enhances network flexibility. The use of SDN/Network Functions Virtualization (NFV) technologies by telecommunications service providers to reduce CAPEX and OPEX highlights the importance of NFV in separating network functions from hardware, allowing high-performance services on universal server equipment [9]. NFV offers a solution by replacing specialized hardware with software-based Virtual Network Functions (VNF) on standard virtual platforms. This aligns with concepts like multi-access edge computing, placing VNFs nearer to users, thus reducing unnecessary data movement and bandwidth usage, which is key for future telecom networks [10].

In real-world networks, Virtual Network Functions (VNFs) are organized into Service Function Chains (SFCs) to meet specific requirements. SFCs order VNFs to manage user traffic effectively, adapting to dynamic user demands and traffic changes. VNF scaling plays a crucial role in this, allowing for dynamic resource allocation to SFCs, and enhancing network flexibility. Scaling-out is used to deploy additional VNFs and increase network resources during high demand, while scaling-in reduces allocated resources by removing excess VNFs during low usage periods [11]. Traditional NFV orchestrators, which often overlook temporal traffic and topology changes, can lead to QoS and SLA violations [9], [12], [13]. Fig. 1 shows a high-level overview of SDN architecture.

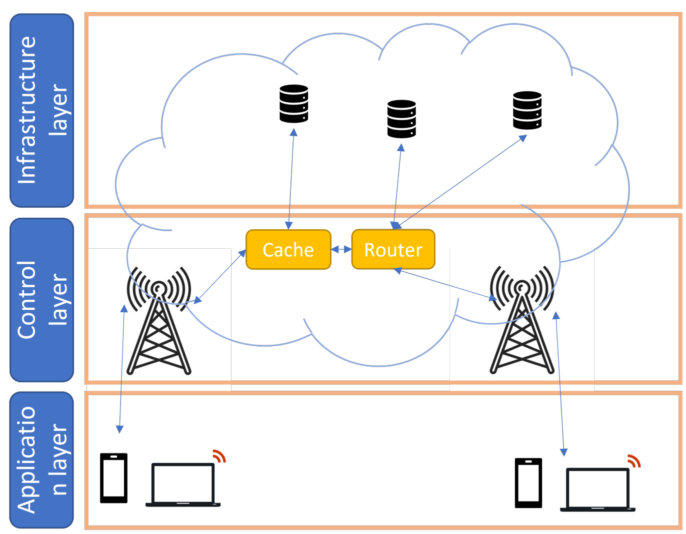

Fig. 1. A high-level overview of SDN architecture.

The paper emphasizes the integration of deep learning-based cloudlet VNF deployment in next-generation networks for ultra-low communication delay in migrating services. It proposes using deep learning techniques, specifically Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs), to optimize VNF orchestration, providing a clear and concise representation of the areas needing further exploration and development. These techniques aim to minimize communication delays and migration overheads, providing efficient solutions for large-scale networks. The paper validates the models through simulations, demonstrating their effectiveness over traditional methods, such as Integer Linear Programming (ILP) and convex optimization.

The rest of the paper is organized as follows. In section II we provide a summary of the recent studies on VNF. Section III illustrates the considered CNN and ANN models and VNF in detail. Section IV describes labeled data generation. Section V explains the simulation environment for training and testing of the proposed model. Section VI discusses and compares the results obtained from different models against the ILP-optimized framework. Finally, Section VII concludes the paper.

II. Literature Review

Research in VNF placement actively explores numerous optimization techniques from the computing domain. Algorithms like Dynamic Programming, greedy algorithm, Dijkstra's, multi-stage, Other shortest path finding algorithms, ILP, and Mixed-Integer Linear Programming (MILP) are commonly applied to achieve resource-efficient VNF placement, ensuring resources are not squandered within the confines of finite physical infrastructures. The core goal is to refine the placement of VNFs for the most effective use of the available network resources.

The authors in [14] proposed a system architecture using a VNF to deploy SDN controller clusters. This approach addresses scalability challenges in multi-controller environments by leveraging Open Source MANO (OSM), OpenStack, and ONOS as the SDN controller, ensuring high scalability and flexibility. The architecture aims to dynamically create SDN controllers, manage their lifecycle, and redistribute resources, thus avoiding controller overload. The system's reliability, scalability, and consistency were verified using the Cbench measurement tool.

The research in [15] presents a framework that ensures reliable service placement across distributed locations, optimizing latency and resource use while enforcing connectivity policies for service reliability, safety, and security. It introduces an approach using optimization modulo theories (MaxSMT) to tackle the virtual network embedding problem with expressive constraints, aiming to address the challenges of service graph complexity and real-time reconfiguration in the dynamic landscape of industrial networks [16], [17].

An over-deployment strategy has been proposed in [18] that accounts for the actual running load of VNFs rather than just peak resource requests, combined with an automatic migration process to balance loads across physical machines.

Apart from the static algorithms, the application of machine learning to predict optimal VNF deployment in Multi-access Edge Computing (MEC) topologies has been proposed in [19]. They utilized the Edge-Connected Convolution algorithm within a Graph Neural Network Using ILP solutions from simulated service requests, the study trains a machine learning model to forecast future VNF deployments with over 90% accuracy for a 5-minute future point compared to the ILP benchmark. This research could significantly benefit dynamic resource allocation for large-scale IoT services, leading to improved Quality of Service (QoS) and operational efficiency.

Reinforcement Learning models also show better results when compared to traditional algorithms. The research in [20] proposes a reinforcement learning (RL)-based method to optimize QoS and minimize resource consumption. The approach involves formulating VNF deployment as a Markov decision process and applying RL algorithms for scaling decisions in dynamic network environments. The authors demonstrate that their RL-based method outperforms traditional ILP-based deployments, offering more efficient resource utilization and better QoS management.

And Deep Reinforcement Learning methods also show significant results [21]. The study focuses on overcoming challenges in dynamically changing network environments by formulating the VNF placement problem as a Binary Integer Programming (BIP) model. It introduces a Double Deep Q Network-based VNF Placement Algorithm (DDQN-VNFPA), which aims to efficiently manage network performance metrics such as service function chain request (SFCR) rejection rates, throughput, end-to-end delay, and load balancing. The proposed solution demonstrates improved network performance compared to existing algorithms, indicating a significant advancement in VNF deployment strategies.

Apart from single models, Ensemble models also show optimal results, [22] proposes innovative VNF deployment strategies using ensemble deep learning techniques for IoT services in 6G networks. It introduces two deep learning models, E-ConvNets and E-ANN, for VNF deployment, focusing on minimizing communication delays and migration overheads. The study demonstrates these methods' superiority over traditional models in terms of performance, scalability, and real-time solution provisioning. The research highlights the potential of deep learning in enhancing VNF orchestration and management in dynamic network environments.

III. Proposed Deep Learning-Aided VNF Deployment

A network architecture for VNF management comprises two main areas: the edge domain with small data centers (cloudlets) and the Access Network domain with base stations (gNBs). Each gNB, serving numerous users, connects to a single cloudlet, facilitating service delivery. The system adapts to user movements, necessitating the reevaluation and potential relocation of VNFs to maintain optimal service, while also aiming to reduce the associated costs of such relocations.

For this system, we are considering the ILP-optimized framework proposed in [22], a total of 7 Convolution Neural Networks and 7 Artificial Neural Networks with varying numbers of hidden layers/nodes and optimizers.

A. Convolution Neural Network (CNN)

CNN structures are highly effective in pattern recognition for various networking tasks [23]. A CNN model employs convolutional layers to automatically and adaptively learn spatial hierarchies of features from input data. This ability to extract and learn features makes a CNN model particularly suitable for tasks like image classification, object detection, and facial recognition, making CNNs a cornerstone in the field of computer vision and pattern recognition.

In this project, we are using the CNNs with 7 different optimizers (Adam, SGD, Adamax, RMSProp, etc). We followed [22] for the other hyperparameters like kernel size, activation function, batch size, and dropout rate.

B. Artificial Neural Networks

Artificial Neural Networks (ANNs) are widely used in pattern recognition due to their ability to learn and model complex patterns in data. They consist of interconnected nodes or neurons that process input data, learning to recognize patterns through training. ANNs are versatile, handling various types of data and applications, from speech and image recognition to text classification. Their strength lies in their ability to learn from examples, making them highly effective for tasks where explicit programming for pattern recognition is challenging.

For this study, we considered a total of 7 ANN models with varying numbers of hidden nodes and optimizers. For all other hyperparameters, we followed [22].

The aim is to position pre-trained best models in cloudlet Data Centers (DCs), enabling the deployment decisions made during the testing phase to provide real-time solutions. This approach is designed to significantly reduce the execution time required for predictions, thereby offering rapid responses in operational environments.

IV. Labeled Data Generation

In this subsection, we describe the process of generating a labeled dataset for training the machine learning models. Due to a lack of standardized datasets, we are using the ILP-optimized framework proposed in [22] for generating the data set. ILP's optimal results help create supervised learning datasets, and due to reinforcement learning's challenges with large solution spaces, deep learning is favored for its ability to manage decision spaces and avoid convergence issues. The training model inputs incorporate various network features and the ILP-derived target variable, creating a labeled dataset that can inform efficient VNF allocation decisions. Service providers can use resource allocation analytics to build pre-trained models for effective VNF allocation.

We are selecting all the parameters in table I at random, and also assign VNFs to gNBs at random. And generate a random system; then we solve the random system using the ILP-optimized framework proposed in [22], and then combine the decision variables with the generated random system values to form a data set for a single epoch. We are using the Gurobi optimizer for implementing the ILP optimization framework and generating the training data sets.

V. Simulation Environment

In this paper, we considered a network with 12 cloudlet DCs, and the capacity of each cloudlet DC is taken as 5000. i.e., each cloudlet DC can run at a maximum of 5000 VNFs. The number of gNBs that each cloudlet DC can manage is in the range of 5 to 25. We assigned these at random for each cloudlet DC. And the number of VNFs per gNB can vary from 100 to 500. The communication delay between a cloudlet DC and gNB can vary from 10 to 200 milliseconds and the communication delay between two cloudlet DCs can vary from 2 to 5 milliseconds. The transfer rate between any two DCs is 500MBps and the size of each VNF can vary from 1 - 5 MB. Finally, the priority factor η is taken as 0.5 to give equal priorities for the VNFs running on a given cloudlet DC and the ones that are not running on that cloudlet DC.

TABLE I
ILP-optimization framework parameters.

Notation	Description
$D = \{d_1, d_2, \ldots, d_{\lvert D\rvert}\}$	The set of cloudlet DCs in the network
$E = \{e_1, e_2, \ldots, e_{\lvert E\rvert}\}$	The set of all gNBs connected to the cloudlet DC
$V = \{v_1, v_2, \ldots, v_{\lvert V\rvert}\}$	The set of all VNFs
v_{worst}	The VNF with worst delay
$t_{e,l}$	The communication delay between a cloudlet DC, d_l and the cloudlet DC, d_k provided that $e_k \in E$
$t_{j,k}$	The communication delay between gNB, $e_j \in E$ and the cloudlet DC, $d_k \in D$
S_k	Size of VNF, $v_k \in V$
ϕ_k	Cost to place any VNF to some cloudlet DC, $d_k \in D$
ψ_k	Cost to take service from cloudlet DC, $d_k \in D$
C_k	Capacity of the cloudlet DC, $d_k \in D$ for holding VNFs
σ_i	Execution time of VNF, $v_i \in V$
η_i	Priority factor of VNF migration or relocation
N_e	Number of VNFs that are already executing in cloudlet DC, $d_k \in D$
$\tau_{r,k,j}$	Summation of communication, relocation, and execution time of VNF, $v_k \in V$ of gNB, $e_j \in E$ is placed at cloudlet DC, $d_k \in D$

The execution time of each VNF can vary from 0 to 5 milliseconds. The cost to take services from a cloudlet DC and the cost to place any VNF that is not running on that cloudlet DC varied from 0.1 to 0.9. Finally, the maximum communication delay of the migration has been considered 5000 milliseconds, based on which we solve the ILP-optimized framework to generate decision variables.

VI. Results and Discussion

In this section, we explain the training and testing of all the CNN and ANN models and compare their performance against the ILP-optimized framework.

As described in Section IV, after generating the labeled data, we train all ANN and CNN models on the same random system with the decision variables obtained by solving the ILP. We trained all the models with 100 different random systems.

For testing, we used the same ILP-optimized framework for generating some random system with the given parameters. We perform the testing in two different scenarios. First, we kept the number of gNBs per each cloudlet DC as constant and varied the number of VNFs per each gNB and tried to migrate i) 5% of total VNFs (Fig. 2), ii) 10% of total VNFs (Fig. 3), and iii) 15% of total VNFs (Fig. 4). And performed the same by increasing the number of gNBs per cloudlet DC from 5 to 25 with an increment of 5 in each step. In the second scenario, the number of gNBs per cloudlet DC varies and we generated a random system for migrating a constant number of VNFs in incremental order ranging from 100 to 400. The performance metrics of ANNs and CNNs are as follows.

TABLE II
CNN Selected Parameters and Accuracy.

Optimizer	Activation Function	Kernel Size	Batch Size	Dropout Rate	Epoch	Accuracy
Adadelta	ReLU	3	50	0.5	10	91.58%
Nadam	ReLU	3	50	0.5	10	86.22%
SGD	ReLU	3	50	0.5	10	88.12%
RMSprop	ReLU	3	50	0.5	10	88.02%
Adagrad	ReLU	3	50	0.5	10	83.30%
Adam	ReLU	3	50	0.5	10	91.72%
Adamax	ReLU	3	50	0.5	10	90.95%

TABLE III
ANN Selected Parameters and Accuracy.

Optimizer	Activation Function	Hidden Nodes	Batch Size	Epoch	Accuracy
Adadelta	ReLU	64	50	10	89.23%
Nadam	ReLU	50	50	10	88.38%
SGD	ReLU	64	50	10	90.04%
RMSprop	ReLU	30	50	10	89.92%
Adagrad	ReLU	64	50	10	91.45%
Adam	ReLU	60	50	10	92.62%
Adamax	ReLU	50	50	10	92.86%

A. Constant number of gNBs under each cloudlet DC

Figs. 2 and 3 represent the migration time associated with the constant number of gNBs per cloudlet DC and the varying number of VNFs per gNB. It can be clearly seen that CNN performs better than both ANN and ILP-optimized. Also in CNN, the models with SGD and Adamax optimizer perform well when compared to the other optimizers, as shown in table II. And there is no big difference between ANN and ILP-optimized, both are taking almost similar time in predicting the optimal path for VNF migration. The ANN model with Adam optimizer performs well when compared to the other optimizers, as shown in table III.

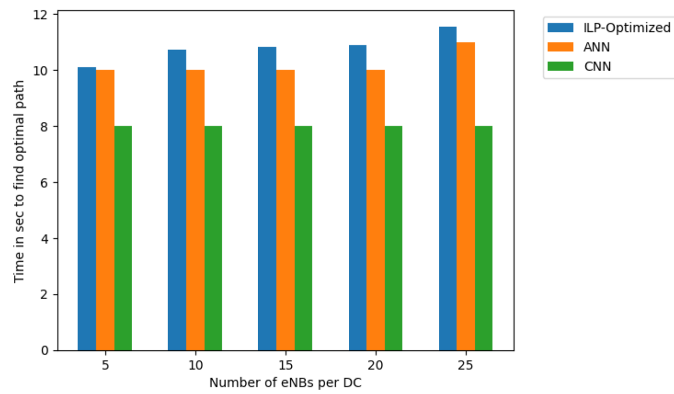

Fig. 2. Performance comparison for 5% of VNFs migration.

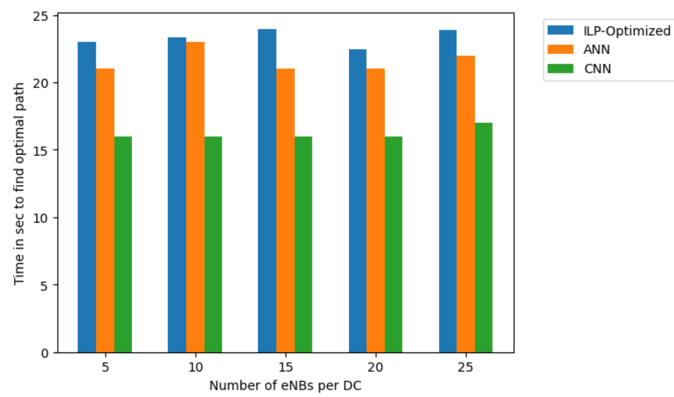

Fig. 3. Performance comparison for 10% of VNFs migration.

We continued the experiment by increasing the percentage of VNFs migrating to 10% and 15% of total VNFs. Both experiments show similar results. CNN models with SGD and Adamax optimizer take very little time compared to the ANN and ILP-optimized frameworks. However, as the number of VNFs increased, the ANN models also showed better results than the ILP-optimized framework.

B. Constant number of VNF migrations

Fig. 5 represents the migration time associated with the migration of a constant number of VNF. This time, the number of gNBs per cloudletDC varied between 5 to 25, and we simulated a system for migrating 100 to 400 VNFs. CNN models perform better than both ANNs and ILP-optimized frameworks. The performance of ANNs has increased with the increase in the number of VNFs.

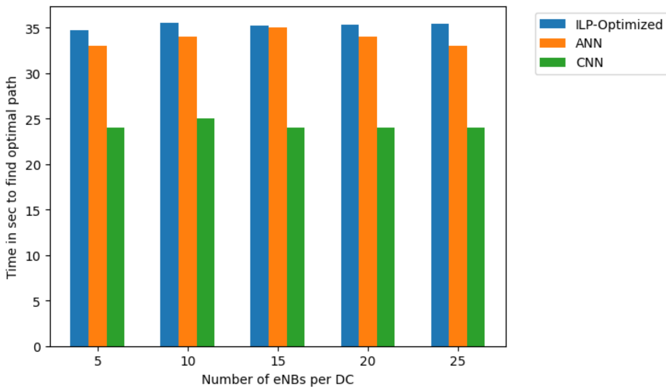

Fig. 4. Performance comparison of Models for 15% of VNFs migration.

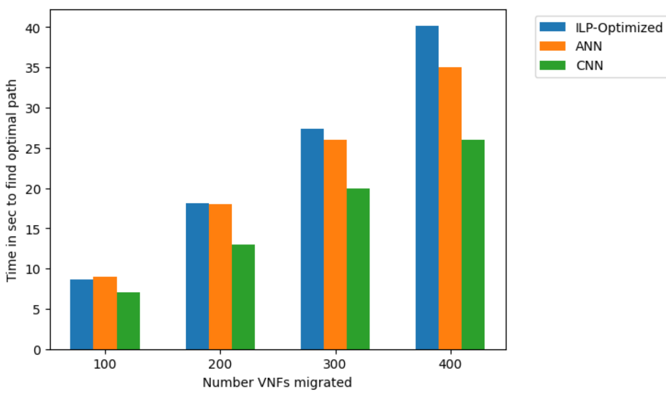

Fig. 5. Performance comparison for a constant number of VNFs.

VII. Conclusion

This paper demonstrates the potential of Convolution Neural Networks and Artificial Neural networks in enhancing the efficiency and effectiveness of VNF migration. The proposed framework, leveraging advanced algorithms, significantly reduces migration time and cost, proving crucial for modern network environments. This research not only showcases the practical application of machine learning in network function virtualization but also sets a precedent for future innovations in this rapidly evolving field.

References

[1] S. Hashima, Z. M. Fadlullah, M. M. Fouda, E. M. Mohamed, K. Hatano, B. M. ElHalawany, and M. Guizani, "On softwarization of intelligence in 6G networks for ultra-fast optimal policy selection: Challenges and opportunities," *IEEE Network*, vol. 37, no. 2, pp. 190–197, 2023.

[2] Z. M. Fadlullah and M. M. Fouda, "Programmable computing in 6G," in *Towards a Wireless Connected World: Achievements and New Technologies*, A.-S. K. Pathan, Ed. Springer International Publishing, 2022.

[3] B. Mao, F. Tang, Z. M. Fadlullah, and N. Kato, "An intelligent route computation approach based on real-time deep learning strategy for software defined communication systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1554–1565, 2021.

[4] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5141–5154, 2018.

[5] F. Tang, B. Mao, Z. M. Fadlullah, and N. Kato, "On a novel deep-learning-based intelligent partially overlapping channel assignment in SDN-IoT," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 80–86, 2018.

[6] B. Mao, F. Tang, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "A novel non-supervised deep-learning-based network traffic control method for software defined wireless networks," *IEEE Wireless Communications*, vol. 25, no. 4, pp. 74–81, 2018.

[7] M. I. Hamed, B. M. ElHalawany, M. M. Fouda, and A. S. T. Eldien, "A novel approach for resource utilization and management in SDN," in *2017 13th International Computer Engineering Conference (ICENCO)*, 2017, pp. 337–342.

[8] M. I. Hamed, B. M. ElHalawany, M. M. Fouda, and A. S. Tag Eldien, "A new approach for server-based load balancing using software-defined networking," in *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2017, pp. 30–35.

[9] S. I. Kim and H. S. Kim, "A VNF placement method based on VNF characteristics," in *2021 International Conference on Information Networking (ICOIN)*, 2021.

[10] M. Emu, M. Y. Hassan, Z. M. Fadlullah, and S. Choudhury, "Algorithm package of AI-driven SDN controller-switch load balancing strategies," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2023, pp. 347–354.

[11] X. Zhao, X. Jia, and Y. Hua, "An efficient VNF deployment algorithm for SFC scaling-out based on the proposed scaling management mechanism," in *2020 Information Communication Technologies Conference*, 2020.

[12] Z. M. Fadlullah, B. Mao, and N. Kato, "Balancing qos and security in the edge: Existing practices, challenges, and 6G opportunities with machine learning," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2419–2448, 2022.

[13] Z. M. Fadlullah, C. Wei, Z. Shi, and N. Kato, "GT-QoSec: A game-theoretic joint optimization of qos and security for differentiated services in next generation heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1037–1050, 2017.

[14] M.-H. Cheng, W.-S. Hwang, Y.-J. Wu, Y.-T. Guo, and M. C. Chen, "A dynamic VNF deployment to avoid controller overload in SDN-cluster," in *2023 9th International Conference on Applied System Innovation (ICASI)*, 2023.

[15] G. Marchetto, R. Sisto, F. Valenza, J. Yusupov, and A. Ksentini, "A formal approach to verify connectivity and optimize VNF placement in industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, 2021.

[16] X. Zhou, B. Yi, X. Wang, and M. Huang, "Approach for minimising network effect of vnf migration," *IET Communications*, 2018.

[17] Y. Hao, L. Zhang, L. Zheng, Z. Zhao, and Z. Li, "An efficient vnf deployment scheme for reliable service chain in software defined satellite networks," in *Proceedings of the 2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2022.

[18] J. Fu and G. Li, "An efficient VNF deployment scheme for cloud networks," in *Proceedings of the 2019 IEEE 11th International Conference on Communication Software and Networks*, 2019.

[19] S. Park, H.-G. Kim, J. Hong, S. Lange, J.-H. Yoo, and J. W.-K. Hong, "Machine learning-based optimal VNF deployment," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 67–72.

[20] N. Seo, D. Heo, J. Hong, H.-G. Kim, J.-H. Yoo, J. W.-K. Hong, and H. Choi, "Updating VNF deployment with scaling actions using reinforcement algorithms," in *Proceedings of the 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2022.

[21] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal vnf placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 263–276, 2020.

[22] M. Emu and S. Choudhury, "Ensemble deep learning assisted VNF deployment strategy for next-generation IoT services," *Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada*, 2021.

[23] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.