International Journal of Network Dynamics and Intelligence Scilight

Article

The Promise of Applying Machine Learning Techniques to Network Function Virtualization

Houda Jmila [1], Mohamed Ibn Khedher [2,*], and Mounim A. El-Yacoubi [3]

[1] Institute LIST, CEA, Paris-Saclay University, 91190 Palaiseau, France
[2] IRT-SystemX, 2 Bd Thomas Gobert, 91120 Palaiseau, France
[3] Samovar, Telecom SudParis, Institut Polytechnique de Paris, 19 place Marguerite Perey, 91120 Palaiseau, France
[*] Correspondence: mohamed.ibn-khedher@irt-systemx.fr

Received: 28 December 2023
Accepted: 15 August 2024
Published: 24 December 2024

Abstract: "Network Function Virtualization" (NFV) is an emerging technology and 5G key enabler. It promises operating expenditure savings and high flexibility in managing the network by decoupling the network functions, like firewalls, proxies etc., from the physical equipments on which they run. In order to reap the full benefits of NFV, some challenges still need to be overcome, namely those related to resource management, security and anomaly detection. Recently, Machine learning (ML) has been applied in different fields and has demonstrated amazing results. Utilizing Machine learning to address the challenges faced by NFV is a promising research field that requires further investigation. In this paper, we shed light on this domain by discussing the potential and challenges of ML application to NFV and by surveying existing works.

Keywords: machine learning; network function virtualization (NFV); software defined network (SDN); cloud computing; resource management; traffic classification; survey

1. Introduction

Network Function Virtualization [1–3] has recently gained increading attention as this technology allows for greater network flexibility and time/cost reduction to introduce new services. Thanks to NFV, network functions, like load balancers, WAN optimizers and Intrusion Detection Systems (IDSs), that were traditionally provided by dedicated and special-purpose hardware can now be implemented by software running on virtual machines or containers in a cloud computing infrastructure. In this way, these Virtualized Network Functions (VNFs) can be relocated and instantiated at different regions without requiring the purchase and installation of new hardware [4, 5]. Moreover, by separating the software from the hardware, the infrastructure resources can be shared and reassigned efficiently to allow faster deployment of network services over the same physical platform.

Despite the excessive speed at which NFV is being accepted by both academia and industry, this technology is still in its infancy and faces critical issues. Main challenges concern resource management and orchestration, security and fault tolerance in addition to energy savings [6]. The use of ML techniques to solve such problems is an effervescent and attractive research field. Machine learning is a branch of artificial intelligence that allows computer systems to learn directly from examples and data. It helps making sense of the available data by extracting valuable information. Thanks to recent advances in network architecture and telemetry, the modeling and implementation of ML techniques in the network environment has become easier and more efficient. Specifically, the Software Defined Network (SDN) paradigm [7] decouples the "data plane" from the "control plane" and thus enables central control of the network [8]. Current data plane elements (routers, switches etc.) are equipped with more powerful storage and computing techniques able to gather richer data monitoring view of the network [9]. These conditions ease and ameliorate learning about the network to better supervise it.

The purpose of this paper is to discuss the benefits and limitation of applying such techniques in the NFV ecosystem by surveying existing approaches. The objective is to alert the research community to the potential role that ML could play in resolving problems related to NFV. Note that many good surveys have already investigated the

Copyright: © 2024 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license https://creativecommons.org/licenses/by/4.0/.

ML applications to different domains linked to NFV like the Cloud Computing in [10–14], the SDN [15], the Big data in [16, 17] the Internet of Things in [18, 19] and for networking in general [20]. However, applying machine learning to NFV and adapting them to this technology still requires further study.

Our study reveals that current ML applications to NFV focus mainly on *i*) resource management and *ii*) anomaly detection. This paper provides an extensive and comprehensive survey of existing contributions and discusses potential opportunities for ML. We belief that this work can be useful to the research communities of both Machine Learning and Networks.

As showed on the road-map below, this paper is structured in seven parts (Figure 1): After introducing the NFV and machine learning concepts in sections 2 and 3, we move, in sections 4 and 5, to analyzing ML application on two main challenges faced by NFV which are efficient resource management and fault/anomaly detection. Section 6 provides a fruitful discussion on challenges and opportunities on ML for NFV, and section 7 concludes the paper.

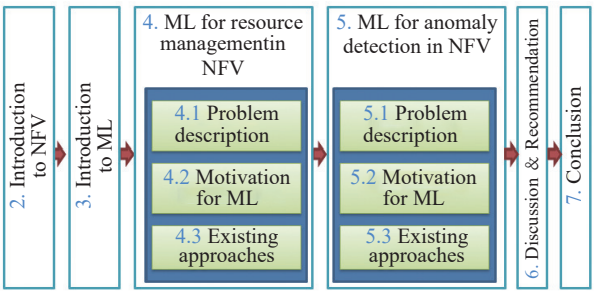

Figure 1. The road-map of this paper.

2. Introduction to NFV

In this section, we provide a short background on NFV, including relevant aspects like its architectural framework.

2.1. Principle

Network Function Virtualization decouples the physical network equipments from the functions that run on them. Hence, these network functions can be implemented as an instance of software running on one or more physical servers. Thus, a network service like a Content Delivery Network transporting live and on demand video traffic to end customers can be decomposed into a set of Virtual Network Functions (VNFs) like service classifier, firewall, anti-virus, Video optimize and parental control, that can be implemented on virtual machines running in a cloud infrastructure. These VNFs may then be relocated at different regions without the need to deploy and configure new physical equipments, or may be consolidated into high volume servers and storage. In this way, NFV promises Telecommunication Service Providers (TSPs) capital expenditures (CAPEX) and operational expenditures (OPEX) savings in addition to scalability and high flexibility in the management of the network. It also brings benefits to the users by allowing the on-demand provision and execution of customized network functions.

2.2. Architecture

The NFV concept was born on 2012 from the collaboration of a number of the world's leading TSPs with the objective of building more dynamic and service aware networks. The European Telecommunications Standards Institute (ETSI) [21] was later selected to be the home of the Industry Specification Group for NFV (ETSI ISG NFV). ETSI aims to produce requirements and potential specifications that TSPs can follow to implement efficient VNF solutions. Namely, ETSI NFV [22] defines the architecture of NFV by means of three functional entities: the Network Function Virtualization Infrastructure (NFVI), the Virtual Network Functions and the NFV Management and Orchestration (NFV MANO). These components displayed in Figure 2 are described down:

- The ***NFVI*** is the set of hardware and software components which build up the environment in which the VNFs are deployed. The physical resources provide compute, storage and network components to VNFs. These resources are abstracted through a virtualization layer. The virtualized resources may be represented as virtual machines connected by virtual links.
- A ***VNF*** is an implementation of a network function (like DHCP, firewall, etc.) on virtual resources. It may be composed of multiple internal components and thus may be deployed over different VMs. The VNFs are usually connected together to support a required service.
- The ***MANO*** is responsible for the management of the NFVI and the life cycle of the instantiated VNFs. It

includes the required functionalities for the configuration and orchestration of VNFs respecting the deployment model and the properties of the supported service.

While both industry and academia adopt NFV at remarkable speed, its development is still at early stage and many questions remain open. Different techniques are used to solve these problems and one particularly interesting and promising one is machine learning, introduced below.

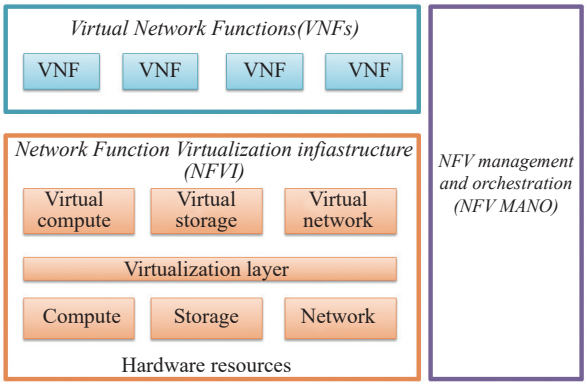

Figure 2. A simplified architecture of NFV.

3. Introduction to Machine Learning

Machine learning (ML) is an important area of artificial intelligence. ML tries to construct intelligent algorithms and models able to learn and deal with new situations without being explicitly programmed. This is achieved by extracting valuable information from the data to retrieve knowledge and intelligence. ML has reached unprecedented levels of performance in various applications, including computer vision applications [23–27], cybersecurity [28], robotics [30].

Generally, the field of machine learning is divided into three subdomains, as shown in Figure 3 : *i*) supervised learning (SL) *ii*) unsupervised learning (USL) and *iii*) reinforcement learning (RL). It is worth to say that the majority of practical ML uses supervised learning. Supervised learning and unsupervised learning are often used for data analysis while reinforcement learning is preferred for decision-making problems. Below, we give more detail for each learning mode.

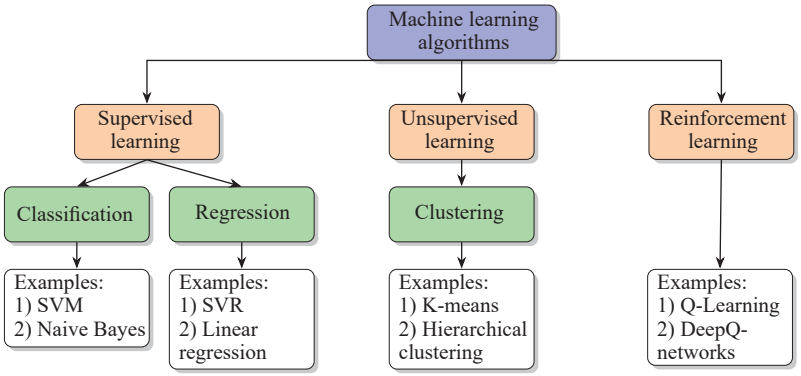

Figure 3. Classification of ML algorithms.

● ***Supervised learning*** is commonly used for applications where a labeled dataset is available. Such algorithms require data composed of *pre-labeled* inputs **X** and desired outputs **Y** to learn a mapping function f from the input to the output $\mathbf{Y}=f(\mathbf{X})$. The aim is to approximate the mapping function so that the output variables can be predicted for a query data. Learning stops when the algorithm achieves an acceptable level of performance. Supervised learning problems can be grouped into Regression and Classification taks depending on the output data type. When such data is discrete, we talk about Classification and when it is continuous, it is called Regression. Common types of classification (resp. regression) problems include traffic classification (resp. time series prediction). Popular supervised ML algorithms used for classification problems are Support Vector Machine (SVM) [31] and Naive Bayes [32]. Linear regression (LR) and support vector regression (SVR) [33] can be used for regression problems. Random Forest [34] can be used for both classification and regression.

● ***Unsupervised Learning*** This learning is suitable for problems where no labeled data is available, i.e. only input data **X** is accessible but not the corresponding output variables. These algorithms are able to discover and extract hidden patterns from **unlabeled** data without a pre-learning phase. Unsupervised learning problems are usually used for clustering problems. The K-means algorithm and Hierarchical clustering are examples of unsupervised Machine Learning.

● ***Reinforcement Learning*** [35, 36] In RL, agents learn to make better decisions directly from experience by interacting with the environment. The agents start without any knowledge about the handled, task and learn by reinforcement learning according to a reward, received based on its performance on the task. For each learning episode, the action taken by the agent leads to a change in the state of the environment, and the desirability of this change is measured by the reward. The agent's task is to maximize the overall reward it achieves throughout the learning period.

Dealing with complicated problems is one of the most important benefits of ML. Since the network field often sees complex problems that require efficient solutions, it is promising to bring ML algorithms into the network domain to leverage its amazing power for higher network performance. Existing ML applications to networks cover a large scope, ranging from traffic prediction and classification to resource management and network adaptation, including fault tolerance and intrusion detection. Nearly the same domains have been investigated in NFV. Particularly, our study reveals that resource management and anomaly detection using traffic classification are the most examined domains. In the following, we present each problem, argue the potential of ML techniques to manage it and then survey existing ML based approaches. Next, in section 6, we discuss these contributions and gives recommendations.

4. ML for Resource Management in NFV

4.1. Description of the Problem

The application of NFV introduces the problem of efficient resource management. The physical resources used to host the VNFs have a finite amount of compute, memory and storage capacity. Physical links connecting these resources have also limited amount of bandwidth. Therefore, these physical resources should be managed conveniently to gain the economical benefits promised by NFV. The resource management problem in NFV can be divided into two sub-problems: *initial VNFs placement and chaining* and *dynamic resource scaling/allocation*.

● ***The initial VNFs placement and chaining:*** In NFV, services are composed of one or more VNFs connected in a specific order to create a Service Function Chain (SFC) supporting the service. The top of Figure 4 shows an example of SFC composed of three VNFs (Firewall, Intrusion Detection System, Proxy). Each VNF requires an amount of resources to process the traffic passing through it. To deploy a SFC, an operator needs to find the right placement of VNFs into the nodes (virtual machine, container, etc.) of the physical network having enough available resources. Once the hosts are selected, the required virtual resources are created and booted to instantiate the VNFs. Then, optimal physical paths having enough bandwidth should be identified to chain the VNFs and steer the traffic between them. These two steps *i*) VNFs placement and *ii*) chaining can be tackled separately or in one shot for greater efficiency. The VNF placement and chaining is often studied with respect to different objectives like load balancing, energy conservation, compliance with the service Level of Agreement (SLA), etc. Different metrics like the mapping cost, the algorithm rapidity and complexity can be used to evaluate the efficiency of the placement process. An example of such a mapping is displayed in Figure 4.

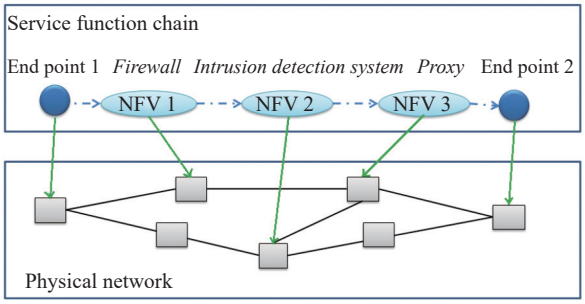

Figure 4. An example of service function chain mapping.

● ***Dynamic resource scaling/allocation*** One essential objective of NFV is to achieve dynamic resource scaling of VNFs. The traffic flowing between VNFs can fluctuate dynamically during service lifetime. To continue process-

ing it, VNFs resource needs may vary (increase or decrease) over time. Elastic resource allocation to VNFs in accordance with the varying workloads is required to continuously meet the VNF performance needs. Such elasticity can be achieved either by performing vertical scaling (increase or reduce the resources allocated to already deployed VNF instances), or horizontal scaling (increase or reduce the number of VNF instances by creating or removing instances). Allocated resources are finally freed when the lifetime of the service expires.

4.2. Motivation for the Use of ML

As highlighted in the survey [37, 38], the resource allocation problem in NFV is an NP-hard optimization problem widely addressed using exact, heuristic or meta-heuristic optimization strategies [39]. However, in the complex and dynamic VNF environment, the effectiveness of such strategies is heavily correlated with the right understanding and interpretation of the environment to complete appropriate on-line decisions. But the VNF ecosystem is very hard to explain due to the various and complicated synergies between its components, like the interaction between the VNFs, between each VNF and the hosting infrastructures, between a VNF and the processed traffic, etc. This makes the comprehension and prediction of the different factors and components behavior difficult. In this context, ML is proposed as a powerful tool able to capture all complicated and correlated synergies between various elements by extracting meaningful knowledge from data describing them. Thus, ML can be well suitable to this problem. The following section surveys existing attempts to leverage the ML power in this domain.

4.3. Survey of Existing Approaches

The are a handful of research works that applied ML for resource management in NFV. They all used a supervised algorithm to predict one or more parameters influencing the VNF placement, to enhance and accelerate the resource allocation process. Below, we give more details about these contributions. These details are summarized in Table 1.

Table 1 Summary of existing ML applications to resource management in NFV

topWorktop	topSupervised ML model:top *f*	topInput features: Xtop	topPredicted output:top topYtop=*f*(topXtop)	topDatasettop
[41]	Deep RL (DDPG)	Characteristics of the traffic entering to the VNF	Optimize placement of VNFs while addressing the enormous number of real-time traffic requests	Dataset collected from the RL environment
[40]	Deep RL (A-DDPG)	Remaining resources of the virtual links and nodes	Optimize trade-off between revenue and cost	Synthetic Dataset from the RL environment
[42, 43]	Graphic Neural Network [44]	VNF actual and past resource requirements in terms of cpu, memory and processing delay	VNF resource needs (cpu, memory and processing delay)	Data collected after implementing the open source Clearwater project [45]
[46]	Support Vector Regression [33]	Characteristics of the traffic entering to the VNF	Amount of CPU required by the VNF to process that traffic	Traces available in [47]
[48]	Deep Neural Networks [49]	VNF-level Infrastructure-level	VNF resource needs	Data collected from two VNFs
[50]	Bayesian learning [51]	Historic usage of a Cloud resource	Reliability of that Cloud resource	Data collected through simulations
[52]	Support Vector Regression [33]	Characteristic of a physical path and traffic passing through it	Delay on that physical path	Data collected through simulations

Reinforcement learning (RL) and deep reinforcement learning (DRL) have been used to successfully manage network function virtualization (NFV) resources. Reference [40] describes a DQN-based framework that dynamically orchestrates service function chains (SFCs). This framework decides where to place VNFs (in the cloud or at the edge) and how to connect them to meet real-time traffic demands. The DQN agent takes into account the traffic flow rate of services and the status of VNFs when making decisions. The framework defines the action space as the number of VNFs to activate and the traffic flow to schedule. The performance of the framework was evaluated using heterogeneous NFV/MEC-enabled IoT network scenarios emulated with the network tool. These scenarios were used to create a synthetic dataset that was helpful in training the framework's algorithms.

In [41], the authors propose an Attention mechanism-based Deep [26] Deterministic Policy Gradients (A-DDPG) framework to solve the VNF placement problem. The proposed framework utilizes the Actor-Critic network structure, where both the Actor and Critic networks employ double networks (i.e., the main network and the target network). The Actor network is responsible for learning the optimal VNF placement policy, while the Critic network evaluates the policy's performance. The Attention mechanism is integrated into the Actor network to enhance its ability to capture the complex relationships between VNFs and network resources.

Mijumbi et al. [42, 43] focused on dynamic resource scaling for already mapped VNFs during their lifetime. To

do so, they designed a Graphic Neural Network (GNN) [44] based model to predict the *VNFs resource needs*. Determining resource needs ahead of time would avoid system outages and QoS degradation due to the non-negligible delay in spinning-up (create, boot and instantiate) new resources. The authors argue that resource requirements of a VNF depend on those of its neighboring VNF since traffic flows between them. This dependency motivated them to use the GNN to predict each VNF requirements by observing its historical resource utilization and those of its neighbors. GNN is a novel connectionist approach suited for problems whose domain can be represented by a set of patterns and relationships between them. In those problems, a prediction about a given pattern can be carried out by exploiting all the related information, which includes the pattern features, and the pattern relationships. To achieve this, the authors described each VNF by a vector feature representing its actual and past required resources (memory m_n, CPU c_n, and processing delay d_n) included in a finite time horizon π:

$$f_n(t) = \begin{bmatrix} c_n(t) & m_n(t) & d_n(t) & \cdot & \cdot & \cdot \\ c_n(t-\pi) & m_n(t-\pi) & d_n(t-\pi) & & & \end{bmatrix}^T \quad (1)$$

A GNN model is then designed to reflect the topology of the star connecting the VNF to its neighbors. Feed forward Neural Network functions (FNN) are applied in different GNN layers to compute the VNF demand. To validate their proposal, the authors used a deployment of a virtualized IP Multimedia Subsystem (IMS) provided by the open source "Clearwater project" [45] and real VoIP traffic traces to construct their dataset. Results showed good prediction accuracy and an increase of the calls acceptance rate that proves that resource prediction enhanced the resource allocation process.

In our previous work [46], we applied a Support Vector Regression (SVR) based approach to estimate VNFs needs in term of CPU as a function of the processed traffic. We explained that studying the behavior of a VNF as a function of its environment helps modeling its resource requirements and favors its dynamic allocation. The SVR model was trained using a dataset provided by [47], composed of pairs $(\mathbf{TRF_i}, CPU_i)$, where $\mathbf{TRF_i}$ is a vector describing the entering traffic, and CPU_i is the amount of CPU used by the VNF to process the traffic. Experimental results showed the efficiency of the SVR model and superiority over a neural network based solution.

An interesting recent contribution is proposed in [48, 53] where the authors designed ENVI, an elastic VNF resource allocation scheme. ENVI uses neural network (NN) model to detect *resource flexing events* (increase or decrease of resource requirements) during the VNF chain lifetime. The goal is to determine the appropriate time when a VNF needs to be scaled and allocate new required resources. To construct the initial NN model, ENVI uses a combination of VNF-level features and infrastructure-level features. VNF-level features describe the VNF capacity/performance specifications like the request queue size, maximum throughput, etc. Infrastructure level features describes the resource utilization information. These features are collected periodically by the VNF monitor using internal statistics reports. The initial model is used to predict scaling events during the online resource allocation. To cope with workload variations that were not captured during offline training, ENVI continues collecting and labeling new operational data and updating the initial model periodically. The used NN model comprises four layers: an input layer, two hidden layers and an output layer to extract the dependence relationship among all features. The authors evaluated their proposal over a dataset they constructed using two VNFs and synthetically generated entering workload.

In [50], the authors dealt with the cost effective resource allocation problem in NFV. They applied a Bayesian learning method [51] to predict *the reliability* of cloud resources based on their historical usage. Reliability represents the ability of a resource to ensure constant system operation without disruption. The reliability prediction result was used to improve the performance of a Markov Decision Process applied to allocate VNFs on demand. BL is a probabilistic approach to inference, able to track the changes of resource reliability in an evolving environment. In BL, each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct. Prior knowledge can be combined with observed data to determine the final probability of a hypothesis. In [50], the BL algorithm is triggered when an NFV component is created and allocated to cloud resources. The algorithm captures resource reliability and continues the training as the time goes. Simulations were used to generate the data and evaluate the model.

The authors of [52] examined the VNF placement and chaining while minimizing the end-to-end latency. They used a SVR model to forecast *the delay* on different paths of the physical infrastructure. They argued that the solution of VNF placement based on calculations at time t can be inappropriate for time (t+1) where the VNF are effectively placed. This is due to the resources spinning-up process that can take a long time; thus the physical network state (particularly the physical paths delay) may change meanwhile. The authors proposed to predict the state at (t+1) so that the placement remains consistent with the requirements. They used a training vector that captures the parameters affecting the delay: *i*) the inference caused by resource sharing on end nodes, *ii*) the length of link connecting

them and *iii*) the traffic passing through it. To assess their approach, the authors generated a training dataset using a stochastic modeling for delay analysis of a VoIP network [54]. The results showed that the system predicts the delay rapidly and increases the number of successful service chains embedding.

Conclusion: Most of research works focused on using a supervised ML algorithm to predict an output parameter influencing the resource allocation process. The predicted parameters fall into one of the two following criteria categories: *i*) VNF future resource requirements [40, 41, 43, 46, 48] or *ii*) physical resource characteristics (reliability [50], path delay [52]). Different features were used to predict future resource requirements, namely the historical resource usage of a VNF and its neighbors [40, 41, 42, 43, 48], the workload/traffic patterns [46] and the VNF internal characteristics [48]. To validate their approach, most authors generate their own dataset by implementing a prototype as in [48], monitoring an open source VNF environment like the Clearwater project [45] or creating a synthetic dataset using the RL environment [40, 41].

5. ML for anomaly Detection in NFV

5.1. Description of the problem

Anomaly detection is an old problem, well investigated in the literature [55]. In NFV, the anomaly/fault detection has become a top priority for achieving the benefits promised by this technique. In fact, network function chains are more performance stringent than common IT applications. Detecting preliminary symptoms of service degradation is thus crucial to avoid Service Level Agreement (SLA) violation. Anomaly detection in a NFV environment consists on reporting in a timely manner any abnormal condition in an NFV service. This requires continuous examination and classification of traffic patterns related to that service.

The first step in this process is to identify the set of features which best illustrate the VNF status. This can include CPU and RAM usage, I/O operations of different partitions and network interfaces and protocols. Next, these features should be extracted from the monitoring data collected continuously from hardware components of the system and the operation system of the VNFs. These metrics are then analyzed overtime to decide whether an anomaly should be reported if the operating conditions are abnormal. In this case, the root cause of the failure is determined and adequate countermeasures are planned.

The anomalies/failures that can occur in the NFV environment are numerous and can be divided into three failure types [56] : process failure, network failure and throughput degradation.

- *The process failures* are caused by abnormal behavior of CPU and memory usage (eg. memory leak)
- *The network failures*such as network congestion are generally caused by traffic overload due to misconfiguration or unexpected traffic.
- *The throughput degradations* are due to surges in the number of TCP sessions processed by the OS on the virtualization environment and lead to packet loss and delay affecting the VNF application performance.

Traffic classification [57] is an efficient approach commonly used to detect intrusions and resolve network management problems. It consists in identifying and categorizing network data flows by examining relevant traffic features. Traffic classification is very important in NFV-based networks where a considerable amount of hidden traffic communicates among virtual machines [58].

Note that the anomaly detection and traffic classification in NFV are challenging for many reasons. The first is the softwarization of the network functions which exposes them to anomalies present in IT applications and others specific to virtualized infrastructures, related to resource sharing, scheduling and Virtual Machine (VM) live migration. Second, unlike traditional network functions, commonly provided by single vendor specific hardware, VNFs can share infrastructure resources from several vendors which makes anomaly detection and localization harder. And finally, due to the dynamic changes in the service and network configurations and the high availability performance required by VNF, traffic classification and failure detection should be performed on a real-time basis and at an early stage before the occurrence of a critical degradation.

5.2. Motivation for the Use of ML

As outlined above, anomaly detection in NFV is very difficult, especially with the expansion of the network scale and functionality in virtualized environments witch leads to an increase in the number and amount of managed resources and data. This calls for solutions able to deal with huge amount of data to scale with the NFV environment. ML has the potential to face these challenges as demonstrated in its successful applications in Big Data [16].

Due to the lack of information about the VNFs, either for privacy issues or simply because vendors can not provide it, anomaly detection systems should perform without requiring specific information about the VNFs. VNF's performance depends on many factors like the underlying NFV infrastructure, resource sizing and workload dynam-

ics, which makes it challenging for VNF vendors to provide complete capacity information [59]. ML models, proposed as a powerful tool to capture the unknown synergies between the VNF and its environment, are shown to be suitable for the anomaly detection problem in NFV.

5.3. Survey of Existing Approaches

The use of ML to classify Internet traffic and detect anomalies in Network is widely studied [60, 61]. This survey covers the research works that have adapted the ML techniques to suit the VNF environment requirements. A handful of research papers addressed this problem. They can be classified into supervised and unsupervised approaches as presented below.

5.3.1. Supervised Approaches

Supervised ML algorithms enable learning from historical observations to get predictions about the future. Applied to anomaly detection, this means to first observe the VNFs behavior both during normal and anomalous operations, and then to construct an estimation model, off-line using the observed data. The model is subsequently used to estimate the VNF system behavior at runtime (Table 2).

Table 2 Summary of existing Supervised based ML applications to anomaly detection in NFV

Work	Description	ML algorithm	Dataset
[62]	Detect and localize anomalies using a black box monitoring source	Random Forest [34]	Data collected after implementing the Clearwater framework [45]
[63]	Detect and localize SLA violations using a gray box monitoring source	Random Forest [34]	Data collected by implementing the Clearwater framework [45]
[64]	Evaluation of supervised ML algorithms for IP traffic classification in NFV	J48 [65], NaiveBayes [32], BayesNet [66]	Data collected through simulations
[67]	Design of a Traffic Classifier as a VNF automatically selecting the most suitable supervised or unsupervised ML technique	K-Nearest-Neighbors [68], SVM [31], Decision Tree [69], Adaboost [70], NaiveBayes [32], Multi-Layer Perceptron [71]	KDD dataset [72]
[73]	Anomaly detection for virtual network functions in service function chains (SFCs)	Sequential Deep Learning	Login Authentication Data (LAD) and Web Service Data(WSD)
[74]	Detect and localize SLA violations	Four algorithms including XGBoost and Deep Learning	Data collected from a private OpenStack Framework
[75]	Distributed anomaly detection for Virtualized Network Slicing	generative adversarial network (GAN) and Federated Learning (FL)	VNFDataset (virtual IP Multimedia IP system) [76]

Formally, using the same notation as in section 3, the set $\mathbf{X} = \{x_1, x_2, ...x_n\}$ of input features are the observation samples describing the VNFs system behavior. The desired output is $\mathbf{Y} \in \{normal, abnormal\}$ classifying the behavior to normal and abnormal. Y can be extended to $Y \in \{nrm, abnrm\_type\_1, abnrm\_type\_2, \ldots, abnrm\_type\_m\}$ to describe the types of the anomaly causing the abnormal behavior.

The authors of [62] propose a black-box anomaly detection and localization approach applied to VNF. They first define different fault campaigns and inject them into a VNF execution environment. The monitoring data periodically collected from the VNF virtual machine hypervisor, called black-box source, is used to train a Random Forest (RF) based model [34]. The model anomaly detection is then applied in each single VM to detect the anomalous behavior and localize the faulty one. The approach was validated on the IMS Clearwater project platform [45].

In [63], the same authors enhanced the previous work to detect and localize Service Lever Agreements (SLAs) violations. They use the RF based model and train it with richer data collected from the Operating System (OS) of the VNF virtual machines, a grey-box monitoring source. Experiments demonstrate that the grey-box monitoring source achieves better classification results compared to the black-box source based approach.

The authors of [64, 67] focused on traffic classification in NFV. In [64], the authors performed a benchmarking of the behavior of supervised ML algorithms in the IP traffic classification in NFV regarding their efficiency in terms of response time and precision. Their experiments, conducted on a simulation prototype reflecting a small VNF environment, demonstrated that the Naive Bayes [32] algorithm is a good traffic classifier. In [67], the authors demonstrated that the effectiveness of different ML based classification models depends on the protocol types and the features collected from network data. They designed VTC, a Virtual Traffic Classifier network function that automatically selects and applies the best ML learning classifier at run time. The proposal was evaluated using KDD flow traces [72], which do not reflect traffic in an NFV environment in our opinion, as it was constructed before the emergence of the NFV concept.

The authors of [73] argue that traditional anomaly detection methods, such as RF, gradient boosting machine,

and deep NN, are not well-suited for capturing the temporal dependencies and sequential patterns inherent in network data. As a result, these methods often fail to detect anomalies in complex NFV networks. To address this limitation, the authors proposed several sequential DL models that are specifically designed to learn time-series and sequential patterns of VNFs in SFCs (Service Function Chains). These models overcome the shortcomings of traditional methods by effectively capturing temporal dependencies and adapting to SFCs with varying lengths. The authors evaluated the proposed models on a real-world dataset of network traffic from an SDN and NFV environment. The results show that the proposed models significantly outperform traditional methods in terms of detection accuracy.

The authors of [74] proposed a ML-based anomaly detection framework for VNFs in SDN and NFV environments. The framework is designed to detect anomalies specifically related to SLA violations, which are critical for ensuring the quality of service (QoS) experienced by end-users. The framework utilizes datasets collected from VNF service function chain (SFC) scenarios implemented on an OpenStack environment. As ML algorithms, the authors evaluated four algorithms including Deep Learning and XGBoost.

In [75], the authors proposed to combine Generative Adversarial Network (GAN) and Federated Learning (FL) to effectively detect anomalies in virtual machines (VMs) in a virtualized network slicing environment. The algorithm utilizes a hierarchical cooperation mechanism among VM monitors, network slice managers, and the network controller to achieve global VM anomaly detection. After training generators and encoders on network slice managers, their updated parameters are sent to the network controller for aggregation using the FL framework. This collaboration, among VM monitors, network slice managers, and the network controller, allows to create a global VM anomaly detection model.

5.3.2. Unsupervised approaches

VNF applications can be exposed to unexpected failures resulting from software bugs and resource exhaustion. Therefore, the anomaly detection is expected to catch unknown behavior in a virtualization environment without experience and knowledge in the past. Unsupervised approaches can support anomaly detection without previously learning them. Unsupervised techniques are also commonly used for traffic classification (Table 3).

Table 3 Summary of existing Unsupervised based ML applications to anomaly detection in NFV

Work	Description	ML algorithm	Dataset
[77]	Detect anomalies by classifying monitoring data	Self Organizing Map [78]	Data collected through simulations
[56]	Enhanced universal version of the classifier proposed in [56]	Self Organizing Map [78]	Data collected through simulations
[79]	Train normal data patterns and detect any deviation	NoisyStudent [80]	ITU AI/ML in 5G challenge ([81] for more details)

The authors of [77] designed a scalable distributed fault detection framework for NFV called vNMF. vNMF uses a self-orgalizing map algorithm (SOM) [78] to analyze and classify the physical layer statistical data collected from the VMs. A network testbed simulating memory-leak and network congestion faults was used to evaluate the proposal. The results showed that SOM compared favorably with the k-means clustering algorithm. Unfortunately, the solution relies upon manually configuring the SOM clustering parameters and selecting the statistics for each failure type in advance, which results in a high maintenance load.

In [56], the same authors enhanced their previous work by proposing a more universal solution where a small set of local statistics and SOM clustering parameters can be used to detect different types of faults. To do so, they evaluated the SOM approach and determined the best set of clustering parameters that should be used to detect the faults. The effectiveness of the selected parameters was evaluated on a testbed simulating a virtualized residential gateways service.

Due to the scarcity of labeled faulty data, Unsupervised Learning (UL) methods have gained significant traction for detecting and localizing anomalies in NFV systems. In a UL approach, training is exclusively conducted on normal data to learn normal data patterns, and any deviation from the norm is considered an anomaly. However, it has been demonstrated that even small percentages of anomalous samples in the training data can substantially degrade the performance of UL methods. To address this issue, the aurhors of [79] proposed an anomaly detection approach based on the *NoisyStudent* technique, initially introduced to leverage unlabeled datasets in computer vision classification problems.

Conclusion: Research works applying ML to detect anomalies in NFV are very scare. As discussed in next section, this may be due to the lack of public datasets to evaluate such algorithms. Constructing one's own dataset requires implementing artificial fault injection techniques to emulate widespread faults existing in common computing systems, like the increase of resource consumption, misuse of memory, network packet loss, heavy workload, etc.

Moreover, observations related to a VNF VMs behavior should be collected periodically from different monitoring sources. Note that these sources can be classified as black-box or grey-box sources. Black-box sources, commonly hypervisor, do not need any tool to be installed to the VMs. Grey-box sources, by contrast, require the installation of monitoring agents in VMs. Grey-box sources generally provide richer information.

6. Discussion and Recommendation

Current ML applications to NFV encompasses scare but significant contributions. To encourage researchers to implement ML solutions for NFV, many important challenges need to be addressed and new ML opportunities should be explored. In the following, we discuss the most relevant ones.

6.1. Challenges

Lack of public datasets: The availability of standardized datasets, where researchers can test and validate their algorithms, have a huge influence on the evolution of ML techniques and applications. The need to publish datasets is very high, especially in networks where collecting a large amount of high quality labeled data is expensive and labor intensive, as discussed previously.

Although the publication of datasets is already a common practice in several popular ML applications, such as the PRID dataset [82] for person re-identification, public datasets in NFV, describing the NFV ecosystem components and synergies between them, are unfortunately scare or even nonexistent. As described in the survey presented above, most existing contributions using ML in NFV generated their own traces, either by conducting simulations, on after implementing a realistic NFV scenario like Clearwater [45].

The use of Clearwater is very popular in NFV. Clearwater is an open source implementation of an IMS (IP Multimedia Subsystem) for cloud platforms. It provides SIP-based (Session Initiation Protocol) voice and video calling, and messaging applications. It implements six VNFs, dynamically chained together, each one hosted on a VM, making it thereby well suited for NFV related studies. However, even if Clearwater is open source, implementing it and monitoring its components to generate data is labor intensive and requires software engineering skills and expensive hardware material, as most researchers used at least 3 high performance servers for their experiments [48, 62].

Regarding public datasets, there are few datasets proposed in the literature. As part of a scientific challenge, a dataset has been suggested to allow researchers to compare their algorithms in various applications related to the NFV domain [81].

Authors of [47] published a small dataset describing the evolution of CPU resource requirements of three NFVs as a function of the entering traffic. Unfortunately, the size of such data is not sufficient to evaluate many ML algorithms like deep learning where abundant data is required.

Recently, the authors of [83] published a dataset representing traffic passing through VNF chains and how it affects the network functions (NF) scaling. Again, due to the scarcity of public traffic for VNF chains, the traffic data was derived from empirical analyses and some assumptions. The authors randomly selected NF types composing the chains, then distributed web traffic traces among the chains. To study the auto scaling, the authors assumed that the NF resource requirement is proportionate to the traffic flowing through it. However, in real situations, the connections between traffic flow and VNF resource needs is much more complex as demonstrated in [46], and thus the dataset does not describe real synergism between the VNF and its environment.

Through this paper, we would like to attract the attention of the research community to the scarcity of public data describing the NFV environment and encourage them to publish such datasets where researchers can develop, test and compare ML based solutions. This is an essential step towards fully exploiting the potential of applying ML to NFV.

Evolution of ML techniques: The progress of ML is significantly driven by ML applications like computer vision, anomaly detection, etc. Similarly, applying ML to control networks requires adapting existing ML schemes and developing new ones. For example, modeling the systems as graphs [44], which is the most common representation of the network are required. Moreover, new dynamic and scalable learning techniques, more suitable to the new generations of networks (Cloud, 5G etc) are desired. In the following section, we present some of such emerging techniques.

6.2. Opportunities

We believe that the following learning models can be very useful to resolve research issues related to NFV and networks.

Deep learning [84] is one of the hottest research trends in ML. It is attractive for its capacity to extract high-level, complex hidden patterns from large amounts of input data. It often outperforms the state of the art relying on

hand-made features. Deep belief networks (DBNs) [85] and convolutional neural networks (CNNs) [86] are examples of deep learning approaches. To ensure the effectiveness of deep learning models, it is essential to invest in the creation of, diverse, and large datasets.

Distributed learning [87] is different from classical learning requiring the collection of data in a dataset for central processing. Even if a centralized and global view based network control can be achieved with SDN, learning in a distributed manner can be useful if a distributed control of the network is designed.

Online learning [88] is a learning technique proposed as an alternative to batch learning where the entire dataset should be available to create the model, which can no longer be modified. Conversely, online learning takes an initial prediction model and then uses available data streams to update and enhance the predictor accuracy. Thus, online learning is more suitable to dynamic networks where the prediction models should adapt with new situations.

Transfer learning [89] is the transfer of knowledge from a related task already learned to another task to improve learning in a particular domain, referred to as the *target domain*, where the data size is insufficient or the learning task is difficult. An example in NFV is the resource requirement prediction for a specific service chain, for which no sufficient data is available for learning the model from scratch. In such a case, other datasets collected from other similar services, with comparable consumption patterns, could be leveraged through transfer learning.

6.3. Exploring the Use of ML in Other Network Applications

In addition to the use of ML in the field of virtualization, ML is also utilized in other networking applications. In cloud computing, ML is employed for resource optimization [13], enabling dynamic allocation that enhances efficiency and reduces costs [14]. For performance analysis, ML can be used to continuously monitor the performance of cloud services, detect anomalies or service degradations [90], and trigger automatic alerts or corrective actions. Furthermore, with the increasing use of containers and microservices, ML aids in optimizing orchestration by predicting resource needs and dynamically adjusting allocation [91].

In the Internet of Things (IoT), ML is used to optimize the energy consumption of devices by learning and predicting their usage patterns [92]. Additionally, ML analyzes sensor data to predict failures before they occur, enabling proactive maintenance that can prevent costly downtimes [93]. ML also helps adjust network settings to maintain quality of service, especially in scenarios where networks are overloaded or undergoing dynamic changes [94].

In the realm of communication protocols, ML is used to analyze and optimize parameters of protocols such as WSN (Wireless Sensor Networks) [95]. In mobile network optimization, ML can optimize handover processes in cellular networks to ensure a smooth transition and maintain quality of service [96]. In mobile networks, ML helps analyze and optimize the use of frequencies to avoid interference and maximize coverage [97].

7. Conclusion

NFV promises operating expenditure savings and high flexibility in managing networks. However, some challenges still need to be overcome, namely for resource allocation, security and anomaly detection. In this paper, we discussed the potential of ML to overcome these problems. We included a survey of existing contributions and highlighted some opportunities and research directions. Our study reveals that the use of ML is a promising solution to control the NFV environment. However, some issues need to be addressed, essentially the lack of public datasets describing the VNF environment. In this context, the construction of public datasets become of utmost importance, as they are not only necessary for the evaluation and comparison of different ML techniques in the VNF context, but also for the developement of advanced deep learning models for VNF modeling. Another challenge is that the AI algorithm should be robust to uncertain environments [29, 98−102].

Author Contributions: **Houda Jmila:** Elaboration of the structure of the paper, Investigation, Methodology, Writing-original draft, Writing—review & editing; **Mohamed Ibn Khedher:** Elaboration of the structure of the paper, Investigation, Methodology, Writing-original draft, Writing—review & editing; **Mounim A. El Yacoubi:** Elaboration of the structure of the paper, Writing some parts, Reading and correction of the whole paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mijumbi, R.; Serrat, J.; Gorricho, J.L.; *et al*. Network function virtualization: State-of-the-art and research challenges. IEEE Commun. Surv. Tutorials, **2016**, *18*: 236−262. doi: 10.1109/COMST.2015.2477041
2. Sun, J.; Zhang, Y.; Liu, F.; *et al*. A survey on the placement of virtual network functions. J. Netw. Comput. Appl., **2022**, *202*: 103361. doi: 10.1016/j.jnca.2022.103361
3. Yi, B.; Wang, X.W.; Li, K.Q.; *et al*. A comprehensive survey of network function virtualization. Comput. Netw., **2018**, *133*: 212−262. doi: 10.1016/j.comnet.2018.01.021
4. Houda Jmila, Ines Houidi, and Djamal Zeghlache. Rsforevn: Node reallocation algorithm forvirtual networks adaptation. In 2014 IEEE Symposium on Computers and Communications (ISCC), pages 1−7. IEEE, 2014.
5. Jmila, H.; Houidi, I.; Zeghlache, D. Designing security-aware service requests for nfv-enablednetworks. *In 2019 28th International Conference on Computer Communication and Networks* (*ICCCN*), *Funchal*, *Portugal*, *23−26 June 2014*; IEEE: New York, NY, USA, 2014; pp. 1−7.
6. Piakaray, D.; Reddy D.S.; Goswami, S.; *et al*. A survey on the utilization of artificial intelligence and machine learning in the field of network functions virtualization and software defined networking. In *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 12–13 May 2023; IEEE: New York, 2023; pp. 442–445*. doi:10.1109/ICACITE57410.2023.10182596
7. Singh, S.; Jha, R.K. A survey on software defined networking: Architecture for next generation network. arXiv 2020, arXiv: 2001.10165. doi:10.48550/arXiv.2001.10165
8. Shubbar, R.; Alhisnawi, M.; Abdulhassan, A.; *et al*. A comprehensive survey on software-defined network controllers. In *International Conference on Next Generation of Internet of Things, Odisha, India, 5–6 February 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 199–231*. doi:10.1007/978-981-16-0666-3_18
9. Islam, M.S.; Al-Mukhtar, M.; Khan, M.R.K.; *et al*. A survey on SDN and SDCN traffic measurement: Existing approaches and research challenges. Engineering, **2023**, *4*: 1071−1115. doi: 10.3390/eng4020063
10. Soni, D.; Kumar, N. Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy. J. Netw. Comput. Appl., **2022**, *205*: 103419. doi: 10.1016/j.jnca.2022.103419
11. Kumar, Y.; Kaul, S.; Hu, Y.C. Machine learning for energy-resource allocation, workflow scheduling and live migration in cloud computing: State-of-the-art survey. Sustain. Comput. Inf. Syst., **2022**, *36*: 100780. doi: 10.1016/j.suscom.2022.100780
12. Ahmad, S.; Shakeel, I.; Mehfuz, S.; *et al*. Deep learning models for cloud, edge, fog, and IoT computing paradigms: Survey, recent advances, and future directions. Comput. Sci. Rev., **2023**, *49*: 100568. doi: 10.1016/j.cosrev.2023.100568
13. Khan, T.; Tian, W.H.; Zhou, G.Y.; *et al*. Machine learning (ML)-centric resource management in cloud computing: A review and future directions. J. Netw. Comput. Appl., **2022**, *204*: 103405. doi: 10.1016/j.jnca.2022.103405
14. Goodarzy, S.; Nazari, M.; Han, R.; *et al*. Resource management in cloud computing using machine learning: A survey. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 14–17 December 2020; IEEE: New York, 2020; pp. 811–816*. doi:10.1109/ICMLA51294.2020.00132
15. Faezi, S.; Shirmarz, A. A comprehensive survey on machine learning using in software defined networks (SDN). Hum.-Cent. Intell. Syst., **2023**, *3*: 312−343. doi: 10.1007/s44230-023-00025-3
16. Goswami, S.; Kumar, A. Survey of deep-learning techniques in big-data analytics. Wirel. Pers. Commun., **2022**, *126*: 1321−1343. doi: 10.1007/s11277-022-09793-w
17. Alromaihi, N.; Al-Omary, A.Y. Machine learning and big data based IDS system extensive survey. In *2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakheer, Bahrain, 20–21 November 2022; IEEE: New York, USA, 2022; pp. 7–12*. doi:10.1109/3ICT56508.2022.9990066
18. Messaoud, S.; Bradai, A.; Bukhari, S.H.R.; *et al*. A survey on machine learning in internet of things: Algorithms, strategies, and applications. Internet Things, **2020**, *12*: 100314. doi: 10.1016/j.iot.2020.100314
19. Arikumar, K.S.; Prathiba, S.B.; Moorthy, R.S.; *et al*. The role of machine learning in IoT: A survey. In *2022 3rd International Conference on Smart Electronics and Communication* (*ICOSEC*), *Trichy, India, 20–22 October 2022*; IEEE: New York, NY, USA, 2022; pp. 451–457. doi:10.1109/ICOSEC54921.2022.9952042
20. Yazici, İ.; Shayea, I.; Din, J. A survey of applications of artificial intelligence and machine learning in future mobile networks-enabled systems. Eng. Sci. Technol. Int. J., **2023**, *44*: 101455. doi: 10.1016/j.jestch.2023.101455
21. European Telecommunications Standards Institute. Available online: http://www.etsi.org/.
22. ETSI GS NFV 002 v1.1.1 (2013-10) Network Functions Virtualization (NFV); Architectural Framework. Available online: https://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf.
23. Khedher, M.I.; Jmila, H.; Yacoubi, M.A.E. Fusion of interest point/image based descriptors for efficient person re-identification. In *2018 International Joint Conference on Neural Networks (IJCNN 2018), Rio de Janeiro, Brazil, 8–13 July 2018; IEEE: New York, NY, USA, 2018; pp. 1–7*. doi:10.1109/IJCNN.2018.8489111
24. Qin, H.F.; El-Yacoubi, M.A. Finger-vein quality assessment based on deep features from grayscale and binary images. *Intern. J. Pattern Recognit. Artif. Intell*. **2019**, *33*, 1940022. doi: 10.1142/s0218001419400226
25. Yu, N.X.; Yang, R.; Huang, M.J. Deep common spatial pattern based motor imagery classification with improved objective function. Int. J. Netw. Dyn. Intell., **2022**, *1*: 73−84. doi: 10.53941/ijndi0101007
26. Li, X.; Li, M.L.; Yan, P.F.; *et al*. Deep learning attention mechanism in medical image analysis: Basics and beyonds. Int. J. Netw. Dyn. Intell., **2023**, *2*: 93−116. doi: 10.53941/ijndi0201006
27. Dao, Q.; El-Yacoubi, M.A.; Rigaud, A.S. Detection of Alzheimer disease on online handwriting using 1D convolutional neural network. IEEE Access, **2023**, *11*: 2148−2155. doi: 10.1109/access.2022.3232396
28. Jmila, H.; Khedher, M.I.; Blanc, G.; *et al*. Siamese network based feature learning for improved intrusion detection. In *26th International Conference on Neural Information Processing, Sydney, NSW, Australia, 12–15 December 2019*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 377–389. doi:10.1007/978-3-030-36708-4_31
29. Khedher, M.I.; Mziou-Sallami, M.; Hadji, M. Improving decision making-process for robot navigation under uncertainty. *In Ana Paula Rocha*, *Luc Steels*, *and H. Jaap van den Herik*, *editors*, *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*, *ICAART 2021*, *Volume 2*, *Online Streaming*, *February 4–6*, *2021*; SciTePress: Setúbal, Portugal, 2021; Volume 2, pp. 1105–1113. doi:10.5220/0010323311051113.

30. Khedher, M.I.; Mziou, M.S.; Hadji, M. Improving decision-making-process for robot navigation under uncertainty. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART 2021, Volume 2, 4–6 February 2021; SciTePress: Setúbal, Portugal, 2021; pp. 1105–1113*. doi:10.5220/0010323311051113
31. Hearst, M.A.; Dumais, S.T.; Osuna, E.; *et al*. Support vector machines. IEEE Intell. Syst. Their Appl., **1998**, *13*: 18−28. doi: 10.1109/5254.708428
32. Lewis, D.D. Naive (Bayes) at forty: The independence assumption in information retrieval. In *10th European Conference on Machine Learning, Chemnitz, Germany, 21–23 April 1998*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 4–15. doi:10.1007/BFb0026666
33. Cortes, C.; Vapnik, V. Support-vector networks. Mach. Learn., **1995**, *20*: 273−297. doi: 10.1007/BF00994018
34. Breiman, L. Random forests. Mach. Learn., **2001**, *45*: 5−32. doi: 10.1023/A:1010933404324
35. Shakya, A.K.; Pillai, G.; Chakrabarty, S. Reinforcement learning algorithms: A brief survey. Expert Syst. Appl., **2023**, *231*: 120495. doi: 10.1016/j.eswa.2023.120495
36. Guo, X.W.; Bi, Z.L.; Wang, J.C.; *et al*. Reinforcement learning for disassembly system optimization problems: A survey. Int. J. Netw. Dyn. Intell., **2023**, *2*: 1−14. doi: 10.53941/ijndi0201001
37. Gil Herrera, J.; Botero, J.F. Resource allocation in NFV: A comprehensive survey. IEEE Trans. Netw. Serv. Manage., **2016**, *13*: 518−532. doi: 10.1109/TNSM.2016.2598420
38. Schardong, F.; Nunes, I.; Schaeffer-Filho, A. NFV resource allocation: A systematic review and taxonomy of VNF forwarding graph embedding. Comput. Netw., **2021**, *185*: 107726. doi: 10.1016/j.comnet.2020.107726
39. Fang, J.Z.; Liu, W.B.; Chen, L.W.; *et al*. A survey of algorithms, applications and trends for particle swarm optimization. Int. J. Netw. Dyn. Intell., **2023**, *2*: 24−50. doi: 10.53941/ijndi0201002
40. Liu, Y.C.; Lu, H.; Li, X.; *et al*. Dynamic service function chain orchestration for NFV/MEC-enabled IoT networks: A deep reinforcement learning approach. IEEE Internet Things J., **2021**, *8*: 7450−7465. doi: 10.1109/JIOT.2020.3038793
41. He, N.; Yang, S.; Li, F.; *et al*. Leveraging deep reinforcement learning with attention mechanism for virtual network function placement and routing. IEEE Trans. Parallel Distrib. Syst., **2023**, *34*: 1186−1201. doi: 10.1109/TPDS.2023.3240404
42. Mijumbi, R.; Hasija, S.; Davy, S.; *et al*. A connectionist approach to dynamic resource management for virtualised network functions. In *12th International Conference on Network and Service Management* (*CNSM*), *Montreal, QC, Canada, 31 October–4 November 2016*; IEEE: New York, NY, USA, 2016; pp. 1–9. doi:10.1109/CNSM.2016.7818394
43. Mijumbi, R.; Hasija, S.; Davy, S.; *et al*. Topology-aware prediction of virtual network function resource requirements. IEEE Trans. Netw. Serv. Manage., **2017**, *14*: 106−120. doi: 10.1109/TNSM.2017.2666781
44. Scarselli, F.; Gori, M.; Tsoi, A.C.; *et al*. The graph neural network model. IEEE Trans. Neural Netw., **2009**, *20*: 61−80. doi: 10.1109/TNN.2008.2005605
45. Clearwater Project. Available online: http://www.projectclearwater.org/.
46. Jmila, H.; Khedher, M.I.; El Yacoubi, M.A. Estimating VNF resource requirements using machine learning techniques. In *24th International Conference on Neural Information Processing, Guangzhou, China, 14–18 November 2017;* Springer: Berlin/Heidelberg, Germany, 2017; pp. 883–892. doi:10.1007/978-3-319-70087-8_90
47. Mestres, A.; Rodriguez-Natal, A.; Carner, J.; *et al*. Knowledge-defined networking. ACM SIGCOMM Comput. Commun. Rev., **2017**, *47*: 2−10. doi: 10.1145/3138808.3138810
48. Cao, L.J.; Sharma, P.; Fahmy, S.; *et al*. ENVI: Elastic resource flexing for network function virtualization. In *9th USENIX Workshop on Hot Topics in Cloud Computing, HotCloud 2017, Santa Clara, CA, USA, 10–11 July 2017;* USENIX Association: Berkeley, CA, USA, 2017.
49. Zadeh, L.A. Fuzzy logic, neural networks, and soft computing. Commun. ACM, **1994**, *37*: 77−84. doi: 10.1145/175247.175255
50. Shi, R.Y.; Zhang, J.; Chu, W.J.; *et al*. MDP and machine learning-based cost-optimization of dynamic resource allocation for network function virtualization. In *2015 IEEE International Conference on Services Computing, New York, NY, USA, 27 June–2 July 2015*; IEEE: New York, NY, USA, 2015; pp. 65–73. doi:10.1109/SCC.2015.19
51. Neal, R.M. *Bayesian Learning for Neural Networks*; Springer: New York, NY, USA, 2012. doi:10.1007/978-1-4612-0745-0
52. Gupta, L.; Samaka, M.; Jain, R.; *et al*. COLAP: A predictive framework for service function chain placement in a multi-cloud environment. In *IEEE 7th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 9–11 January 2017*; IEEE: New York, 2017; pp. 1–9. doi:10.1109/CCWC.2017.7868377
53. Cao, L.J.; Sharma, P.; Fahmy, S.; *et al*. NFV-VITAL: A framework for characterizing the performance of virtual network functions. In *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), San Francisco, CA, USA, 18–21 November 2015*; IEEE: New York, 2015; pp. 93–99. doi:10.1109/NFV-SDN.2015.7387412
54. Gupta, V.; Dharmaraja, S.; Arunachalam, V. Stochastic modeling for delay analysis of a VoIP network. Ann. Oper. Res., **2015**, *233*: 171−180. doi: 10.1007/s10479-013-1472-7
55. Samariya, D.; Thakkar, A. A comprehensive survey of anomaly detection algorithms. Ann. Data Sci., **2021**, *10*: 829−850. doi: 10.1007/s40745-021-00362-9
56. Niwa, T.; Miyazawa, M.; Hayashi, M.; *et al*. Universal fault detection for NFV using SOM-based clustering. In *2015 17th Asia-Pacific Network Operations and Management Symposium* (*APNOMS*), *Busan, Korea* (*South*), *19–21 August 2015*; IEEE: New York, NY, USA, 2015; pp. 315–320. doi:10.1109/APNOMS.2015.7275446
57. Getman, A.I.; Ikonnikova, M.K. A survey of network traffic classification methods using machine learning. Program. Comput. Sof., **2022**, *48*: 413−423. doi: 10.1134/s0361768822070052
58. Ilievsk, G.; Latkoski, P. Network traffic classification in an NFV environment using supervised ml algorithms. J. Telecommun. Inf. Technol., **2021**, *3*: 23−31. doi: 10.26636/jtit.2021.153421
59. Troia, S.; Savi, M.; Nava, G.; *et al*. Performance characterization and profiling of chained CPU-bound virtual network functions. Comput. Netw., **2023**, *231*: 109815. doi: 10.1016/j.comnet.2023.109815
60. Chatterjee, A.; Ahmed, B.S. IoT anomaly detection methods and applications: A survey. Internet Things, **2022**, *19*: 100568. doi: 10.1016/j.iot.2022.100568
61. Azab, A.; Khasawneh, M.; Alrabaee, S.; *et al*. Network traffic classification: Techniques, datasets, and challenges. Digital Commun. Netw., **2024**, *10*: 676−692. doi: 10.1016/j.dcan.2022.09.009
62. Sauvanaud, C.; Lazri, K.; Kaaniche, M.; *et al*. Towards black-box anomaly detection in virtual network functions. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W), Toulouse, France, 28*

June–1 July 2016; IEEE: New York, NY, USA, 2016; pp. 254–257. doi:10.1109/DSN-W.2016.17
63. Sauvanaud, C.; Lazri, K.; Kaâniche, M.; *et al*. Anomaly detection and root cause localization in virtual network functions. In *2016 IEEE 27th International Symposium on Software Reliability Engineering* (*ISSRE*), *Ottawa, ON, Canada, 23–27 October 2016*; IEEE: New York, NY, USA, 2016; pp. 196–206. doi:10.1109/ISSRE.2016.32
64. Vergara-Reyes, J.; Martinez-Ordonez, M.C.; Ordonez, A.; *et al*. IP traffic classification in NFV: A benchmarking of supervised machine learning algorithms. In *2017 IEEE Colombian Conference on Communications and Computing (COLCOM), Cartagena, Colombia, 16–18 August 2017*; IEEE: New York, NY, USA, 2017; pp. 1–6. doi:10.1109/ColComCon.2017.8088199
65. Bhargava, N.; Sharma, G.; Bhargava, R.; *et al*. Decision tree analysis on J48 algorithm for data mining. *Proc. Int. J. Adv. Res. Comput. Sci. Sof. Eng*. **2013**, *3*, 2013.
66. Murphy, K.P. The Bayes net toolbox for matlab. Comput. Sci. Stat., **2001**, *33*: 1024−1034.
67. He, L.; Xu, C.; Luo, Y. vTC: Machine learning based traffic classification as a virtual network function. In *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, New Orleans, LA, USA, 11 March 2016;* ACM: New York, NY, USA, 2016; pp. 53–56. doi:10.1145/2876019.2876029
68. Peterson, L.E. K-nearest neighbor. Scholarpedia, **2009**, *4*: 1883. doi: 10.4249/scholarpedia.1883
69. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. IEEE Trans. Syst. Man Cybern., **1991**, *21*: 660−674. doi: 10.1109/21.97458
70. Margineantu, D.D.; Dietterich, T.G. Pruning adaptive boosting. In *Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, USA, 8–12 July 1997*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1997; pp. 211–218.
71. Pal, S.K.; Mitra, S. Multilayer perceptron, fuzzy sets, and classification. IEEE Trans. Neural Netw., **1992**, *3*: 683−697. doi: 10.1109/72.159058
72. Bay, S.D.; Kibler, D.; Pazzani, M.J.; *et al*. The UCI KDD archive of large data sets for data mining research and experimentation. ACM SIGKDD Explor. Newsl., **2000**, *2*: 81−85. doi: 10.1145/380995.381030
73. Lee, C.; Hong, J.B.M.; Heo, D.; *et al*. Sequential deep learning architectures for anomaly detection in virtual network function chains. In *2021 International Conference on Information and Communication Technology Convergence* (*ICTC*), *Jeju Island, Korea, Republic of, 20–22 October 2021*; IEEE: New York, NY, USA, 2021; pp. 1163–1168. doi:10.1109/ICTC52510.2021.9621043
74. Hong, J.; Park, S.; Yoo, J.H.; *et al*. A machine learning based SLA-aware VNF anomaly detection method in virtual networks. In *2020 International Conference on Information and Communication Technology Convergence* (*ICTC*), *Jeju, Korea* (*South*), *21–23 October 2020*; IEEE: New York, NY, USA, 2020; pp. 1051–1056. doi:10.1109/ICTC49870.2020.9289547
75. Wang, W.L.; Liang, C.C.; Tang, L.; *et al*. Federated multi-discriminator BiWGAN-GP based collaborative anomaly detection for virtualized network slicing. IEEE Trans. Mobile Comput., **2023**, *22*: 6445−6459. doi: 10.1109/TMC.2022.3200059
76. Yahia, I.B. VNFDataset: Virtual IP multimedia IP system. Available online: https://www.kaggle.com/imenbenyahia/clearwatervnf-virtual-ip-multimedia-ip-system.
77. Miyazawa, M.; Hayashi, M.; Stadler, R. vNMF: Distributed fault detection using clustering approach for network function virtualization. In *2015 IFIP/IEEE International Symposium on Integrated Network Management* (*IM*), *Ottawa, ON, Canada, 11–15 May 2015*; IEEE: New York, NY, USA, 2015; pp. 640–645. doi:10.1109/INM.2015.7140349
78. Kohonen, T. The self-organizing map. Neurocomputing, **1998**, *21*: 1−6. doi: 10.1016/S0925-2312(98)00030-7
79. Johari, S.S.; Shahriar, N.; Tornatore, M.; *et al*. Anomaly detection and localization in NFV systems: An unsupervised learning approach. In *2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2022*; IEEE: New York, NY, USA, 2022; pp. 1–9. doi:10.1109/NOMS54207.2022.9789938
80. Xie, Q.Z.; Luong, M.T.; Hovy, E.; *et al*. Self-training with noisy student improves ImageNet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), *Seattle, WA, USA, 13–19 June 2020*; IEEE: New York, NY, USA, 2020; pp. 10684–10695. doi:10.1109/CVPR42600.2020.01070
81. ITU-AI-ML-in-5G-challenge. Available online: https://github.com/ITU-AI-ML-in-5G-Challenge/.
82. Person Re-identification Datasets. Available online: http://robustsystems.coe.neu.edu/sites/robustsystems.coe.neu.edu\/files/systems/projectpages/reiddataset.html.
83. Rankothge, W.; Le, F.; Russo, A.; *et al*. Data modelling for the evaluation of virtualized network functions resource allocation algorithms. arXiv: 1702.00369, 2017. doi:10.48550/arXiv.1702.00369.
84. Deng, L.; Yu, D. Deep learning: Methods and applications. Found. Trends Signal Process., **2014**, *7*: 197−387. doi: 10.1561/2000000039
85. Lee, H.; Grosse, R.; Ranganath, R.; *et al*. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Canada, 14–18 June 2009*; ACM: New York, NY, USA, 2009; pp. 609–616. doi:10.1145/1553374.1553453
86. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012*; Curran Associates, Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
87. Catena, T.; Eramo, V.; Panella, M.; *et al*. Distributed LSTM-based cloud resource allocation in network function virtualization architectures. Comput. Netw., **2022**, *213*: 109111. doi: 10.1016/j.comnet.2022.109111
88. Chen, X.W.; Lin, X. Big data deep learning: Challenges and perspectives. IEEE Access, **2014**, *2*: 514−525. doi: 10.1109/ACCESS.2014.2325029
89. Azimi, Y.; Yousefi, S.; Kalbkhani, H.; *et al*. Applications of machine learning in resource management for RAN-slicing in 5G and beyond networks: A survey. IEEE Access, **2022**, *10*: 106581−106612. doi: 10.1109/access.2022.3210254
90. Chkirbene, Z.; Erbad, A.; Hamila, R.; *et al*. Machine learning based cloud computing anomalies detection. IEEE Netw., **2020**, *34*: 178−183. doi: 10.1109/MNET.011.2000097
91. Zheng, W.J. Efficient Resource Management for Deep Learning Applications with Virtual Containers. Master's Thesis, Fordham University, New York, NY, USA, 2020. doi:10.13140/RG.2.2.21705.77926
92. Javed, A.; Larijani, H.; Wixted, A. Improving energy consumption of a commercial building with IoT and machine learning. IT Prof., **2018**, *20*: 30−38. doi: 10.1109/MITP.2018.053891335
93. Teoh, Y.K.; Gill, S.S.; Parlikad, A.K. IoT and fog-computing-based predictive maintenance model for effective asset management

in industry 4.0 using machine learning. IEEE Internet Things J., **2023**, *10*: 2087−2094. doi: 10.1109/JIOT.2021.3050441
94. CheSuh, L.N.; Fernández-Diaz, R.Á.; Alija-Perez, J.M.; *et al*. Improve quality of service for the internet of things using blockchain & machine learning algorithms. Internet Things, **2024**, *26*: 101123. doi: 10.1016/j.iot.2024.101123
95. Karunanayake, P.N.; Könsgen, A.; Weerawardane, T.; *et al*. Q learning based adaptive protocol parameters for WSNs. J. Commun. Netw., **2023**, *25*: 76−87. doi: 10.23919/JCN.2022.000035
96. Farooq, M.U.B.; Manalastas, M.; Zaidi, S.M.A.; *et al*. Machine learning aided holistic handover optimization for emerging networks. In *ICC 2022 - IEEE International Conference on Communications, Seoul, Korea, Republic of, 16–20 May 2022*; IEEE: New York, NY, USA, 2022; pp. 710–715. doi:10.1109/ICC45855.2022.9839024
97. Sun, Y.H.; Peng, M.G.; Zhou, Y.C.; *et al*. Application of machine learning in wireless networks: Key techniques and open issues. IEEE Commun. Surv. Tutorials, **2019**, *21*: 3072−3108. doi: 10.1109/COMST.2019.2924243
98. Mohamed Ibn Khedher, Houda Jmila, and Mounim El Yacoubi. On the formal evaluation of the robustness of neural networks and its pivotal relevance for ai-based safety-critical domains. Int. J. Netw. Dyn. Intell., **2023**, 100018. doi: 10.53941/ijndi.2023.100018
99. Jmila, H.; Khedher, M.I. Adversarial machine learning for network intrusion detection: A comparative study. Comput. Netw, **2022**, *214*: 109073. doi: 10.1016/j.comnet.2022.109073
100. El Mellouki, O.; Khedher, M.I.; El-Yacoubi, M.A. Abstract layer for leakyReLU for neural network verification based on abstract interpretation. IEEE Access, **2023**, *11*: 33401−33413.
101. Khedher, M.I.; Ibn-Khedher, H.; Hadji, M. Dynamic and Scalable Deep Neural Network Verification Algorithm. *In Proceedings of the 13th International Conference on Agents and Artificial Intelligence* (*ICAART 2021*), *Online Streaming*, *4–6 February 2021*; pp. 1122–1130.
102. Ibn-Khedher, H.; Khedher, M.I.; Hadji, M. (2021, February). Mathematical Programming Approach for Adversarial Attack Modelling. *In Proceedings of the 13th International Conference on Agents and Artificial Intelligence* (*ICAART 2021*), *Online Streaming*, *4–6 February 2021*; pp. 343–350.

Citation: Jmila, H; Khedher, M; El-Yacoubi, M. The Promise of Applying Machine Learning Techniques to Network Function Virtualization. *International Journal of Network Dynamics and Intelligence*. 2024, 3(4), 100020. doi: 10.53941/ijndi.2024.100020

Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.