Traffic-Aware Placement of Network Function Virtualization (NFV) in Cloud Environment: Issues and Open Challenges

Nadim Rana
Department of Computer Sciences, College of Engineering and Computer Sciences, Jazan University, Jazan, Saudi Arabia

Zeba Khan
Department of Computer, Applied College, Jazan University, Jazan, Saudi Arabia

Javed Azmi
Department of Computer Science & Engineering, School of Engineering Science and Technology, Jamia Hamdard, New Delhi 110062, India

ABSTRACT
Network Function Virtualization (NFV) marks a fundamental shift in how network services are designed and deployed by separating network functions from proprietary hardware and running them as software on standard cloud infrastructures. While NFV provides flexibility, scalability, and cost savings, the performance of virtualized services depends heavily on the placement of Virtual Network Functions (VNFs), especially under changing and diverse traffic patterns. Poor placement can lead to increased packet delay, inefficient resource utilization, and breaches of service-level agreements. This paper offers a thorough analysis of traffic-aware VNF placement in cloud and hybrid cloud–edge settings. This study reviews current NFV placement methods, including Integer Linear Programming (ILP), Binary Integer Programming (BIP), mixed-integer models, heuristic algorithms, and recent AI-driven approaches, and discusses their advantages and limitations with respect to traffic management, latency, scalability, and computational demand. Emphasis is placed on how traffic features, service chaining, and deployment architecture affect placement choices. Additionally, the paper explores how analytics and intelligent orchestration can improve VNF placement decisions. It proposes a tool-supported framework that leverages historical traffic data, Microsoft Azure infrastructure, and the Open Network Automation Platform (ONAP) to improve placement strategies. Finally, the paper identifies key research gaps, challenges, and issues, including handling traffic fluctuations, multi-objective optimization, cloud–edge coordination, security concerns, and the explanation of AI-based solutions. This work serves as a reference for researchers and practitioners interested in developing scalable, traffic-aware, and deployable NFV placement solutions.

General Terms
Network Function Virtualization (NFV), Virtualized Network Function (VNF), Binary Integer Programming (BIP), Network Automation Platform (ONAP)

Keywords
Network Function Virtualization (NFV), Virtualized Network Function (VNF), Binary Integer Programming (BIP), Network Automation Platform (ONAP)

1. INTRODUCTION
The primary motivation for Network Function Virtualization (NFV) is to decouple network functions (NFs) from specialized hardware and execute them in a virtualized environment on standard hardware. NFV is a recent development in networking that marks a major shift in network design, enabling functions to be located in small cloud nodes or distributed across the network, using software-defined mechanisms to manage network flows. Running a virtualized function requires resources such as a processor, memory, and storage, which can be local or distributed. Mapping a VNF to underlying resources (physical or virtual nodes) and linking logical connections to physical links is known as VNF placement, which greatly affects NFV performance. Similar to VM placement, VNF placement involves resource assignment but differs in key respects. Typically, VM placement occurs within a single data center. Conversely, VNF placement can span multiple data centers, including the edge network, thereby reducing the applicability of VM placement research. Additionally, the placement of traditional VNFs differs from that of VNFs; traditional VNFs are usually assigned to a specific physical node, making it easier to measure performance between the node and the VNF. However, with VNF, performance constraints must be accounted for for both the VM hosting the virtualized function and the physical host on which the VM operates [1].

As an emerging technology, NFV faces several challenges, one of which is the efficient placement of NFV. The placement of software-implemented network functions directly impacts traffic operations. If the function is placed in data centers, data traffic may take indirect paths, potentially causing packet delays. The VMs hosting virtualized functions should be placed where and when they are needed, as they have a vital impact on the performance of offered services. Some services have a performance advantage when running on the edge network, and others might have it on the center network. For instance, in operators' networks, SIP service functionality must run at the edge of the operator network, where services like deep packet inspection (DPI) are not location-specific and can run anywhere as long as traffic flow is directed [2].

Several studies in the literature have addressed the NVF problem. The use of existing NVP placement algorithms and compare the efficiency in multiple cloud data centers using the GEANT Network dataset. The studies show both Binary Integer Programming (BIP) and Integer Linear Programming (ILP) based algorithms. Microsoft Azure is used as an NFV infrastructure to instantiate VMs. On the VMs, the VNF will be created using ONAP (an open-source software platform for NFV design, creation, orchestration, and management), and it will be mapped to the GEANT Network. GEANT Network is an advanced network infrastructure for research and education. SNDLIB is a library of test instances for Survivable fixed telecommunication network design, serving the research community as a standardized benchmark for testing,

evaluating, and comparing network design models. In another work, analytics are used to determine traffic to the virtualized function based on packet delay and latency. The solution places the virtualized function in suitable locations: the data center, the end-user location, and across different geographic regions. The algorithm will also check the server's hardware profile, and a virtualized network function (VNF) will be deployed where the required resources are available [2-4]. This study tries to answer the following questions.

Research Questions:

1. How do existing NFV placement approaches compare in terms of traffic optimization, packet delay, scalability, and computational complexity?
2. How does traffic-aware VNF placement differ across centralized cloud, hybrid cloud–edge, and distributed NFV architectures?
3. How can traffic analytics and intelligent orchestration improve VNF placement decisions?
4. How can emerging technologies such as artificial intelligence, machine learning, and reinforcement learning enhance automation, adaptability, and fault tolerance in traffic-aware NFV placement?
5. What are the unresolved research gaps and open challenges that must be addressed to enable scalable, secure, and real-world deployable traffic-aware NFV placement solutions?
6. What challenges limit the effectiveness of traffic-aware VNF placement in cloud and hybrid cloud–edge environments?
7. What open challenges remain in achieving scalable, secure, and adaptive traffic-aware NFV placement?

These questions focus on understanding the challenges, comparing existing solutions, exploring optimization methods, and investigating the integration of new technologies in NFV. Each question helps frame a specific area within the NFV domain for deeper exploration in the literature review.

Section 2 conducts the literature review, evaluating related research. Section 3discusses the NFV architecture and the contemporary literature on NFV placement. Section 4 highlighted the research gap and open challenges in the NFV placement, and Section 5 concludes the study.

2. PREVIOUS STUDY

In this section, prior work on NFV placement is discussed. The author of [5] developed a model that formalizes the chaining of network functions (NFs) using context-free grammars. Also, process deployment requests and construct a virtual network function graph that maps to the network. The mapping is formulated as a mixed-integer quadratically constrained program (MIQCP) to determine the placement of the network functions and to chain them together, while respecting limited network resources and the functions' requirements. In [6], the author formulated the Traffic Aware Placement of Interdependent Middleboxes problem as a graph optimization problem. When the flow path is predetermined, the optimal design of algorithms places a non-ordered or totally ordered middlebox set, and proposes an efficient heuristic for the general scenario of a partially ordered middlebox set, after proving its NP-hardness. If the flow path is not predetermined, the problem is NP-hard, even for a non-ordered middlebox set, and the authors propose a traffic- and space-aware routing heuristic. In [7], the author proposed a framework for service placement decision, an integer linear programming model to resolve service placement and minimize network transport delay, and a heuristic solution based on the improved quantum genetic algorithm. In [8], the authors proposed a sampling-based Markov approximation (MA) approach to solve the combinatorial NP-hard problem OPNET. Due to the long convergence of MA, a novel approach that combines MA with matching theory, named SAMA, is proposed to obtain an efficient solution.

In [9], the author presented a scheme that optimizes joint routing and the placement of virtual network functions (NFs). The proposed Integer Linear Program (ILP) increases the number of NFs satisfied whilst reducing the resources needed to serve the requested NFs and route each flow of traffic and to the best of their knowledge, increasing the number of requests that satisfy the joint problem, under the condition that datacenters may lack enough resources to fulfill all the requests that have not been previously studied. This lack of resources typically causes network components or data centers to fail. Furthermore, the proposed ILP considers assorted environments where (a) the set of functions that the datacenter implements, and (b) the cost and resources that are needed to implement a virtual network function are mostly datacenter dependent. Numerical results show that the ILP yields the most favorable solution for small- to medium-sized networks within a reasonable time limit. The results also show that, in many NFs, the gap between the ILP and the LP relaxation was small (a few points). In addition, this paper presents a low-complexity, greedy heuristic approach suitable for large networks. The authors also investigate the application of the proposed scheme to scenarios with varying parameter values, including request counts, data centers, resources, and a large number of functions. The scalability of the ILP with respect to the aforementioned parameters is under study. In conclusion, the implementation of the proposed scheme using an open network operating system software-defined network (ONOS-SDN) is being evaluated.

The author of [10] demonstrated a new technique for addressing the network function placement problem by using a distributed algorithm implemented over a network protocol. A simulation study of the distributed algorithm indicates that it scales well with the number of flows in the network test, yielding near-optimal VNF instance placement. More experiments with VNF instances, such as dynamic transfer and data flow among VNFs across nodes, will demonstrate the value of this approach.

The author in [11] showed that a distributed online processing network service chain was developed to reduce the time-average cost of VFN while stabilizing the network. Queues of unexecuted network services were configured at the VNF level rather than at the service level, thereby facilitating queue management and reducing signaling. Using Lyapunov optimization, asymptotically optimal decisions for processing and routing network services are generated immediately within each VM, thereby adapting to the network topology and service arrivals. The placement of automated VNFs was proposed to reduce the length of unexecuted network service queues, thereby enabling a stable, adaptive deployment of VNFs in response to individual VM connectivity and service distribution. A simulation study also showed that the proposed schemes reduce the time-average NFV cost by 71% and the delay by 74%.

In [12], the distribution of cloud resources across many locations is presented, with several reasons, such as locating resources closer to end users, reducing bandwidth utilization,

increasing availability, etc. The same concept could be applied to VNFs to minimize the distance that network traffic must travel to reach the network function, thereby reducing latency. However, the authors in [13] have a different approach to VNF placement. They say that when the demand for VNF is low, then the VNF should move to a smaller host, and the unused host should be turned off to save energy. When demands increase, the VNF should be migrated to the larger node to meet them, and this process should be automatic. Sometimes, the migration process consumes a large amount of bandwidth, which can adversely affect the network. The author says that minimizing high-bandwidth utilization when demand increases can be achieved by predicting traffic variation. According to the proposed model, the system will start migrating VNFs in advance by predicting the traffic pattern. They have used a model predictive control approach to solve this problem. However, they have tested the model in a simulated environment, and the results may differ from those in the actual network traffic. Another limitation of the model is the algorithm's distributed nature when dynamically placing VNFs.

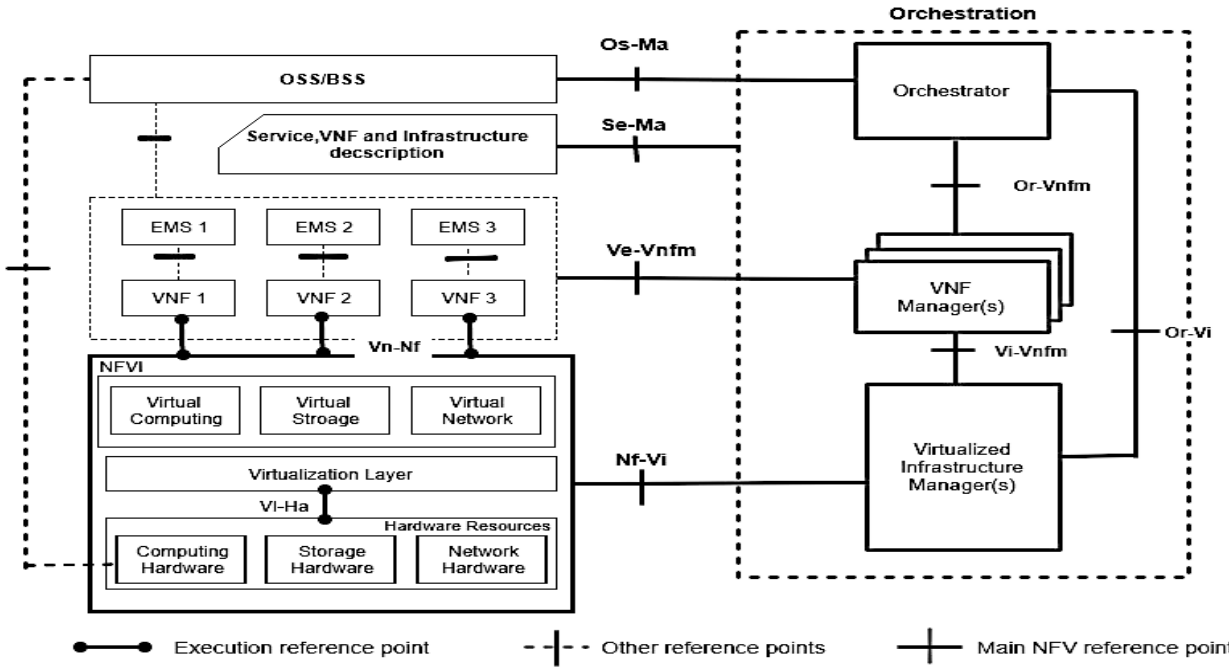

Fig 1: NFV Reference Architectural Framework

In [14], the authors address the problem of creating and destroying virtual links and nodes in response to traffic demand and service requests while minimizing energy consumption. They have also added that it complicates the task when network functions run as applications in a virtualized environment across multiple networks and locations. To solve the problem, they have used an embedded algorithm, but it accepts only a limited set of matrices; it should account for additional constraints across both infrastructure and services. In the following sub-section, the architecture of the network function virtualization platform is discussed, followed by its placement problem.

3. NETWORK FUNCTION VIRTUALIZATION ARCHITECTURE

To establish a standard NFV architecture and ensure a consistent approach, the European Telecommunications Standards Institute (ETSI) has created the Industry Specification Group (ISG). According to [15], ISG has over 150 active members from both industry and academia and manages several working groups, including Infrastructure Architecture, Management and Orchestration, Software Architecture and Reliability, Security Performance, and Portability. In Figure 1, the NFV reference architectural framework comprises three key entities: Virtualization Infrastructure (NFVI), which replaces dedicated traditional hardware; Virtual Network Functions (VNFs); and Management and Orchestration (MANO).

NFV Infrastructure: NFVI is the hardware and software components on which NFV is deployed. It could span multiple locations, including the edge network. The virtualization layer sits on top of physical hardware, serves as an abstraction layer for virtualized functions, and meets their resource requirements.

Virtualized Network Function: The actual network function, such as a firewall, IPS, or IDS.

Management and Orchestrator: These components orchestrate and manage NVF infrastructure, software resources and network services.

3.1 NFV Placement Problem

Integer linear programming (ILP) was used to address service placement and reduce network transport delay; a heuristic solution, the improved quantum genetic algorithm, was then applied. The simulation results showed that the transport delay is lower than that of other schemes, indicating improved performance. And 30% less in the average traffic transport delay. Even NFV remains under investigation and standardization, and many prior studies propose solutions to the placement problem in cloud environments and operator networks using recent mathematical and optimization approaches such as ILP, BIP, MIQCP, and OP. In many prior studies, the problem is treated as NP-hard, and most have used heuristics to solve the placement problem [16].

Moreover, the author in [17] has used ILP to solve the VNF placement problem. They have focused on a specific network function called Deep Packet Inspection (DPI). They found that

the placement of DPI based on traffic demand and its mapping to available resources is problematic. They have implemented the solution using GLPK and a heuristic using Java Universal Network/Graph Framework. They have used the GNU Linear Programming Kit (an open-source linear programming solver written in C) as the linear solver and the GEANT Network as the network traffic dataset.

Further, they have used a centrality-based greedy algorithm to assess and validate the real dataset. [18] have also presented an integer linear programming (ILP)- based solution to optimize VNFs and their endpoint traffic for both routing and placement. The objective of the scheme is to maximize the number of efficient and optimal VNF placements. At the same time, the scheme minimizes both routing and infrastructure costs while satisfying the requests. While numerical results demonstrate that the proposed ILP can be applied to small- to medium-sized networks, the paper also presents a low-complexity greedy heuristic for large networks. According to the authors, their work is the first study to address the problem of maximizing the number of requested NFs fulfilled while minimizing routing and infrastructure costs. They also emphasize that this approach is better suited to scenarios in which the data center may lack sufficient resources to satisfy all requests. The proposed solution is well-suited to the routing and placement of VNFs when traffic flows are flexible and resources are limited. Still, there are some limitations: it is a low-complexity greedy heuristic approach, and the results also show a small gap in the number of satisfied NFs between the ILP and the corresponding LP relaxation.

In [19] the author focused on the centralization of NFV, which leads to cost reduction, and decentralization, which leads to performance and security. They argue that placing NVF in multiple data centers would provide resilience and an alternate traffic path. Still, this approach will not be suitable in every case, such as for those with fewer data centers. Resilience is not required for all network functions, such as DHCP, which can tolerate a few hours of downtime. There are additional issues in this model: when traffic between the service provider and the service consumer increases, it may not function as expected. Yo, future tackle growth: this model will require a more advanced approach and emerging technologies to improve its area of work. They are more motivated to save energy by placing unused physical hosts into shutdown mode and by efficiently placing VNFs. They demonstrate that energy-aware function placement, scheduling, and chaining algorithms can reduce energy consumption by lowering CPU speeds and partially disabling hardware components. The NFV placement algorithm should be intelligent enough to place the VNF where it suits most based on the service response. Efficient algorithms should determine on which physical resources (servers) network functions are placed and be able to move tasks from one server to another for objectives such as load balancing, energy savings, recovery from failures, etc.

3.2 NFV Placement in a Hybrid Environment

NFV is an upcoming paradigm in network function virtualization that provides greater flexibility and scalability. Still, its success depends on the algorithm's performance, which determines where it is instantiated. The author has focused on the issue that NFV is designed for end-to-end networks, not only for data centers. Outside the data center network, network constraints, such as bandwidth and latency, would directly affect NFV. Another issue the author addresses is that, traditionally, network functions have been implemented on expensive dedicated hardware, which offers better performance and efficiency than a virtualized network function. The author has suggested that base NF should run on dedicated hardware (private cloud) and burst NF should run in the virtualized environment (public cloud) to meet service demand. Nevertheless, there should be a service-aware management algorithm for managing NFV in both environments. The model was ILP-based and was suitable for small environments and limited-service types. For various requests and service types, the model still requires further attention. However, the algorithm's performance is better [20].

Figure 2 describes the scenario where a virtualized network function has deployed in a physical device. This occurs when virtualized infrastructure is unavailable or when VNFs have not been adopted. Figure 3 describes the scenario in which a virtualized network function is deployed on commodity hardware using a virtualization platform. The author has proposed a model to determine suitable VNF placement in both environments. The proposed VNF placement approach is simple in nature but complex to apply in real networks. The study employs the second approach for evaluation. In [21], a more comprehensive and realistic approach was proposed for the VNF placement in a hybrid environment. The author's goal in their model is to minimize total bandwidth consumption and overall link utilization.

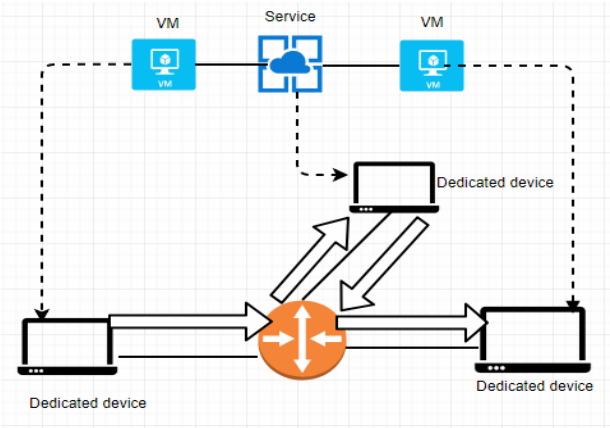

Fig 2: VNF on Dedicated Devices

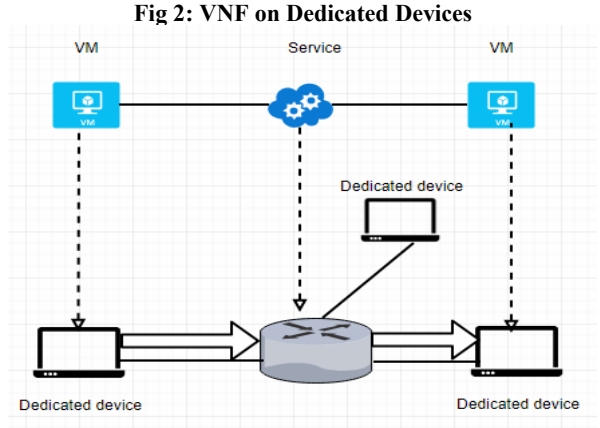

Fig 3: VNF on Commodity Hardware

3.3 Traffic-Aware NFV Placement

The placement of Network Functions Virtualization (NFVs) in cloud and cloud–edge environments has attracted significant research attention in recent years, particularly amid increasing demand for traffic-aware, latency-sensitive, and scalable network services. In recent years, NFV research has evolved from static optimization models to dynamic, intelligent, and distributed placement strategies that consider traffic variability, service chaining, and real-time orchestration. A comprehensive

summary in the literature is given in Table 1, whereas Table 2 shows the key Parameters Used in Traffic-Aware NFV Placement.

Several studies have focused on traffic- and latency-aware placement of NFV and Service Function Chains (SFCs)**.** In [22], a latency-aware NFV chain deployment framework was proposed that jointly considers path selection and NFV placement, with particular emphasis on edge environments. Their approach highlights the importance of end-to-end delay constraints in traffic-sensitive applications. Similarly, [23] addressed traffic-aware placement for parallelized service function chains, demonstrating that parallel execution of VNFs can significantly improve performance under high traffic loads. However, such approaches often prioritize latency over other factors, including energy efficiency and fault tolerance.

To address scalability and traffic growth, researchers have explored replication, elasticity, and dynamic scaling, and, in [24], investigated VNF placement with replication for SFCs to minimize bandwidth consumption and service response time. Their results show that replication can effectively handle traffic surges, although it introduces additional resource overhead. In [25], this line of work was further advanced by incorporating dynamic instance scaling into SFC placement, enabling NFVs to adapt to fluctuating traffic demands. Despite their effectiveness, these approaches rely heavily on accurate traffic prediction and may suffer from control overhead in highly dynamic environments. Another active research direction involves traffic-aware migration and service restoration**.** In [26], a traffic-aware NFV migration strategy was proposed to restore services in the event of network failures. By combining an ILP formulation with a heuristic algorithm, their approach achieves near-optimal performance while significantly reducing computational time.

Table 1: Summary of Traffic-Aware NFV Placement in the Literature

Reference	Method	Parameters	Results	Limitations of Research
[27]	Latency-aware VNF chain deployment (two-stage; path selection + deployment)	Latency constraints, path selection, edge NFV chain deployment	Proposes a latency-aware deployment scheme for VNF chains (edge-centric) with efficient path selection	Focused primarily on latency; other factors (e.g., security, failures, energy) typically not central
[24]	VNF placement + replica placement for SFCs (VNFRP)	Bandwidth consumption, service response time, overall cost; replica decisions	Introduces a placement + replication approach to improve SFC provisioning efficiency	Replication overhead/cost trade-offs are scenario-dependent; scalability/real-time adaptation may be limited
[23]	Traffic-aware placement of *parallelized* SFCs	Parallelized chains, traffic-aware placement objectives	Addresses traffic-aware placement for parallelized SFC scenarios (well-cited in this area)	Access to full experimental details may require subscription; implementation assumptions vary
[28]	Security-aware SFC deployment: ILP model + consolidation + Viterbi-based path selection	Server security levels, node load, hosting capacity, resource constraints; minimize delay; link load constraints	Improves acceptance ratio and reduces transmission delay; load balancing considered in deployment	Authors state it focuses on SFC deployment with security demand and does not consider cyber-attacks explicitly
[29]	Survey on VNF placement problem (VNFPP)	VNFPP definition, classifications, constraints/objectives, solution families	Solid synthesis of placement methods and modeling patterns for research framing	Survey; doesn't itself provide traffic-aware algorithmic contributions
[30]	Robustness-aware VNF placement + request scheduling ("Reveal")	Robustness constraints (e.g., failures/malicious behavior), edge-cloud request handling	Targets robustness-aware placement/scheduling beyond classic delay/throughput-only goals	Still constrained by modeled threat/failure assumptions; practical deployment details may vary
[25]	Dynamic SFC placement with instance scaling	Scaling/elasticity of VNF instances, dynamic service demand	Tackles dynamic SFC placement with scaling to respond to varying load	May require accurate demand estimation; control overhead and stability can be challenging
[31]	SFC methodology for ultra-low latency NFV networks	Latency minimization; physical resource utilization	Proposes SFC methodology prioritizing very low latency services	Ultra-low-latency focus may under-emphasize cost/energy/robustness trade-offs
[26]	Function traffic-aware VNF migration for service restoration: ILP	Traffic volume dynamics per function, migration cost, restoration under failures	Heuristic close to optimal ILP with much lower runtime; beats simulated	Restoration-centric (failure-driven) scenario; performance depends on failure/traffic model

	+ heuristic		annealing in cost and time	realism
[32]	Distributed SFC deployment: game-theoretic learning	Minimum-cost SFC deployment; NFVI access via edge routers; distributed decision-making	Addresses NP-hard SFC deployment with distributed learning approach	Game-theoretic methods can face convergence/stability concerns and require careful incentive modeling
[33]	GNN + DRL for VNF placement to enhance [33]SFC fault tolerance (GRL-SFT)	MDP formulation; acceptance ratio + completion delay; chain graph representation	Improves fault tolerance with fast decision-making and real-time restoration orientation	DRL generalization depends on training environment; explainability and reproducibility are common gaps
[34]	Joint VNF deployment + SFC scheduling in cloud–edge: BILP + heuristics (dual-time slot–frame)	Cloud–edge resource limits, deployment cost, response delay; Dijkstra-based path selection; queuing-based execution time	Reported improvements vs baselines: lower resource consumption, lower delay, higher acceptance rate	Simulation-based; effectiveness depends on assumed cloud–edge topology/workload and cost models
[35]	DRL for context-aware online VNF placement & migration	Online policy updates; placement/migration control	Uses DRL to continuously update policies for placement/migration in distributed settings	DRL training data/simulator fidelity matters; some details may be behind ACM access limits
[36]	Availability-aware VNF placement + routing: constrained optimization + randomized rounding	Multi-dimensional resources (CPU/RAM/uplink/downlink), reliability/latency for uRLLC	Polynomial-time approximation approach for NP-hard placement+routing in MEC-enabled 5G	Work-in-progress framing; practical deployment considerations may be limited
[37]	Predictive auto-scaling for energy/SLA-aware VNF placement	Predictive scaling + SLA/energy goals	Useful as a 2026-direction pointer (predictive + autoscaling trend)	ResearchGate-indexed / early visibility: treat as non-primary until peer-reviewed source is confirmed

This line of work emphasizes the importance of traffic characteristics at the function level rather than treating VNFs uniformly. However, such solutions are often evaluated under specific failure scenarios, limiting their generalizability.

Recent studies have increasingly leveraged machine learning and reinforcement learning (RL) techniques for traffic-aware NFV orchestration. In [30], a robustness-aware NFV placement and scheduling framework was introduced to handle failures and malicious behaviours in edge-cloud environments. More advanced learning-based approaches include [33], which employed Graph Neural Networks (GNNs) combined with Deep Reinforcement Learning (DRL) to enhance fault tolerance in SFC deployment. In [35], the authors further explored DRL-based online NFV placement and migration, thereby enabling real-time adaptive decision-making. While these AI-driven methods demonstrate superior adaptability and performance, their effectiveness depends on the quality of the training data, the accuracy of environment modeling, and the stability of convergence.

Table 2: Key Parameters Used in Traffic-Aware NFV Placement Studies

Parameter Category	Parameter	Description / Role in Placement	Representative Studies
Traffic Parameters	Traffic volume / flow rate	Amount of traffic processed by each VNF or SFC	[23], [24], [26], [34]
	Traffic variability / burstiness	Temporal variation in traffic demand	[25], [26], [35]
	Traffic locality	Proximity of traffic sources to VNFs	[22], [27], [31]
	Traffic prediction	Forecasting future demand for proactive placement	[25], [37]
Latency & QoS	End-to-end latency	Total delay across service function chain	[22], [27], [31], [34]
	Processing delay	Execution time of VNFs	[23], [31]
	SLA constraints	QoS guarantees (delay, reliability)	[28], [36]

Resource Constraints	CPU capacity	Processing capability of hosting nodes	[24], [28], [34]
	Memory (RAM)	Memory requirements per VNF	[24], [36]
	Bandwidth	Link capacity between VNFs	[24], [26], [34]
	Storage	VNF image and state requirements	[28], [36]
Topology & Architecture	Network topology	Physical/logical network graph	[27], [26], [34]
	Cloud–edge location	Placement domain (cloud vs edge)	[22], [31], [34]
	VNF chaining order	Sequential or parallel SFC structure	[23], [31]
Optimization Objectives	Latency minimization	Reduce service response time	[22], [27], [31]
	Bandwidth minimization	Reduce overall traffic load	[24], [34]
	Cost minimization	Deployment and operational cost	[30], [34]
	Energy efficiency	Reduce power consumption	[14], [37]
	Acceptance ratio	Maximize served requests	[27], [33]
Reliability & Security	Availability	VNF/service uptime requirements	[36]
	Fault tolerance	Ability to handle failures	[29], [33]
	Security level	Trust or security capability of nodes	[27]
Adaptation & Dynamics	VNF migration cost	Cost and overhead of moving VNFs	[31], [35]
	Scaling elasticity	Dynamic instantiation/removal of VNFs	[25], [37]
	Online decision making	Real-time placement decisions	[33], [35]
Solution Methodology	ILP / BILP variables	Exact optimization decision variables	[24], [34], [34]
	Heuristic parameters	Greedy or approximation controls	[31], [32]
	Learning state/action space	RL or GNN-based decision inputs	[33], [35]

In parallel, researchers have examined distributed and game-theoretic approaches to traffic-aware SFC deployment. [32] proposed a distributed learning-based framework using game theory to solve the NP-hard SFC deployment problem. This approach reduces the overhead of centralized control and improves scalability but may face convergence and incentive-design challenges in real-world deployments. The integration of cloud–edge computing and ultra-low latency services has further influenced traffic-aware NFV placement research. In [31], a methodology for ultra-low latency NFV networks was proposed, emphasizing physical resource utilization and delay minimization, and in [34], addressed joint NFV deployment and SFC scheduling in cloud–edge environments using a bi-level integer linear programming (BILP) model combined with heuristics, reporting improvements in acceptance ratio and response delay. However, these solutions are often validated through simulations and depend heavily on assumed network topologies and workload models.

Security and reliability have also been incorporated into traffic-aware placement decisions. In [28], a security-aware SFC deployment framework was proposed that uses an ILP model and Viterbi-based path selection, accounting for server security levels and network load. Similarly, in [36], the authors focused on availability-aware NFV placement and routing for uRLLC services in 5G networks, addressing multi-dimensional resource constraints. Although these works enhance reliability and security, they typically do not address proactive attack detection or long-term adaptability.

Finally, several surveys and benchmarking studies provide valuable context and highlight open research challenges. In [38] and [29], comprehensive surveys on SFC and NFV placement, respectively, were presented, outlining classification schemes, optimization objectives, and solution techniques. These surveys emphasize unresolved challenges, including real-time traffic prediction, cross-layer optimization, energy-aware placement, and explainable AI-based orchestration. Emerging tools such as the Virne benchmark framework further underscore the need for standardized evaluation environments for learning-based NFV solutions.

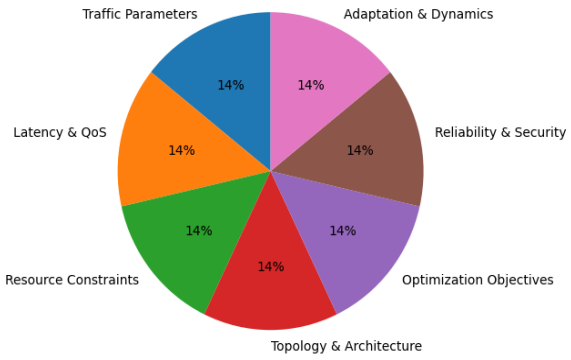

Fig 4: Distribution of Parameter Categories in Traffic-Aware NFV

Figure 4 presents the category-wise distribution of NVF parameters used in the literature, and Figure 5 illustrates a taxonomy of traffic-aware NFV placement approaches based on four primary dimensions: solution methodology, architectural paradigm, adaptation strategy, and optimization

objectives. Existing research can be broadly categorized into optimization-based and AI-driven approaches. Optimization-based methods, such as ILP, BILP, and heuristic algorithms, provide mathematically rigorous formulations but face scalability challenges in large-scale deployments. In contrast, AI-based approaches, including reinforcement learning and graph neural networks, enable adaptive, real-time placement decisions but raise concerns about explainability and training stability. Architecturally, solutions range from centralized orchestration models to distributed and hybrid cloud–edge frameworks. Furthermore, placement strategies have evolved from static optimization to dynamic scaling and predictive migration mechanisms. Finally, optimization objectives extend beyond latency minimization to include energy efficiency, reliability, security, and multi-objective trade-offs. This taxonomy highlights the multi-dimensional nature of traffic-aware NFV placement and underscores the need for integrated, scalable, and intelligent frameworks.

3.4 Comparative Analysis of Traffic-Aware NFV Placement Approaches

A closer examination of Table 1 reveals several important trends in traffic-aware NFV placement research. First, there is a clear methodological shift from traditional optimization-based models to AI-driven approaches. Early works predominantly relied on Integer Linear Programming (ILP), BILP, MIQCP, and related exact formulations to achieve optimal or near-optimal placement decisions under constrained environments. These methods provide strong mathematical rigor and clear objective formulation; however, they suffer from scalability limitations and high computational complexity in large-scale cloud–edge networks. In contrast, recent approaches increasingly employ machine learning, deep reinforcement learning (DRL), and graph neural networks (GNNs) to enable adaptive and real-time placement decisions. While AI-based methods demonstrate superior scalability and responsiveness to traffic variability, they introduce challenges related to training data dependency, convergence stability, and explainability.

Second, the literature shows a transition from centralized placement architectures toward distributed and hybrid cloud–edge orchestration models. Centralized solutions typically assume a global controller with full network visibility, enabling globally optimized placement decisions. However, such models may face bottlenecks and single points of failure in large-scale deployments. Distributed and game-theoretic approaches aim to improve scalability and resilience by allowing edge nodes or local controllers to make decentralized decisions. Although distributed strategies reduce control overhead and improve fault tolerance, they often encounter coordination complexity, convergence issues, and reduced global optimality.

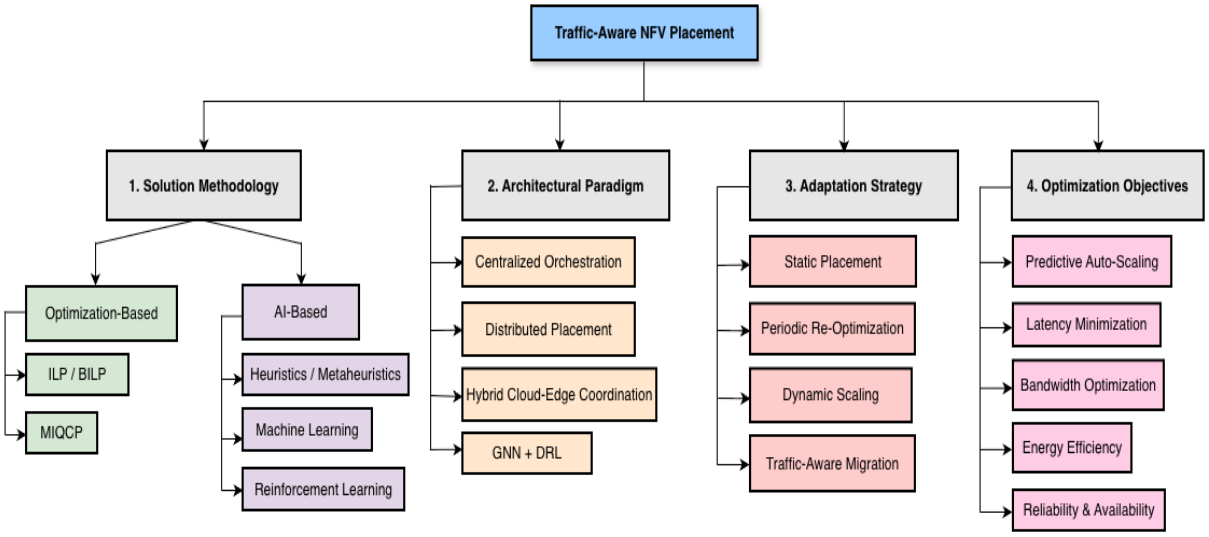

Fig 5: Taxonomy of Traffic-Aware NFV Placement Approaches

Third, the evolution from static placement models to dynamic and adaptive strategies is evident. Static approaches assume fixed traffic matrices or periodic updates and focus primarily on initial placement optimization. However, modern networks require dynamic scaling, migration, and real-time adaptation due to traffic burstiness and service elasticity requirements. Recent works integrate dynamic instance scaling, predictive auto-scaling, and online learning mechanisms to handle fluctuating demand. Despite these advancements, many dynamic models rely heavily on accurate traffic prediction and may incur additional migration overhead or control signaling costs.

Overall, the comparative trends indicate a gradual movement toward intelligent, distributed, and adaptive NFV placement frameworks. Nevertheless, no single approach fully resolves the trade-offs among scalability, optimality, explainability, and deployment feasibility. This observation further underscores the need for holistic, traffic-aware NFV placement models that integrate rigorous optimization with adaptive intelligence and cloud–edge coordination.

4. TSUMMARY AND RESEARCH GAP

Overall, recent literature demonstrates a clear shift toward traffic-aware, dynamic, and intelligent NFV placement in cloud and cloud–edge environments. While significant progress has been made in latency reduction, scalability, and adaptability, open challenges remain in unified multi-objective optimization, realistic traffic modeling, explainability of AI-driven decisions, and real-world deployment validation. These gaps motivate further research into holistic traffic-aware NFV placement frameworks that can jointly address performance, reliability, energy efficiency, and operational complexity. This paper surveyed recent work in traffic-aware placement of Network Function Virtualization in cloud environments. While significant progress has been made, several gaps remain that

offer promising avenues for future research:

Dynamic Scaling and Optimization: Most existing solutions focus on initial VNF placement. Research into algorithms that dynamically adapt VNF placement and resource allocation in response to real-time traffic fluctuations is crucial. This could involve machine learning techniques to predict traffic patterns and proactively adjust deployments.

Security Considerations: The paper primarily focuses on performance aspects. Future work should address the security implications of NFV placement, such as secure VNF instantiation, secure communication between chained VNFs, and protection against attacks targeting virtualized network functions.

Multi-tenancy and Resource Sharing: Efficiently sharing resources among multiple tenants while guaranteeing performance isolation is a challenge. Developing strategies for fair and efficient resource allocation in multi-tenant NFV environments is essential.

Integration with Emerging Technologies: Exploring the synergy among NFV, edge computing, blockchain, and intent-based networking presents exciting research opportunities. For instance, how can edge computing be leveraged to improve the performance and security of NFV deployments?

4.1 Issues and Open Challenges in Traffic-Aware NFV Placement

Despite substantial progress in traffic-aware NFV and Service Function Chain (SFC) placement, several **critical issues and open challenges** remain unresolved. Recent studies reveal that existing solutions often optimize a limited subset of objectives and rely on simplified assumptions that restrict their applicability in real-world cloud environments.

Dynamic and Unpredictable Traffic Patterns: Most traffic-aware NFV placement models assume either static or periodically updated traffic demands. However, real network traffic exhibits high temporal variability and burstiness, especially in cloud–edge and 5G environments. Although dynamic scaling and migration approaches have been proposed, they heavily depend on accurate traffic prediction models. Designing robust placement mechanisms that can adapt to unpredictable traffic fluctuations in real time remains an open challenge.

Multi-Objective Optimization Complexity: Traffic-aware NFV placement inherently involves conflicting objectives, including minimizing latency, bandwidth consumption, energy usage, and deployment cost, while maximizing reliability and acceptance ratio. Many existing works address only one or two objectives, leading to suboptimal trade-offs. Developing scalable multi-objective optimization frameworks that balance performance, cost, and sustainability remains a significant research challenge.

Scalability and Computational Overhead: Exact optimization approaches such as ILP, BILP, and MIQCP provide near-optimal solutions but are computationally infeasible for large-scale networks**.** Although heuristics and approximation algorithms reduce complexity, they often sacrifice optimality or lack performance guarantees. Ensuring scalable, near-optimal solutions suitable for large-scale cloud and cloud–edge infrastructures remains an open problem.

Integration of AI and Explainability: Recent studies increasingly adopt machine learning and deep reinforcement learning for traffic-aware NFV orchestration. While these approaches demonstrate promising adaptability and performance improvements, they introduce new challenges related to data dependence in training, convergence stability, and explainability. The absence of interpretable decision-making limits trust and adoption in operational networks. Hence, explainable and reliable AI-driven NFV placement remains an open research direction.

Traffic-Aware Migration and Service Continuity: VNF migration is essential for load balancing, failure recovery, and SLA assurance. However, migration itself generates additional traffic and service disruption. Existing solutions often consider migration cost in isolation or under specific failure scenarios. Designing traffic-aware migration strategies that minimize disruption while ensuring service continuity under dynamic network conditions is still challenging.

Cloud–Edge Coordination and Heterogeneity: With the rise of edge computing, NFV placement must operate across heterogeneous cloud–edge infrastructures with diverse resource capabilities and latency constraints. Coordinating traffic-aware placement decisions across centralized clouds and distributed edge nodes, while maintaining global optimality, remains insufficiently addressed. Unified cloud–edge orchestration models are still in their infancy.

Security, Reliability, and Availability Awareness: Although some recent works incorporate security levels and availability constraints, security-aware traffic modeling and proactive attack mitigation are rarely integrated into NFV placement decisions. Additionally, most reliability-aware approaches focus on redundancy rather than predictive fault prevention. Developing holistic placement frameworks that jointly consider traffic, security, and resilience remains an open challenge.

Lack of Realistic Evaluation and Standard Benchmarks: Most traffic-aware NFV placement studies rely on simulations with synthetic traffic and idealized network topologies. The lack of real-world datasets, standardized benchmarks, and reproducible evaluation frameworks makes fair comparison difficult. This gap highlights the need for benchmark-driven and deployment-oriented evaluation methodologies.

The above challenges indicate that existing traffic-aware NFV placement solutions are fragmented and often context-specific. There is a clear need for a holistic, scalable, and intelligent NFV placement framework that can dynamically adapt to traffic variations, support cloud–edge environments, ensure reliability and security, and remain computationally efficient for real-world deployment.

5. CONCLUSION

Network Function Virtualization has emerged as a cornerstone technology for enabling flexible, scalable, and cost-effective deployment of network services in modern cloud and cloud–edge environments. This paper presented a comprehensive review of traffic-aware VNF placement strategies, highlighting how traffic characteristics, service function chaining, and deployment architectures critically influence network performance. Classical optimization techniques such as ILP, BIP, and heuristic methods were analyzed alongside recent AI-driven and learning-based approaches, revealing clear trade-offs between optimality, scalability, adaptability, and deployment feasibility.

The survey indicates that, although significant progress has been made in reducing latency, improving resource utilization, and supporting dynamic traffic demands, several unresolved challenges remain. Existing approaches often rely on static or simplified traffic assumptions, face scalability limitations in large-scale cloud–edge networks, or lack explainability and

robustness when adopting AI-based solutions. Moreover, the absence of standardized benchmarks and real-world evaluation environments limits the reproducibility and operational validation of many proposed techniques.

Future research directions should therefore focus on the following key aspects:

Dynamic and Predictive Traffic-Aware Placement: Future work should develop placement frameworks that continuously adapt to real-time and unpredictable traffic variations. Integrating traffic prediction models with online optimization or learning-based orchestration can enable proactive scaling, migration, and placement decisions.

Multi-Objective Optimization Frameworks: There is a need for unified frameworks that simultaneously optimize latency, bandwidth usage, energy consumption, cost, reliability, and security. Hybrid approaches that combine rigorous optimization with adaptive intelligence may provide practical trade-offs between performance and scalability.

Explainable and Trustworthy AI-Driven NFV Orchestration: While machine learning and reinforcement learning approaches show strong adaptability, their lack of transparency hinders adoption. Future systems should incorporate explainable AI techniques to make placement decisions interpretable and verifiable for network operators.

Cloud–Edge Coordination and Heterogeneity Awareness: As NFV increasingly spans centralized clouds and distributed edge nodes, future research must address coordinated placement across heterogeneous infrastructures to ensure global efficiency while respecting local constraints.

Security- and Reliability-Aware Traffic Modeling: Placement decisions should move beyond performance metrics to incorporate proactive security, availability, and resilience considerations, including failure prediction, attack awareness, and service continuity guarantees.

Benchmark-Driven and Deployment-Oriented Evaluation: Finally, the community would benefit from standardized benchmarks, realistic traffic datasets, and open evaluation platforms to ensure fair comparison and accelerate real-world deployment of traffic-aware NFV solutions.

Overall, traffic-aware NFV placement remains a dynamic and evolving research area. Addressing these future directions will be essential to building intelligent, scalable, and deployment-ready NFV orchestration frameworks that support next-generation network services.

6. REFERENCES

[1] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 1, pp. 236-262, 2016.

[2] I. Maity, G. Giambene, T. X. Vu, C. Kesha, and S. Chatzinotas, "Traffic-aware resource management in sdn/nfv-based satellite networks for remote and urban areas," *IEEE Transactions on Vehicular Technology,* vol. 73, no. 11, pp. 17400-17415, 2024.

[3] Q. Zhang, Y. Xiao, F. Liu, J. C. Lui, J. Guo, and T. Wang, "Joint optimization of chain placement and request scheduling for network function virtualization," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, 2017: IEEE, pp. 731-741.

[4] S. Tomaszek, R. Speith, and A. Schürr, "Virtual network embedding: ensuring correctness and optimality by construction using model transformation and integer linear programming techniques," *Software and systems modeling,* vol. 20, no. 4, pp. 1299-1332, 2021.

[5] M. G. A. Bekhit, *Resource Allocation and Optimal Scheduling of Virtual Network Functions in Software Defined Networks*. University of Technology Sydney (Australia), 2020.

[6] G. Liu, S. Guo, B. Li, and C. Chen, "Joint traffic-aware consolidated middleboxes selection and routing in distributed SDNs," *IEEE Transactions on Network and Service Management,* vol. 18, no. 2, pp. 1415-1429, 2020.

[7] L. A. Gallego Pareja, J. M. López-Lezama, and O. Gómez Carmona, "A mixed-integer linear programming model for the simultaneous optimal distribution network reconfiguration and optimal placement of distributed generation," *Energies,* vol. 15, no. 9, p. 3063, 2022.

[8] W. He, S. Guo, Y. Liang, and X. Qiu, "Markov approximation method for optimal service orchestration in IoT network," *IEEE Access,* vol. 7, pp. 49538-49548, 2019.

[9] S. Dräxler, H. Karl, and Z. Á. Mann, "Jasper: Joint optimization of scaling, placement, and routing of virtual network services," *IEEE Transactions on Network and Service Management,* vol. 15, no. 3, pp. 946-960, 2018.

[10] S. Yang, F. Li, S. Trajanovski, R. Yahyapour, and X. Fu, "Recent advances of resource allocation in network function virtualization," *IEEE Transactions on Parallel and Distributed Systems,* vol. 32, no. 2, pp. 295-314, 2020.

[11] Y. Mao, J. Zhang, and K. B. Letaief, "A Lyapunov optimization approach for green cellular networks with hybrid energy supplies," *IEEE Journal on Selected Areas in Communications,* vol. 33, no. 12, pp. 2463-2477, 2015.

[12] M. Alicherry and T. Lakshman, "Network aware resource allocation in distributed clouds," in *Infocom, 2012 proceedings IEEE*, 2012: IEEE, pp. 963-971.

[13] K. Kawashima, T. Otoshi, Y. Ohsita, and M. Murata, "Dynamic placement of virtual network functions based on model predictive control," in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*, 2016: IEEE, pp. 1037-1042.

[14] S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca, "The dynamic placement of virtual network functions," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*, 2014: IEEE, pp. 1-9.

[15] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine,* vol. 53, no. 2, pp. 90-97, 2015.

[16] M. Sasabe and T. Hara, "Capacitated shortest path tour problem-based integer linear programming for service chaining and function placement in NFV networks," *IEEE Transactions on Network and Service Management,* vol. 18, no. 1, pp. 104-117, 2020.

[17] J. López, N. Kushik, and D. Zeghlache, "Virtual machine placement quality estimation in cloud infrastructures using integer linear programming," *Software Quality Journal,* vol. 27, no. 2, pp. 731-755, 2019.

[18] J. Crichigno, D. Oliveira, M. Pourvali, N. Ghani, and D. Torres, "A routing and placement scheme for network function virtualization," in *Telecommunications and Signal Processing (TSP), 2017 40th International Conference on*, 2017: IEEE, pp. 26-31.

[19] U. Fiore, P. Zanetti, F. Palmieri, and F. Perla, "Traffic matrix estimation with software-defined NFV: Challenges and opportunities," *Journal of computational science,* vol. 22, pp. 162-170, 2017.

[20] H. Moens and F. De Turck, "VNF-P: A model for efficient placement of virtualized network functions," in *10th International Conference on Network and Service Management (CNSM)*, 2014, pp. 418-423.

[21] J. Cao, Y. Zhang, W. An, X. Chen, Y. Han, and J. Sun, "Vnf placement in hybrid nfv environment: Modeling and genetic algorithms," in *Parallel and Distributed Systems (ICPADS), 2016 IEEE 22nd International Conference on*, 2016: IEEE, pp. 769-777.

[22] Y. Jin, "Latency-Aware Deployment of Service Function Chains at the Network Edge," *IEEE Transactions on Network and Service Management,* 2020.

[23] X. Wang, "Traffic-Aware Placement of Parallelized Service Function Chains," *ACM SIGCOMM Computer Communication Review,* 2021.

[24] M. Abdelaal, "VNF Placement and Replica Placement for Service Function Chains," *IEEE Access,* 2021.

[25] Y. Li, "Dynamic Service Function Chain Placement with VNF Instance Scaling," *Future Generation Computer Systems,* 2023.

[26] T. Pham and D. Nguyen, "Traffic-Aware Virtual Network Function Migration for Service Restoration in NFV Networks," *Computer Communications,* 2024.

[27] P. Jin, X. Fei, Q. Zhang, F. Liu, and B. Li, "Latency-aware VNF chain deployment with efficient resource reuse at network edge," in *IEEE INFOCOM 2020-IEEE conference on computer communications*, 2020: IEEE, pp. 267-276.

[28] X. Zhai, "Security-Aware Service Function Chain Deployment in NFV Networks," *Scientific Reports,* 2022.

[29] J. Sun, "A Survey on the Virtual Network Function Placement Problem," *Computer Networks,* 2022.

[30] F. Fang, "Reveal: Robustness-Aware Placement and Scheduling of Virtual Network Functions in Edge–Cloud Networks," *Computer Networks,* 2023.

[31] M. Erbati, "An Ultra-Low Latency Service Function Chaining Methodology for NFV Networks," *Sensors,* 2023.

[32] M. Alikhani, "A Distributed Learning-Based Approach for Cost-Efficient Service Function Chain Deployment," *Future Generation Computer Systems,* 2024.

[33] F. Ros, "GRL-SFT: Graph Reinforcement Learning for Fault-Tolerant Service Function Chain Placement," *Applied Sciences,* 2024.

[34] Y. Teng, "Joint Deployment and Scheduling of Service Function Chains in Cloud–Edge Computing Environments," *Journal of Network and Computer Applications,* 2025.

[35] D. Wassie, "Context-Aware Online VNF Placement and Migration Using Deep Reinforcement Learning," *ACM Transactions on Internet Technology,* 2025.

[36] M. Sayeed and S. Bera, "Availability-Aware Virtual Network Function Placement and Routing for uRLLC Services," *arXiv,* 2025.

[37] R. Nikbazm, "Predictive Auto-Scaling for Energy- and SLA-Aware VNF Placement," *Future Internet,* 2026.

[38] K. Kaur *et al.*, "A Comprehensive Survey on Service Function Chaining: State of the Art and Research Challenges