

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A survey on Machine Learning Techniques for Routing Optimization in SDN

RASHID AMIN¹, ELISA ROJAS², AQSA AQDUS¹, SADIA RAMZAN¹, DAVID CASILLAS-PEREZ³, and JOSE M. ARCO².

¹Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan. (e-mails: rashid4nw@gmail.com; aqsaqadus1997@gmail.com; sadiaramzan388@gmail.com)

²Universidad de Alcalá. Departamento de Automática, 28805, Alcalá de Henares, Spain (e-mails: elisa.rojas@uah.es; josem.arco@uah.es)

³Departamento de Teoría de la Señal y Comunicaciones, Universidad Rey Juan Carlos, 28943 Fuenlabrada, Spain (e-mail: david.casillas@urjc.es)

Corresponding author: Elisa Rojas (e-mail: elisa.rojas@uah.es).

This work was funded in part by grants from Comunidad de Madrid through project TAPIR-CM (S2018/TCS-4496) and project IRIS-CM (CM/JIN/2019-039), and by Junta de Comunidades de Castilla-La Mancha through project IRIS-JCCM (SBPLY/19/180501/000324).

ABSTRACT In conventional networks, there was a tight bond between the control plane and the data plane. The introduction of Software-Defined Networking (SDN) separated these planes, and provided additional features and tools to solve some of the problems of traditional network (i.e., latency, consistency, efficiency). SDN is a flexible networking paradigm that boosts network control, programmability and automation. It proffers many benefits in many areas, including routing. More specifically, for efficiently organizing, managing and optimizing routing in networks, some intelligence is required, and SDN offers the possibility to easily integrate it. To this purpose, many researchers implemented different machine learning (ML) techniques to enhance SDN routing applications. This article surveys the use of ML techniques for routing optimization in SDN based on three core categories (i.e. supervised learning, unsupervised learning, and reinforcement learning). The main contributions of this survey are threefold. Firstly, it presents detailed summary tables related to these studies and their comparison is also discussed, including a summary of the best works according to our analysis. Secondly, it summarizes the main findings, best works and missing aspects, and it includes a quick guideline to choose the best ML technique in this field (based on available resources and objectives). Finally, it provides specific future research directions divided into six sections to conclude the survey. Our conclusion is that there is a huge trend to use intelligence-based routing in programmable networks, particularly during the last three years, but a lot of effort is still required to achieve comprehensive comparisons and synergies of approaches, meaningful evaluations based on open datasets and topologies, and detailed practical implementations (following recent standards) that could be adopted by industry. In summary, future efforts should be focused on reproducible research rather than on new isolated ideas. Otherwise, most of these applications will be barely implemented in practice.

INDEX TERMS Software-Defined Networking, Machine-Learning, Routing, Optimization, Survey

I. INTRODUCTION

UNTIL few years ago, most company networks followed a traditional approach. In particular, legacy networking devices obeyed an architecture based on a tight bond between control and data planes [1], translated into a vendor lock-in, in which networks became complex and difficult to maintain and manage, particularly as they rapidly grew. When software is tightly bundled with hardware, interfaces are seller-specific. Manufacturers write the code, leading to long delays

in introducing the latest features and functions, i.e., networks are quite static and not flexible enough, which obstructs new business projects and applications. Software-Defined Networking (SDN) overcomes these issues by exchanging the control logic from devices to a central place (the SDN controller), in which networking decisions and overall functionality is developed based on common programming languages. Afterwards, the exchange of control logic is usually implemented by the OpenFlow protocol [2]. Fig. 1 illustrates

the architecture of SDN, in which the data plane (forwarding functions) and control plane (network control) are decoupled. This opens a new wide range of possibilities.

The SDN paradigm can be leveraged for multiple functions, such as traffic engineering, network virtualization, and load balancing, according to the network administrator needs [3]. It is helpful for new business projects and provides the facility of flexibility and virtualization. In particular, SDN has rapidly grown together with the Network Functions Virtualisation (NFV) [4] concept. They combined forces to boost emergent networking applications, including 5G, in which SDN serves as a network resource manager and reinforces network orchestration. Nevertheless, traditional routing algorithms are not good or suitable for SDN because their convergence and response are slow, and they follow a distributed approach, like the OSPF algorithm.

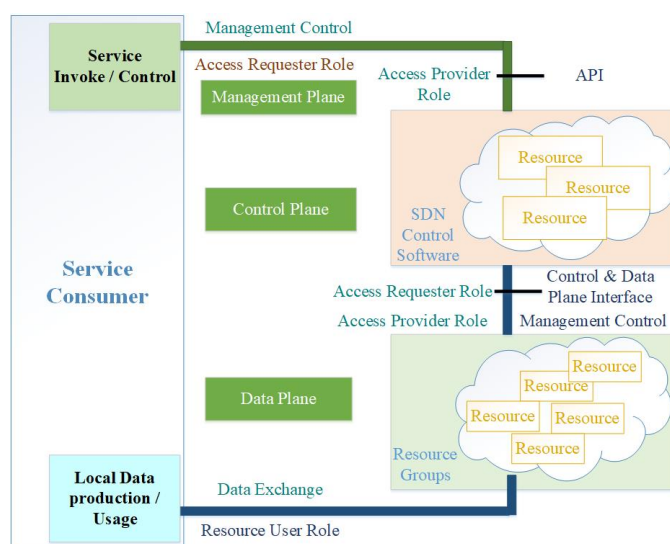


FIGURE 1. An overview of the SDN architecture

On the other hand, the concept of Artificial Intelligence (AI) was introduced by John McCarthy in 1956 [5]. In the field of computer science, AI is also known as Machine Intelligence. Machine Learning (ML) is a category of AI based upon the natural intelligence that can learn from data, make decisions, identify patterns and perform different actions with less human intervention. The devices based on ML perceive the real environment and apply actions according to their needs or requirements to maximize the opportunity to achieve their goal successfully. ML can potentially be used to solve many problems in networking, including design, implementation, performance and verification.

Nowadays the use of ML techniques is increasing. It is considered that these techniques are better as compared to traditional algorithms, particularly for the processing and analysis of large volumes of data. In the area of networking, researchers are paying their attention to the usage of these techniques. For example, the *Knowledge plane* concept was first coined in 2003 by Clark *et al.* [6] and introduced the

primitive view of ML techniques in networking. Different ML techniques are employed in SDN to achieve synergistic effects and to overcome individual limitations.

Additionally, in the specific field of SDN, ML has been leveraged in different applications, including traffic engineering [7], [8], resource management [9], [10], intrusion detection systems [11], [12] and for other security purposes [13], [14]. For instance, Mijumbi *et al.* [15] leverage it for adjusted virtual network and managed resources in virtualized network by using control plane, or Akyildiz *et al.* [16], which introduce the state of art for traffic-engineering in SDN/OpenFlow networks.

As a consequence, in SDN, the role of ML has recently boosted due to its multiple applications. The architectural logic of SDN harmonizes better with ML algorithms than with traditional algorithms. In particular, many research results combine ML techniques with SDN for routing optimization. Furthermore, ML is seen as key technology trend for 6G and beyond [17].

A. CONTRIBUTIONS OF THE SURVEY

In this paper, we survey different approaches of ML techniques for routing in SDN. We try to cover most of the ML techniques and classify them into three primary categories. The main objective is to provide a comprehensive overview of ML techniques in SDN for routing optimization, emphasizing on contributions and learned lessons for future research.

The main contribution of this survey is that it strictly focuses on ML techniques applied for routing in SDN. While other surveys have a more generalist approach (focusing either on SDN or ML, different networking applications, and providing an overall idea), our survey aims to delve into specific routing applications and why ML has become such an important actor thanks to SDN (i.e., centralizing the logic and facilitating the integration of ML, otherwise unfeasible in traditional routing approaches, mostly distributed).

In summary, this survey encompasses the following contributions:

- It provides an in-depth overview of SDN, routing, and ML techniques, performed by a group of researchers coming from different fields and expertise in different areas, which enriches the analysis.
- It presents a qualitative analysis of ML techniques to help new researchers in the field where to start from, as a guideline, based on the context of the scenario to be analyzed and the desired applications.
- It classifies the most recent works in relation with the survey according to three main categories of ML. Most works were published during the last three years.
- It analyzes and compares all works, including the techniques leveraged, their specific objective (considering all of them are focused on routing), their implementation and evaluation, pros and cons. This analysis is concluded by a summary of learned lessons and research trends.

- It provides a comprehensive section including future research directions, which, from our point of view, represents the most interesting part of the survey, as much work still needs to be done in the field to be relevant in a long-term manner.

B. METHODOLOGY OF THE SURVEY

The search of the state of the art was mainly performed using the Google Scholar site, which comprehensively indexes works (articles, patents, etc.) from different journals and sites, and even from archive repositories. During our search, they main keywords used were: *routing*, *SDN* and *ML* (these two latter both using acronyms and the full name), which are the three core terms in relation with the survey, but we also looked for *AI*, *optimization*, *traffic engineering*, *load balancing*, *NFV*, *learning*, *supervised*, *unsupervised* and *reinforcement* (which are directly related with the classification of ML techniques, explained within the following sections), among others. Additionally, we also used *survey*, *overview* and *tutorial* to examine the closest related works, and to evaluate the contributions of our survey.

The search yielded thousands of results, most of them published within the last five years, from which we filtered the ones directly related with our analysis. The growth of publications was particularly relevant within the last two years with an exponential increase for the reinforcement learning-based approaches. For this reason, we applied filters based on number of citations to analyze the most cited ones first, and we focused on articles written in English (which was the most common language) and published in prestigious journals (preferably indexed in JCR).

Finally, we also scrutinized the references of articles already selected for the survey to look for additional relevant works.

C. STRUCTURE OF THE SURVEY

The roadmap of this manuscript is depicted in Fig. 2. The article starts with a extensive analysis of the related work in Section II and core definitions of SDN in Section III. Afterwards, a general description of ML techniques, together with a qualitative comparison, is presented in Section IV, which is divided into three categories i.e. Supervised Learning (SL), Unsupervised Learning (UL), and Reinforcement Learning (RL) (which includes Deep Reinforcement Learning (DRL)). Section V is devoted to the application of these ML techniques together with SDN for routing optimization. This section is finalized by a quick overview that presents learned lessons, current trends and the best published works so far, according to our analysis. Section VI discusses specific future research directions and open issues of routing optimizations in SDN, followed by the overall conclusions in Section VII. Finally, Table 1 alphabetically lists the acronyms used throughout the paper.

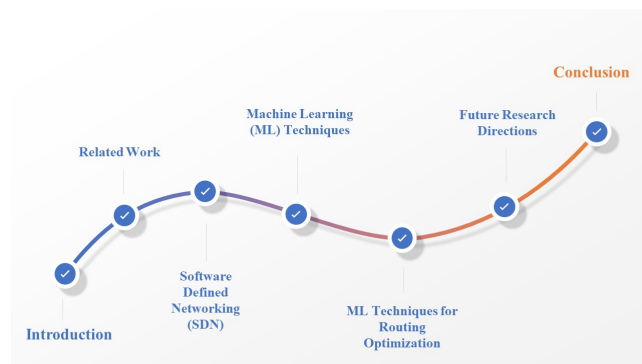


FIGURE 2. Summarized structure of the survey

II. RELATED WORK

To provide a context of the contributions of this survey, the first step is to review some surveys related with the methods and techniques of ML applied to routing SDN, which are summarized in Table 2. This summary presents the authors, the focus of the survey, as well as the coverage of the three areas that characterize our survey: SDN, routing and ML. In particular, an empty cell means that area is not covered, while one or two ticks indicate the topic is partially and fully covered, respectively. Additionally, pros (highlights) and cons (missing aspects in relation with the contributions of our survey) are also included as two separate columns. It is important to note that the selection of works was based on relevance to our survey (at least two of the three ideas covered in our survey should be included) and/or number of citations. Otherwise, if not filtered, there are hundreds of surveys somehow related to ours (either because of SDN, routing or ML), like surveys about SDN controller placement [18] or ML applied to network security [19].

The first two surveys in the list are strictly focused on the SDN paradigm. Although they only focus on one aspect of the three covered in the survey, they are worth mentioning due to its high amount of citations (>1000). Nunes et al. [20] present the state-of-art in programmable networks, with a particular focus on SDN. These networks are depicted from the oldest to the newest development ideas, followed by the architecture of SDN and the standard of OpenFlow. Diverse alternatives are also discussed for the implementation and testing of SDN-based services and protocols. Finally, they provide information about current and future SDN-based application trends, as well as multiple research directions of SDN. Hu et al. [21] survey the implementation of SDN/OpenFlow, including basic concepts, language abstraction, applications, virtualization, controller, security, Quality of Service (QoS), as well as integration with optical and wireless networks. They also compare the merits and demerits of different network implementation schemes. This survey is particularly helpful to understand the progress of SDN/OpenFlow designs.

Afterwards, we would like to highlight two surveys that still mainly focus on SDN, but including some sections to

TABLE 1. Acronym list

Acronym	Full name
AI	Artificial Intelligence
ANN	Artificial Neural Network
BGP	Border Gateway Protocol
CRAM	Cognitive Routing Algorithm Module
CNN	Convolutional Neural Network
CRE	Cognitive Routing Engine
DDPG	Deep Deterministic Policy Gradient
DNN	Deep Neural Network
DoS	Denial of Service
DRL	Deep Reinforcement Learning
FFNN	Feed Forward Neural Networks
GA	Genetic Algorithms
GAN	Generative Adversarial Networks
GLM	Generalized Linear Model
GNN	Graph Neural Network
GRU	Gate Recurrent Unit
HIA	Hybrid Intelligent Approach
HTTP	Hyper Text Transfer Protocol
ICT	Information and Communication Technology
KDN	Knowledge-Defined Networking
KPI	Key Performance Indicator
LSTM	Long Short-Term Memory
MDS	Markov Decision Support
ML	Machine Learning
MLRC	Machine Learning Routing Computation
MLP	Multilayer Perceptron
NBI	North Bound Interface
ONF	Open Networking Foundation
ONOS	Open Network Operating System
OSPF	Open Shortest Path First
PSO	Particle Swarm Optimization
QoE	Quality of Experience
QoS	Quality of Service
SDN	Software Defined Networking
RIP	Routing Information Protocol
RL	Reinforcement Learning
RMON	Remote Network Monitoring
RndNN	Random Neural Network
RNN	Recurrent Neural Network
SARSA	State-Action-Reward-State-Action
SBI	South Bound Interface
SL	Supervised Learning
SNMP	Simple Network Management Protocol
SOM	Self-Organizing Maps
SSH	Secure Shell
SVM	Support Vector Machine
TE	Traffic Engineering
TIDE	Time-relevant DEep reinforcement learning
TM	Traffic Matrix
UL	Unsupervised Learning
VRRP	Virtual Router Redundancy Protocol
XML	Extensible Markup Language

analyze the specificities of routing in this field. **Kreutz *et al.*** [22] is one of the most referenced surveys in the SDN field. It discusses the definition of SDN, its core concepts and differences compared to traditional networks. The architecture of SDN is presented in a bottom-up approach. The authors performed a comprehensive analysis of its architecture, APIs, network programming and network layers. They also focused on the major problems of cross layering

and their solutions. Keeping in view the security, performance, scalability and resilience, the design of controllers and switches are addressed in this study as well. **Mendiola *et al.*** [23] extensively survey approaches for traffic engineering in SDN, indirectly mentioning their application in routing in SDN.

Additionally, with a bigger emphasis on routing and smaller on SDN, **Karakus *et al.*** [24] provide a comprehensive survey and summary of taxonomy and characterization of SDN control plane scalability. Two main areas are discussed: network topologies and mechanism to tackle scalability. In the former, they describe the relationship of the topology with scalability, considering the impact of a centralized/distributed controller and, transversally, hybrid and hierarchical designs. In the later, they studied mechanisms to optimize controller scalability, such as control plane routing and parallelism based optimization. It finalizes summarizing challenges and open problems for scalable SDN control planes. On the other hand, just focusing on ML and routing, without emphasis on SDN, **Chen *et al.*** [25] provide a very good overview on the application of Artificial Neural Networks (ANNs) on wireless networks applications.

The first survey works to address the three features examined in this survey (SDN, routing and ML) are more recent (from the last three years). **Binsahq *et al.*** [26] focus on autonomic provisioning and management of QoS in SDN. As part of that analysis, it encompasses some works related with ML and routing, and the authors specifically have a section devoted to ML for QoS management. **Etengu *et al.*** [27] extensively analyze AI-assisted networks for green routing and load balancing, focused on a pragmatical approach, that is, hybrid SDN, usually leverage for smooth migration from legacy systems. At the end of the survey, the authors provide a set of challenges and future research directions, and they define a specific framework to tackle them. **Qian *et al.*** [28] concisely survey a set of applications in communication networks where reinforcement learning is applied, including network caching or task offloading. It includes very briefly the relationship with SDN and routing applications. **Mammeri *et al.*** [29] comprehensively analyze reinforcement learning approaches for routing, not only for SDN-based networks, but for all types of networks, which provides a very good overview of the evolution of this specific ML technique and its application in communication networks. **Jamshidi *et al.*** [30] explain applications based on ML methods and techniques by dividing them into six categories of networking, namely: traffic prediction, network security, cloud services, application identification, domain name system, and QoS. In all these categories, they determine the ML methods and input datasets. It summarizes the various challenges and major findings of these input data and ML methods. **Zhang *et al.*** [31] presents diverse applications of ML in routing and resource allocation in optical networks, without any specific focus on SDN-enabled networks.

Four works are close to the objectives of our survey. **Boutaba *et al.*** [32] survey ML research opportunities and

TABLE 2. Comparison summary of the related work
-: Not covered; ✓: Partially covered; ✓✓: Fully covered

Ref.	Focus	SDN	Routing	ML	Highlights	Missing aspects in comparison to this survey
Nunes et al. [20]	State of the art of programmable networks	✓✓	-	-	Provides complete history of programmable networks regarding SDN and OpenFlow standard	Lack of diversity in terms of SDN controllers
Hu et al. [21]	SDN/OpenFlow in different applications and network types	✓✓	-	-	Covers important design issues of SDN	Missing in-depth investigation of design issues
Kreutz et al. [22]	Comprehensive survey about SDN core concepts	✓✓	✓	-	Detailed understanding of SDN and its difference from traditional networks	Application of ML together with SDN is not covered
Mendiola et al. [23]	Comprehensive survey about traffic engineering in SDN	✓✓	✓	-	Provides a complete view of SDN contributions to traffic engineering solutions	Influences of only three types of interfaces are discussed
Karakus et al. [24]	Scalability problems of controllers in SDN	✓	✓✓	-	Study of problems and issues in SDN control plane scalability	Limited in terms of approaches, lack of a thorough discussion on challenges and problems for more scalable control planes
Chen et al. [25]	ANN-based ML for wireless networks	-	✓	✓	Good overview of the context of ANNs, as well as the diverse applications in wireless networks (IoT, UAVs, etc.)	Only focuses on a very specific type of ML technique and on one type of network. No particular focus on SDN is provided
Binsahaq et al. [26]	Autonomic Provisioning and Management of QoS in SDN	✓✓	✓	✓	Comprehensive analysis of autonomic management focused on QoS in SDN	Lacks focus on routing and analysis of ML techniques, which are barely addressed
Etengu et al. [27]	AI-Assisted Green-Routing and Load Balancing in Hybrid SDN	✓✓	✓	✓	Comprehensive research insights about AI-assisted green networking, included an architectural design	Lacks focus on routing proposals and there is no research challenges section
Qian et al. [28]	Concise survey of reinforcement learning applications in communication networks	✓	✓	✓	Discusses the applications of AI (RL/DRL) in different communication networks	Lack of diversity in terms of application details
Mammeri et al. [29]	Reinforcement learning approaches for routing (SDN and non-SDN)	✓	✓✓	✓	Provides comprehensive review of RL-based routing protocols	Strictly focused on RL and not other ML methods
Jamshidi et al. [30]	General analysis of ML techniques used in different applications of network system	✓	✓	✓✓	Provides understanding of ML techniques for addressing multiple networking challenges and summarize key findings	Implementation details are not discussed
Zhang et al. [31]	ML-assisted routing in optical networks	✓	✓	✓✓	Comprehensive overview of the application of ML in routing and resource allocation in optical networks	Only focuses on optical networks (and not necessarily SDN-based), the classification of ML techniques is not comprehensive
Boutaba et al. [32]	Applications of ML in different areas of networking	✓✓	✓	✓✓	Extensive knowledge of ML across different networking technologies	It does not provide in-depth fundamental aspects of SDN
Xie et al. [33]	Implementation of ML techniques in SDN, in different terms	✓✓	✓	✓✓	Provides overall understanding of ML algorithms and its working in the domain of SDN	General overview, with only around ten works related to routing and no research challenges section
Zhao et al. [34]	Networking applications based on the combination of SDN and ML	✓✓	✓	✓✓	Provides simple guide for ML applications and their challenges in SDN	Lacks focus on routing and the amount of works is not so comprehensive
Quach et al. [35]	Specific analysis of Reinforcement Learning for efficient routing in SDN	✓✓	✓✓	✓	Discusses RL-based routing in SDN	Strictly focused on RL and not other ML methods
Amin et al.	Analysis of ML techniques for routing optimization in SDN	✓✓	✓✓	✓✓	N/A	N/A

evolution in the field of networking. They provide a brief introduction to ML techniques, engineering techniques, approaches and methods for data gathering in network traffic, followed by an overview of ML techniques in routing, traffic classification, QoS/QoE, anomaly detection, fault management, and intrusion detection. Additionally, they focus on the importance of secure learning support, online learning and the architectural design of systems so that ML can be used easily. Their survey covers above 500 studies. **Xie et al.** [33] present a comprehensive detail of the ML techniques, architecture and working of SDN. Different types of ML algorithms are explained and described in SDN in terms of optimization, QoE/QoS, security, resource management, and traffic classification. Future research and challenges are also discussed. **Zhao et al.** [34] surveys the diverse networking applications that benefit from the combination of SDN and ML, including a section about routing optimization, though not in depth. **Quach et al.** [35] is the closest to our work so far, but it just focuses on approaches based on reinforcement learning. In any case, it is a concise survey about that type of routing in SDN and provides a quick overview about objectives and associated algorithms.

Finally, Farhady et al. [36], Scott-Hayward et al. [37], Al-Heety et al. [38], Hatagundi et al. [39], Chica et al. [40] reviewed different SDN related technologies, the details of SDN planes, benefits, challenges, security, and attacks in SDN but their scope is further from the analysis of this survey, as they do not discuss the applications or use of ML in SDN.

Currently, to the best of our knowledge, no one specifically surveyed the ML techniques for routing optimization in SDN. To fill this gap, in this paper, we provide a detailed study of ML types and their usage in SDN routing. We envision that our discussion and exploration will provide readers with an overall understanding of ML techniques for routing in SDN and foster more subsequent studies on this issue.

III. SOFTWARE-DEFINED NETWORKING (SDN)

Over the last decade, a new wave of innovation has emerged in the networking field thanks to the SDN paradigm [22]. In its origins, it consisted mainly of a protocol, OpenFlow [41], which separated the data and control planes, allowing the flourishing of new network protocols and designs. However, it rapidly evolved into a new architectural approach in which the so-called *dummy* switches (data plane) were managed by a logically centralized entity, the SDN controller (control plane), through the OpenFlow protocol. Although the concept of uncoupling these two planes was not new in the field. SDN unlocked the hardware market, very opaque until that moment, bringing the opportunity for new manufacturers and researchers to cooperate, even in hybrid environments [42]. Currently, the Open Networking Foundation (ONF) is in charge on the main standardization efforts in the field of SDN.

By definition, SDN hides the complexity of the network design. Its architecture (previously depicted in Fig. 1) pro-

vides dynamic, cost-effective, manageable and adaptable network control. An alternative definition of the SDN architecture is illustrated in Fig. 3, in which SDN consists of four planes [43].

At the bottom of the architecture, the **Data Plane** is also known as the forwarding plane, user plane or carrier plane [44]. It consists of the set of network devices (virtual or physical) that transmits the user traffic. The Data Plane handles arriving frames according to the logic of the Control Plane. Some of the actions to be applied include forwarding the frame, modifying it or discarding it.

The **Control Plane** is the network *brain*, responsible of decisions such as routing or traffic signaling [44]. Though originally designed completely separated from the Data Plane, some part of the Control Plane might be delegated to network devices under some circumstances, following a hybrid approach [42]. The communication of these two planes is performed through the Southbound Interface (SBI), originally following the OpenFlow protocol, but currently involves other alternatives such as P4Runtime [45].

Above it, the **Application Plane** is connected through the Northbound Interface (NBI), usually asynchronously (e.g., REST API), to define the overall behavior of the network desired by the network administrator. Some authors merge Application and Control planes, some other do not. The criterion to separate them is that usually the Control Plane consists of core networking functions, common for all types of applications (for instance, topology discovery, shortest-path computation, etc.), while the Application Plane are individual applications that leverage the Control Plane to be executed. The so-called SDN controllers are software platforms that include both Control and Application planes.

Finally, the role of the **Management Plane**, transversal to the three previous planes, is to provide a mean to manage the network for additional aspects such as configuration, monitoring, billing, etc. Some common protocols include classic ones like: HTTP (Hyper Text Transfer Protocol), SNMP (Simple Network Management Protocol), XML (Extensible Markup Language), RMON (Remote Network Monitoring), and SSH (Secure Shell). This plane is clearly the most heterogeneous of the architecture and encompasses diverse challenges [46]. In some specifications, particularly the latest ones, the Management Plane is seen as part of the Control Plane, as a *management-control continuum*.

In summary, the main benefit of the SDN paradigm is that it brings new possibilities for logically centralized network control. For instance, it allows users to access virtual and physical elements from a single location, because of its virtualized control planes and forwarding planes. SDN also allows administrators to monitor everything centrally, which enhances global view management compared to traditional networks. Some major telecom organizations (e.g., Google [47], VMware [48], Microsoft [49], or Facebook [50]) have already adopted the SDN architecture for their data centers. At the same time, some popular network vendors and related companies (namely Cisco [51], Huawei [52], NEC [53], Veri-

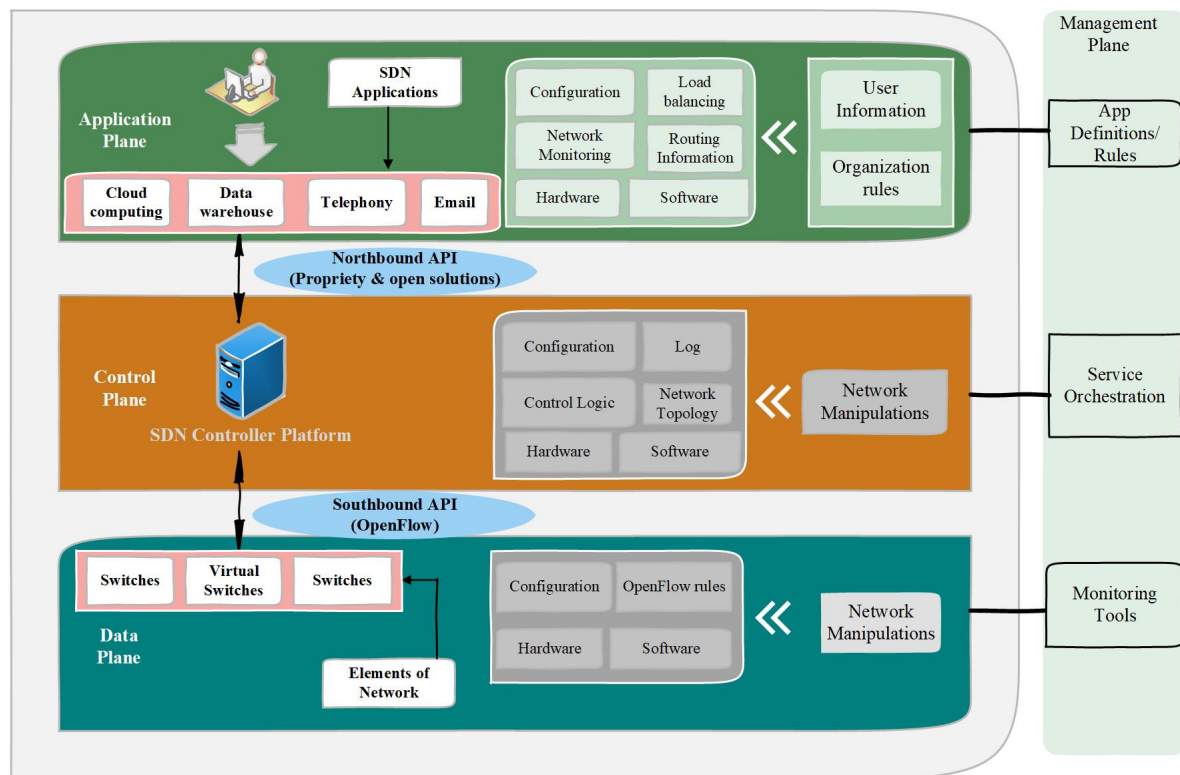


FIGURE 3. Architectural planes of SDN, functions and relationships

son [54], HP [55], and AT&T [56]) are also firmly committed to the application of the SDN architecture by designing and producing SDN-related components. As a consequence, centralized techniques like ML are increasing in SDN, reinforced by its architecture, including applications such as resource management, QoS prediction, traffic engineering, security and routing optimization.

A. ROUTING APPLICATIONS AND CHALLENGES IN SDN

Optimized routing could be considered one of the core objectives in computer networks. In particular, this objective is directly related to network traffic engineering, as this field is founded on one particular idea: to accomplish that traffic is routed according to the exact traffic demands [23]. Therefore, we could claim that traffic engineering is one type of the multiple optimizations of routing, as routing could also be optimized based on other parameters (and not only on traffic demands). Additionally, these traffic demands are variable depending on whether we consider data or control traffic. In this regard, the logically centralized view of the SDN controller facilitates many aspects in comparison to traditional routing. For instance, topology graphs can be easily extracted from the network and shortest-path algorithms, like Dijkstra, can be efficiently –and dynamically– computed to obtain the best paths. This had led to the direct application of computer science algorithms to computer networks [57], without the need of translating them into distributed protocols, like the

generation of disjoint paths for traffic engineering purposes, which is now easier than ever [58]. Consequently, thanks to SDN, routing can be easily parameterized based on types of optimal routing (shortest path, constrained shortest path, etc.), cost functions or resources, for example. This facilitates and easy adaptation and deployment based on the specific scenario [57], as there is not a clear winning type of routing applicable to all networks.

It is also important to highlight that the data and control plane decoupling of SDN implies the incorporation of a new communication channel in the southbound of the architecture, typically implemented with OpenFlow. This channel can be implemented either in an out-of-band or in an in-band mode. In the former, the communication between both planes is direct (though it requires more resources for deployment), while in the former it is not. That is, in-band SDN also requires the application of traffic engineering for optimized routing.

Another example is the opportunity to implement newer functionality, particularly the one related with cloud computing, like ML. In this regard, SDN simplifies the development of ML techniques to support network routing thanks to its centralized monitoring capabilities.

Nevertheless, although SDN is an ideal answer for Information and Communication Technology (ICT) deployments, cloud suppliers and undertakings, SDN faces a few challenges [59] that affect its performance and usage. The set of SDN challenges comprises:

- **Controller location:** SDN implies an additional communication channel between the data and control plane, which might not be completely transparent, particularly in large networks, in which out-of-band communication might be unfeasible. Therefore, the specific location of the controller should be carefully planned.
- **Scalability:** Directly related with the previous aspect, as SDN is logically centralized, network managers should consider to what extent should all control be delegated to the controller, to avoid bottlenecks and scalability issues. However, this decision is not trivial for all use cases.
- **Performance optimization:** Performance optimization is a challenge in all network types per se, but in SDN the way to achieve it changes from a distributed approach to a centralized one.
- **Security:** As SDN is logically centralized, it might be easily threatened.
- **Interoperability:** Particularly relevant in large networks, heterogeneity and interoperability among different types of SDN technologies is still a challenge.
- **Reliability:** Similarly to traditional networks, reliability is also a challenge. However, in SDN is even worse, as the control channel communication represents a new potential failure point that should be reliable and, hence, protected.

One of the consequences is that SDN controllers must be astutely arranged to forestall manual blunders. For example, in a conventional system when one or many system gadgets fall flat, management information errors might be locally kept and do not affect the overall behavior of the network. Whereas in SDN, a solitary controller is accountable for all the systems, and if there is any inaccuracy in it, the entire system might fall. To address this issue, research should be focused on coordination of distributed SDN controllers with security guarantees. Currently, from all existing SDN controllers [60], we would like to highlight two of them: Ryu [61], because of simplicity and easy prototyping, and ONOS [62], as it is supported by the ONF and implements the driving SDN use cases devised by industry.

In summary, the centralized architecture of SDN provides a faster overview of the network status and substantially smoother programmability and updates, but it still requires a control overhead that needs to be carefully managed and that is established now in a north-south (hierarchical) style rather than east-west (flat) manner, typical of distributed legacy systems.

B. ML IN SDN ENVIRONMENTS

Although ML (as well as AI, generally speaking) has been applied in networking for two decades now, its adoption in practical deployments is still in early stages [63]. Thanks to the softwarization of networks, the application of AI and ML in networking is nowadays potentially easier to implement, thus, opening a wide range of new functionalities. In fact, some authors have recently addressed the term Knowledge-

Defined Networking (KDN) [64], which include the so-called **Knowledge Plane** [6], directly related with the inclusion and integration of Artificial Intelligence in SDN environments.

In particular, data-driven networks [65] are one type of computer networks, fostered by both SDN and NFV, which could easily adapt to traffic demands (once again for traffic engineering purposes) or network changes, for example. Although some authors agree that there is still work to be done (in particular regarding models and architectural aspects [65]), it seems we have now reach the right momentum to even accomplish the concept of self-driven networks [66]. For example, a self-driven network benchmarking framework was recently proposed by Zerwas *et al.* [67] and they prove how it can be applied to a well-know SDN software switch, viz. Open vSwitch (OVS).

Finally, we would like to put some additional emphasis in the case of the future 6G networks, as many authors already agree that ML is a key enabler [68], [69]. Some applications included in their roadmap are, for instance, object localization, Unmanned Aerial Vehicle (UAV) communication, surveillance, security and privacy preservation [69]. All of them envisioned as part of fog/edge computing architectures [70].

However, although the SDN architecture allows a very straightforward application of intelligent algorithms, there is still a need to analyze which suits best each type of network and data, as the requirements greatly vary among different network scenarios. Furthermore, open networking datasets are still a scarce resource for the research community, and these are key components to design ML-based frameworks.

IV. MACHINE LEARNING TECHNIQUES

ML was first introduced by Arthur Samuel in 1959. ML is the branch of AI that enables the systems to learn automatically from experience and to improve themselves without being explicitly programmed [71]. It guides systems for making good predictions based on data. ML systems can make decisions and identify different patterns. ML models get the new data independently and make decisions, computations and results by learning from previous state of computation. It provides solution in many problems, such as pattern recognition [72], character recognition [73], speech recognition [74], vision, or robotics.

ML is a very vast field whose methods have been classified attending to multiple categories. A general classification groups ML techniques according to the kind of learning involved, distinguishing the *supervised*, *unsupervised* and *reinforcement learning* (with a particular focus on *deep reinforcement learning*), as depicted in Fig. 4. On the other hand, the irruption of ANN, particularly the Deep Neural Network (DNN) (also Deep Learning in the literature), meant a substantial improvement of the error rates for the different ML tasks, to the point of classifying the methods between the *classical* and the *neural-network-based* methods, or even more specifically *DNN-based* methods. The present survey follows both classifications in parallel. This

is because the provided classification is non-exclusive and that, consequently, methods of one category can be used with other types of learning. However, we have grouped the methods in the mentioned learning categories considering the most frequent learning technique, paying special attention to the area of routing optimization in SDN. Alternative criteria for classifying ML methods exist, such as arranging the methods according to the kind of training algorithm used (distinguishing between closed-form vs. iterative algorithm), or categorizing them attending to the final task in classification or regression methods.

There exists an additional orthogonal learning paradigm called *federated learning* which consists of a set of distributed learners which can be individually trained following one of the other mentioned learning paradigms and coordinately elaborate classifications or predictions. This special paradigm reminds us of the ensemble methods (random forest, boosting and bootstrap), but device distributed, which means both data and learning are individually used to create learners, even in different network nodes, whose predictions are then combined. Unfortunately, the authors did not find works that use this kind of learning for routing optimization in SDN, hence it was excluded of the classification. However, this approach is recently irrupting in near fields, such as mobile and wireless networks [75], [76].

A. SUPERVISED LEARNING (SL)

SL is a learning paradigm based on discovering the unknown function $f: X \rightarrow Y$ that relates the input and output spaces, X and Y respectively, from input-output pairs $(x_i, y_i) \in X \times Y$. This process is called *training* and requires a *labelled dataset* $\mathcal{D} = \{(x_i, y_i) \mid (x_i, y_i) \in X \times Y\}$ for the accomplishment of the task. Literally, supervised training algorithms infer the map f from the provided training data \mathcal{D} , typically minimizing a *loss function* \mathcal{L} which penalizes the committed error. Learning algorithms seek f in specific function spaces $f \in F$, most of them are parametrized, and consequently, the learning task becomes into an optimization problem:

$$f^* = \arg \min_{f \in F} \mathcal{L}(f(x), y) \quad (1)$$

Different parametric function spaces F with different learning algorithms correspond to the existent variety of supervised methods. The following methods are commonly considered as supervised methods, although some of them can be trained in an unsupervised way, or using a reinforcement learning strategy, and consequently, belonging to several categories:

1) Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) [77] consist on a set of connected units known as artificial neurons which emulate the biological neuronal networks of the animal brains. Due to their ability to model complex non-linear relations and their capacity to massively address data, they revolutionized the ML field. ANN-based effective applications include:

adaptive control, laser applications, medical areas, process logging, and energy areas. The Perceptrons and Multilayer Perceptrons (MLP) were the first architectures of ANNs. Also, ANN models relations described by dynamic systems, such as the Recurrent Neuronal Network (RNN) [78].

Deep Neural Network (DNN) [79] is a subcategory of the previous one, which bind together a huge amount of recent networks architectures which have in common the high number of interconnected layers. Deep Learning starts with the Convolutional Neural Network (CNN), a DNN with a sequence of convolutional layers configured in cascade. They are capable of extracting intrinsic local features, the called *deep features*, proving to surpass the result of its predecessor in both classification and regression task. Nowadays, the research efforts are focused on the improvement of the DNNs, as the amount of publications in this field proves. Autoencoders [80], Residual Networks (RESNET) [81] or VGG [82] are CNNs included in this category. DNNs also include networks for temporal sequence, such as, the improved RNN [78], which evolved to the novel Long-Short Term-Memory (LSTM) [83] and Gate Recurrent Unit (GRU) [84]; and the Random Neural Networks (RndNN) [85], which represent a set of cells that are connected in a network that transmits spiking signals. Some of these DNNs can also be trained using reinforcement learning algorithms.

2) Markov Decision Process

Markov decision process [86] is a kind of stochastic process in discrete time. They obey the Markov property which establishes that the probability to pass to a specific state in the next time exclusively depends on the current state. They try to find a good action policy for the decision maker which is affected by noise environment.

3) Linear Regression

Linear Regression [87] is one of the simplest and more effective ML methods. The linear regression assumes that a linear dependence exists between the dependent variable y and the explanatory variables (the independent variables). The simplest estimation algorithm retrieves the coefficients using mean-square-error. Robustness against outliers were introduced driving to the LASSO, Ridge or ElasticNet regressors.

4) Logistic Regression

Logistic Regression [88] is used for classification problems. It is based on the idea of probability and it uses predictive analysis algorithms. The Logistic Regression uses an increasing cost function. This cost capacity can be characterized as the *sigmoid function* (logistic function) rather than a linear function. Logistic regression confines the cost function in the range between 0 and 1. Both Linear and Logistic Regression are included in the called Generalized Linear Model (GLM), a wide model which unify various other statistical models.

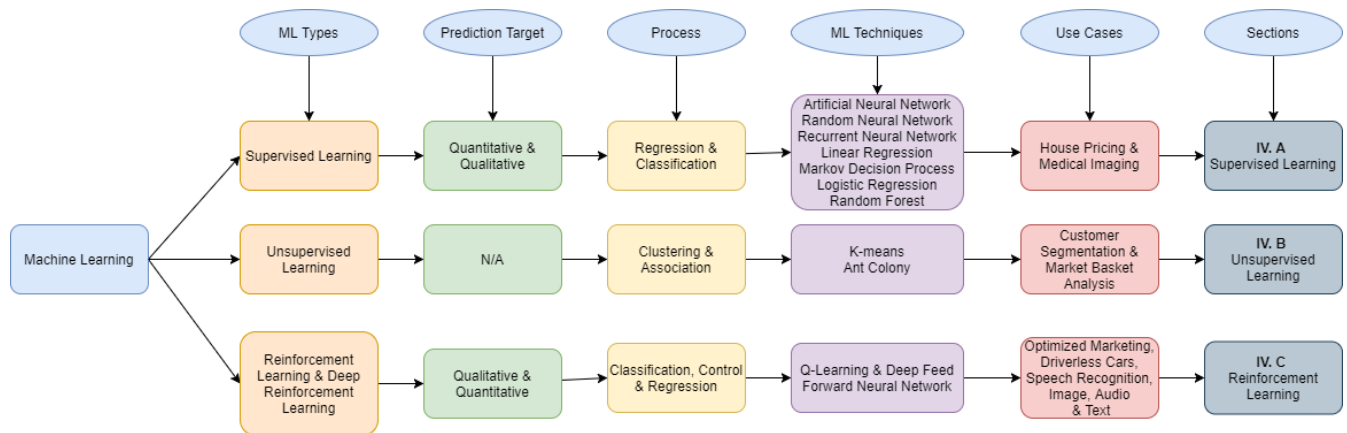


FIGURE 4. Classification of ML techniques

5) Random Forest

Random Forests [89] are supervised learning methods which assemble the result of a large number of decision trees of multiple sizes to estimate a unique value in regression or to yield a class in classification.

6) Evolutionary Algorithms

Genetic Algorithms (GA) are probability search algorithms inspired by the genetic mechanism of Darwinian natural selection and biological evolution. GAs provides the solution to deep problems by the reproduction process and code techniques. In many domains, GAs have been used with considerable efficacy.

B. UNSUPERVISED LEARNING (UL)

UL seeks patterns among unlabelled datasets. Contrary to SL, human supervision disappears due to lack of pre-labelled input-output pairs. Unsupervised methods self infer relations among the variables according to features such as orthogonality, correlations, statistical separability, etc. The *clustering* or *grouping methods* together with the one based on *principal components analysis* are the most common unsupervised methods, but not exclusively. Recently, we count on unsupervised DNN-based methods such as the Generative Adversarial Networks (GAN) [90].

1) K-means

K-means [91] is a ML algorithm, specifically, a vector quantization technique that seeks to group a number of observations $\{x_i\}_{i=1}^n$ in K clusters. This method minimizes the cluster variance. Each observation is associated to the cluster with the nearest distance to the *cluster centroid*.

2) Hierarchical Clustering

Hierarchical Clustering [92] groups near observations in clusters and establishes links between optimizing cluster dissimilarity. As a result, the method returns a partial ordered *dendrogram* which provides the data clusters with a hierarchy.

3) Self-Organizing Maps (SOM)

Self-Organizing Maps (SOM) [93] are ANN trained to retrieve a low-rank discrete representation of the input space, the called *map*, given the unlabeled training data. The method looks for the intrinsic topological properties of the input space.

4) Gaussian mixture models (GMM)

Gaussian mixture models (GMM) [94] assume that observations are generated by a mixture of a finite number of Gaussian variables. It is a probabilistic model which generalizes k-means modelling the uncertainty of cluster assignments by introducing the covariance to the problem.

C. REINFORCEMENT LEARNING (RL)

RL is another machine learning paradigm conceived to teach an *agent* to make local decisions and take *actions* in order to minimize a cumulative penalty or maximize a cumulative reward [95], [96], as illustrated in Fig. 5. Contrary to the SL and UL paradigms, the temporal variable is decisive, and the error metric is time distributed. In particular, in comparison with the supervised approach, RL does not count on labeled datasets. Feedback is obtained from the environment over the agent acts. Typically, Markov Decision Support (MDS) systems comprise the RL framework, where dynamical programming algorithms are used to maximize the reward. Recently, DNN-based frameworks were introduced and significantly improved this learning paradigm [97]–[99].

1) Q-learning

Q-learning [100] is a model-free RL method to teach the agent an action policy according to the state and the observations from the environment. As a model-free RL, the method does not use the transition probability. The method operates under an MDS framework finding an optimal policy using an expectation-maximization algorithm of the cumulative reward computed over all the successive steps, starting from the current state. Nowadays, it constitutes a baseline for the existing RL methods.

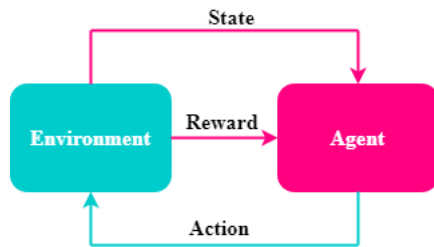


FIGURE 5. Reinforcement Learning

2) Double Q-learning

Double Q-learning [101] is an improvement of Q-learning which overcomes the problem of overestimation of the action values in noise environments, which results in a learning deceleration.

3) State-Action-Reward-State-Action (SARSA)

SARSA [102] is another RL method over MDS. The acronym shows that the updating function of the Q-value depends on five aspects, namely: the current state of the agent, the action the agent chooses, the reward the agent receives for choosing this action, the state that the agent enters after taking that action, and the next action the agent chooses in its new state.

4) Deep Reinforcement Learning (DRL)

DRL [103] is a subtype or subclass of RL that combines ANNs with RL models to enable SDN agents to learn the most efficient path and to achieve their goal. DRL incorporates ANNs to the agents in the RL framework. Traditional RL methods cannot solve high-dimensional decision making problems due to the high complexity of their states. ANNs bring better function approximation to the agent for making a decision, surpassing the mentioned disadvantage, which now can learn accurate policies $\pi(a|s)$ in a supervised way. It enables us to take the important decisions at wide range and solve them. Traditional DRL controllers [104] use fixed pre-processing steps, which are unable to adapt their processing state in response towards the learning signal. DRL is applied to many applications like robotics, healthcare centers, finance, smart grids and many more. The structure of DRL are shown in Fig. 6.

While DRL could be seen as part of RL and not as a differentiated type, we have specifically distinguished it from RL because, particularly during the last two years, there is a growing hype in its application in SDN environments and, for that reason, we believe it deserves its own analysis section. Due to its interesting for the community, we point out a special DRL method, the Deep Q-learning an evolution of Q-learning with ANNs.

5) Deep Q-learning

Deep Q-learning [97] substitutes the MDS framework with DNN and solves the problem of multiple states and massive data. The traditional Q-table, which keeps track of the states,

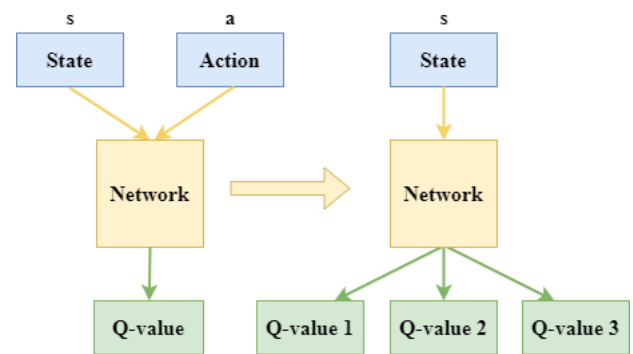


FIGURE 6. Deep Reinforcement Learning

actions, and their expected rewards, is now substituted by an ANN to predict both action and Q-value only from the state. Usually, its methods are based on RNNs, LSTMs and GRU, due its intrinsic evolutionary character, besides CNNs [98], [105].

D. SELECTING THE BEST ML METHOD

After introducing the different techniques, classified into three core types, we would like to provide a quick –and qualitative– overview of which technique or method seems to be more suitable for routing in SDN. There is no straightforward answer for this matter, and we could state that the best solution is strongly conditioned by several factors:

- 1) **Dataset type:** Scenarios where a labeled dataset is available allow the use of supervised ML methods, which are usually more accurate than its non-supervised counterpart. Learning from datasets permits to infer input-outputs relations that can be considered for routing. However, it is very important to have observations that cover the whole variability of situations. In this regard, we want to remark that, as we will examine within the following section, the majority of the works for routing in SDN use simulated datasets for training the ML algorithms. Only a few approaches directly work with real datasets, which better capture the real input-output relation than the synthetic ones. As the access to this kind of information is more difficult and the field does not count on standardized databases that allow testing the different proposals, unsupervised methods are frequently applied to find patterns in unlabeled datasets. On the other hand, RL is specific for dynamical optimization problems, such as, the routing optimization problem in SDN. RL methods have the ability of learning from the environment and adapting to the change of environment conditions. The agent must be trained maximizing a reward function from the environment instead of using a labeled database.
- 2) **Dataset size:** The size and nature of the database strongly constrains the type of ML method we can use for estimating routing parameters. Large databases are suitable for ML techniques that involve a huge

number of parameters such as ANNs or DNNs. Large databases also avoid the overfitting problem and allow to infer new input-output relations difficult to find in small datasets with a few observations. Nevertheless, the use of large databases requires long training time and expensive equipment, such as, graphic cards. The computation time for inferring the parameters tends to be higher than using small databases. Additionally, small datasets are more available and easier to manage for training any ML method than the large ones. However, they may not permit to infer complex input-output patterns.

- 3) **Problem type:** Many routing optimization approaches in SDN divide the routing task into sub-problems that can be individually solved by ML methods, such as, “maximum throughput & minimum cost”, “minimum congestion probability” or “bandwidth prediction” problems. From a ML point of view, we distinguished two different types of problems: classification and regression. In classification, we want to identify which category, from a finite set of different classes, an observation belongs to; while in a regression problem, we want to estimate real vectors that belongs to continuum intervals. ML methods are different depending on the type of problem to solve.

Considering all these factors, large datasets are appropriate for ANN-based and DNN-based approaches, which can extract interesting parameters from data. The difficulty of finding large datasets can be softened by a first training with synthetic database [106]–[108] and, afterwards, using a last fine-tuning step with a small real dataset. ANN-based methods suffer from overfitting if they are trained with medium-size or small dataset. With medium-size dataset, we can try support vector machines and the ensemble methods, including random forest. Specifically, random forest has proven to be faster than other ensemble methods since it is a tree-based ensemble. With small datasets, the best option is to use linear regressors, such as, ridge, lasso or elastic-net regressors, which are simpler but faster than the previous methods and, in most cases, effective enough [109], [110]. With no given dataset, unsupervised clustering methods are required. The most sophisticated unsupervised methods are the hierarchical clustering and the self-organizing maps, which even work with large unlabeled dataset. The more traditional method K-means is also used with medium-size databases [111], [112]. Similar to supervised learning, deep reinforcement learning should be applied in those scenarios where multiple iterations with the environment are permitted, specially the LSTMs and RNNs [113]–[115]. Neural networks need to be extensively trained. Otherwise, reinforcement learning methods based on MDS such as Q-learning or SARSA can be used [116], [117].

V. MACHINE LEARNING TECHNIQUES FOR ROUTING OPTIMIZATION IN SDN

As already presented, ML [118] can play a core role in optimizing routes in SDN, by saving time, money and ensuring the fast delivery of data within the required time. While traditional routing techniques [119]–[121] suffer from complex dynamics in networking, and face some problems such as performance declines and low convergence, ML is particularly appropriate for the SDN architecture, as it is capable of easily centralizing the information gathered in the network. Accordingly, ML together with SDN compose a thriving approach in the game of route optimization.

Although the overall procedure in ML is based on continuously retrieving data, training it, learning from it, predicting the new values and choosing the most efficient route, ML strategies might be utilized depending on the specific strategy and system requirements. In this survey, we comprehensively examine the state of the art of ML techniques that are implementable and applicable in SDN. To this purpose, we classify the ML techniques for routing optimization in SDN following the taxonomy of Section IV in three categories: *Supervised Learning* (SL), *Unsupervised Learning* (UL), and *Reinforcement Learning* (RL). The latter contains an additional subsection dedicated to *Deep Reinforcement Learning* (DRL), and its table is separated as well from the one of classical RL. The large amount of DRL methods in routing optimization of SDN justifies their exposition separately from the reinforcement learning methods, which strictly include them considering the theoretic taxonomy.

Afterwards, the works analyzed are ordered following the different techniques leveraged for the conceptual implementation. All of these ideas are summarized in Tables 3 and 4 for SL, 5 for UL, 6 for RL, and 7 and 8 for DRL, in which we classify the different ML works based on the following parameters: types of techniques, objectives, implementation and evaluation, and advantages and disadvantages. Additionally, this chapter is finalized by providing an overview of *learned lessons* and current research trends.

The order of appearance of the different works is chronological, but also based on the ML techniques used and relating proposals by shared sets of authors. In particular, we started from the oldest work in the different types of ML, and then continued with similar works (using the same ML technique) from oldest to newest, so that all proposals were somehow intertwined and following a logical timeline. We believed this approach could facilitate the description and understanding of the evolution of the different proposals, as strictly following a chronological order could cause the reader miss the relationship between approaches, as well as their pros and cons.

Finally, we would like to highlight that the present survey focuses on the different ML techniques found in routing optimization in SDN. Observe that most of the optimization techniques appear in the literature to complement the ML methods and subordinate to them. That is the case of Sabeeh et al. [122], who propose a hybrid intelligent system,

named Hybrid Intelligent Approach (HIA), which is used to optimize the performance of SDN. In most of the cases, optimization techniques are used for training the ML methods, reducing the number of features, or finding some important hyperparameters.

A. SUPERVISED LEARNING

Dynamic routing is a technique that forwards data using different routes based on given conditions or communication circuits. **NeuRoute** [106] is a framework of dynamic routing for SDN that leverages ML and solves the Maximum Throughput Minimum Cost Dynamic Routing Problem, achieving the same result as other dynamic routing algorithms, but requiring less execution time. NeuRoute is a dynamic framework that is controller-agnostic, which uses a neural network for learning traffic characteristics. Based on a real-time predict traffic matrix, forwarding rules are generated to optimize network throughput. To ensure a certain value of QoS, the common practice is to allocate more network resources than strictly required, based on peak traffic load estimation. In a case when peak loads are predictable, this practice of QoS is quite simple but in the long term, it is not justified economically. The basic motivation of NeuRoute is that, in dynamic routing, due to high computational complexity, the use of traditional algorithm solutions is not practical. Two of its main core blocks are based on DNN: the *traffic matrix predictor* and the *traffic routing unit*. The traffic matrix predictor is a LSTM which accurately predicts the next step. The traffic routing unit is designed with a FFN which learns how to match the traffic demands to the routing paths.

Chen-Xiao *et al.* [107] introduce a load balance resolution system with the benefit of a global network view for SDN. It increases the performance of data broadcasting in SDN. The principle is to outperformed legacy routers, which store routing tables that only contain destination network and next-hop information, hence missing a global routing view. The authors propose a mechanism in which the SDN controller discovers all paths between source node and destination node, and implements a load balancer application to efficiently distribute the traffic. The load balancer server maintains the load in each path [107] based on real-time metrics. More specifically, the load balancer immediately calculates all load conditions of multiple paths that are received from the SDN controller. After receiving the chosen path for transmission, the SDN allocates the flow tables for OpenFlow [136] switches to achieve a certain data-flow transmission. To this purpose, the authors propose an ANN composed by one single hidden layer (with a maximum of 11 neurons), which receives four load features as inputs, namely: bandwidth utilization ratio, packet loss rate, transmission latency, and transmission hop. The ANN infers the integrated load. The authors evaluate this architecture using Mininet and the Floodlight controller [137], and results suggest better performance and a decrease in network latency of 19.3%.

Wu *et al.* [123] present AIER, an ANN to predict the minimum congestion probability among all path configuration. The network is trained to predict the congestion given the loads for all data flows and all the available path configuration.

Sabeeh *et al.* [122] propose a hybrid intelligent system, named Hybrid Intelligent Approach (HIA), which is used to optimize the performance of SDN. HIA, whose architecture can be seen in Fig. 7, is a combination of multiple ML methods and techniques working together or parallel. The performance optimization of SDN is performed using a hybrid intelligent approach. The ML techniques, namely ANNs and Adaptive Network Fuzzy Inference System (ANFIS) [138], are used for mapping and modeling. Additionally, GA [139] and Particle Swarm Optimization (PSO) [140] are optimization techniques that give maximum performance of SDN by using the ANN model. In this paper, the authors performed the simulation of SDN by using Mininet and the POX controller, for collecting input and output datasets.

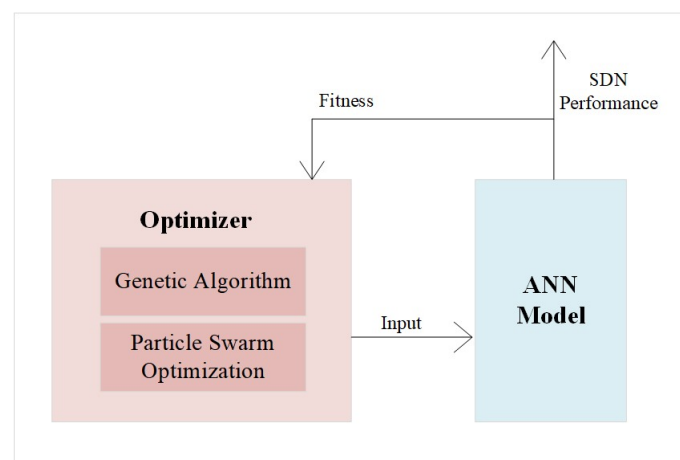


FIGURE 7. Architecture of the proposed model by Sabeeh *et al.* [122]

NeuTM, also proposed by Azzouni *et al.* [124], uses LSTM-RNNs [141] for traffic matrix forecasting. It applies a sliding window technique for obtaining the input-output pairs to feed the Neural Networks. The LSTM is a strong self-learning algorithm with the ability to detect complex non-linear patterns, widely used for time-series predictions. The results show that LSTM performs better than traditional RNNs and obtains high prediction accuracy in a very short training time.

Benamrane *et al.* [125] focus on SDN in avionic networks, where the complexity of security of communication, management, handover between radios, and QoS requirements are the major challenges. The interest of SDN in avionics is the ability to program the aircraft and the ground network devices in a unified and centralized way through software applications. The authors provides an adaptive bandwidth manager based on real-time traffic which runs on top of the SDN controller and ensures the QoS policy fulfillment for the aircraft critical and non-critical services.

TABLE 3. Comparison of supervised learning techniques for routing (1/2)

Ref.	Techniques	Objective	Implementation & Evaluation	Advantage	Disadvantage
Azzouni et al. [106]	ANN	Maximum throughput & minimum cost	POX controller + Mininet. GÉANT network topology and traffic	Fast execution. Min cost. Max throughput	Large networks and datasets not tested yet
Chen-Xiao et al. [107]	ANN	Load balance solution with global network view for SDN	Floodlight controller + Mininet	Enhanced bandwidth utilization ratio, transmission latency, packet loss rate and transmission hop.	DLB strategy not select best path, ignores the load condition, global path lead to the bad impacts.
Wu et al. [123]	ANN	Minimum congestion probability	Ryu controller + Mininet	Improvements in the average throughput, packet loss ratio, and packet delay versus data rate	Simplicity of the layout and the model. The model is not scalable.
Sabeeh et al. [122]	ANN + Evolutionary (HIA)	Maximum performance	POX controller + Mininet, and MATLAB	Cost effective, time effective, good performance index	It lacks proper / reproducible implementation details
Azzouni et al. [124]	LSTM-RNN	Traffic matrix prediction	POX controller. GÉANT network topology and traffic	Successfully applied. Best suited for sequence labeling task and sequence modeling	Traditional non-linear prediction models (ARMA, ARAR, HW) cannot meet the accuracy requirements
Benamrane et al. [125]	ARIMA, LSTM	Adaptive bandwidth manager	Floodlight controller + Mininet	Dynamic changes of QoS policy when the traffic flood the forwarding elements	The time series forecasting is an optional module
Rusek et al. [126], [127]	GNN	Enhanced per-source/destination pair mean delay and jitter estimation	OMNeT++. GÉANT, NSFNet, 50-node Germany50 topologies	Significant delay and jitter reduction	Large amount of data
Troia et al. [109]	Logistic Regression	Optimized routing. Traffic matrix prediction	ONOS controller + Mininet	Improves shortest path algorithm. Dynamically reduces network congestion.	Real datasets are needed for advance models and predictions for industrial applications
Wang et al. [110]	Linear Regression	Enhanced QoE	Theoretical analysis based on the SDN architecture	Best management strategy and performance. Ensures user requirements are met	Missing practical implementation and dataset
Sun et al. [128]	MACCA2-RF&RF	Intelligent routing by leveraging flow classification and avoiding congested links with local routing	Floodlight controller + Mininet. Moore and Li datasets	It can accurately classify flows to their obtain QoS requirements. Local routing adapts to provided QoS.	Evaluated with a relatively old dataset. Requires many entries in the SDN tables.
Choudhury et al. [129]	Random forest	Managing IP and SDN-enabled optical networks	Theoretical proof-of-concept study	Cost effective, better accuracy, enhanced robustness and dynamic capacity.	Missing practical implementation and dataset
EL-Garoui et al. [130]	Naive Bayes	Reduced delay and enhanced resilience	Ryu controller + Mininet-WiFi	Delay reduction compared to Q-learning, multipath, and OLSR routing protocols	Simplistic layout and model. Requires much data
Hardegen et al. [108]	DNN	Optimized flow routing	P4 switches	Low average delays achieved. It uses programmable P4 switches	Missing detailed implementation
Awad et al. [131]	DNN	Multipath routing framework with QoS constraints and flow rule space constraints	Keras (TensorFlow). TOTEM toolbox	High prediction accuracy of the heuristic routing solution and low computation time	Missing comparison with other algorithms. No thoughts on SDN implementation details and implications

TABLE 4. Comparison of supervised learning techniques for routing (2/2)

Ref.	Techniques	Objective	Implementation & Evaluation	Advantage	Disadvantage
Akbar <i>et al.</i> [132]	NSGA-II (Genetic algorithm)	Multi-objective: 1) minimize path delay, 2) maximize path reliability	POX controller + Mininet	Optimal paths for each type of traffic (UDP or TCP). Focus on real AI-based network applications (IoT and fog computing)	Missing comprehensive evaluation using different network topologies (only one fixed custom topology is tested).
Owusu <i>et al.</i> [133]	Random Forest, Decision Trees, K-nearest neighbors	Classify traffic in SDN-IoT networks	Dataset from a real-world ToR network	High accuracy rates above 0.8 with six features	Lack of comparison with any ANNs. Accuracy rates above 0.9? Lack of detail in the SDN implementation
Sacco <i>et al.</i> [134]	ARIMA, SVR, Decision Trees, Linear Regression, Random Forest	Bandwidth prediction	Ryu controller + Mininet. GENI testbed	Simplicity of the regressors. Real traffic traces	Missing comparison with the DNN-based regressor
Todorov <i>et al.</i> [135]	Q-learning, Genetic algorithm, Particle swarm optimization, Hidden Markov model	Architectural design for load balancing and segment routing	Theoretical analysis	It compares four supervised and reinforcement learning techniques	Simple architectural design. No thoughts on implementation details and implications

This bandwidth manager optionally includes a time series forecasting module based on ARIMAs and LSTMs capable to predict future bandwidth variations.

RouteNet, proposed by *Rusek et al.* [126], [127], is a new type of Graph Neural Network (GNN) specifically conceived for modeling computer networks. It is inspired by the Message Passing Neural Network (MPNN) previously proposed in the field of quantum chemistry. RouteNet is capable of capturing the complex relationships between topology, routing and input traffic to produce accurate estimations of the per-source/destination pair mean delay and jitter. It is trained with synthetic data generated by a custom-built packet-level simulator with queues using OMNeT++. The delay and jitter are related to the bandwidth capacity of each corresponding egress links. Using RouteNet as a SDN controller, the authors show the ability to optimize multiple Key Performance Indicator (KPI) and to guarantee the service-level agreements (SLAs) of a particular set of flows.

The **Machine Learning Routing Computation (MLRC)** module, implemented by *Troia et al.* [109] considers it is a big challenge to provide accurate and efficient quality communications to end-users due to the amount of data transported by current telecommunications networks. In this regard, the authors leveraged the ONOS controller [142] to build a machine learning model, called MLRC, to train and configure the optimization in charge of finding the different paths in the SDN network. MLRC implements a logistic regression classifier due to its simplicity and explainability. According to their results, the SDN network is able to recompute its routing configuration and execute it in a very limited lapse of time for any incoming shift in the

traffic matrix. However, the authors anticipated their results are limited and real datasets could facilitate more advance models for optimized routing in real networks with industrial applications.

Wang et al. [110] present a module based on machine learning and implemented in SDN to enhance QoE. It chooses the best path, monitors, and controls and predicts the performance of the network. The researcher uses quality of experience (QoE) [143] to evaluate the performance and condition of the application. An optimal QoE is difficult to achieve for real-time applications, so a set of Key Performance Indicators (KPIs) [144] was defined. Moreover, their SDN module works both with information acquired from both the SBI and the NBI, as the SBI collects the network matrices and the NBI collects KPIs.

Sun et al. [128] combine a variety of ML algorithms to propose a data flow classification method called MACCA2-RFRF, which identifies the data flow category (with almost perfect accuracy) and obtains the QoS requirements. The authors comprehensively evaluate their proposal with real datasets and an SDN implementation based on Floodlight and Mininet, which is quite close to real scenarios. However, some parts of their design still need improvement, such as the amount of table entries installed, which should be reduced to be scalable.

Choudhury et al. [129] introduce ML to control more efficiently SDN-enabled IP/Optical Networks [145] with SDN. The Open ROADM (Reconfigurable Optical Add-Drop Multiplexer) [146] concept together with the SDN controller tools permit the ISPs to more efficiently and homogeneously obtain network performance data to set up the best wavelength paths that meet the requirements of optical networks. For

this purpose, ML is used to predict the best performance of wavelengths in multiple vendors. In their architecture, SDN controls all-optical routers, all-optical nodes, edge routers, and optical nodes, hence providing a global view. In the end, the authors defined two applications in ML that are managing IP and optical networks. The first application provides the facility of long-term prediction with global optimization, while the second produces short-term traffic prediction that helps out in reducing the customer traffic on the network.

EL-Garoui *et al.* [130] leverage SDN and ML for efficient routing in smart cities, where most applications are based on Internet-of-Things (IoT). They develop a framework based on the Naive Bayes algorithm and create a dataset based on the Montreal city open data website and the SUMO urban mobility simulator. After comparison with other protocols, like OSLR, obtaining better results in terms of delay and packet delivery ratio.

Hardegen *et al.* [108] present PFR, which is a flow routing paradigm that aims to efficiently distribute traffic (nearly evenly) over links/paths to avoid high load/congestion. Conditions for flows can be improved by minimizing observed latency/maximizing required throughput. The authors briefly provide a summary of the ML techniques employed. They continuously train a DNN on incoming data while treating the prediction of flow characteristics as a multi-class classification problem. As forecasting is carried out as flows start, only features known ahead of time are usable. Besides a continuous model update, an interface to request a prediction for flow 5-tuples is offered. Finally, a key aspect of this approach is that the authors implement their solution using P4 programmable switches, instead of following the classic centralized SDN model.

Awad *et al.* [131] focus on a rather theoretical analysis of enhanced multipath routing using DNNs. Although they leverage the TOTEM open source traffic engineering toolbox [147] (supported by experts in the field of computer networks) and their evaluation is pretty comprehensive, they do not provide any insights on actual SDN implementations, which limits the scope of their proposal.

Akbar *et al.* [132] design one of the few works analyzed that focuses on real computer network scenarios leveraging AI and SDN. In particular, they present a proposal based on genetic algorithms to achieve adaptive and reliable communication in IoT-fog environments, which could be considered one of the main objectives of the future 6G networks [148], [149]. The authors implement an SDN-based framework to evaluate their proposal and leverage real datasets. However, the evaluated topology is only one fixed custom topology.

Owusu *et al.* [133] propose diverse implementations of ML models to classify traffic in SDN-IoT networks for traffic engineering. The authors compared three different classifiers: Random Forest Classifier, Decision Trees Classifier and K-Nearest Neighbors Classifier. Also they evaluate two feature selection methods: Sequential Feature Selection (SFS) and Shapley additive explanations (SHAP). According to their analysis, the best accuracy rate, 0.83, is obtained by the

random forest classifier with SFS.

RoPE, proposed by Sacco *et al.* [134], is an architecture that adapts the routing strategy of the underlying edge network based on future prediction bandwidth. RoPE is a conglomerate of supervised time-series models and machine learning methods train to predict the bandwidth in such a way the controller can check whether the desired application fits the network load. It automatically chooses the algorithm to apply, in order to guarantee the best possible performance. Choosing the right forecasting method for a given use case is a function of many factors such as the historical data available and exogenous variables (e.g., weather, concerts). Data for training is collected via the Mininet emulator. As a result, the SDN controller tracks the past link loads and takes a new route if the current path is predicted to be congested.

Finally, **Todorov *et al.*** [135] present an architectural design to implement four types of ML techniques to improve load balancing and segment routing in SDN. However, the article does not provide any additional insights on implementation nor provides any type of evaluation.

B. UNSUPERVISED LEARNING

Budhbraja *et al.* [111] state that usual SDN routing approaches do not usually follow privacy and compliance requirements of data transmission. This is particularly magnified considering the fact that SDN routes are usually static or defined specifically for each communication flow, which is prone to suffer from diverse security attacks like, for instance, Denial of Service (DoS). If such a kind of routing is performed in a controlled environment (HIPAA), we can lose important information in case of an attack. In this paper, the author focus on the privacy of sensitive data transmission and the restricted challenges of compliance in SDN environments. Since a big number of packets transmitted via the same data path is considered as a risk, route randomization is performed by monitoring the forwarding path and its transmitted packets. The required results are obtained by using i) ML and analytics for the computation of risk in SDN network; ii) distributed routing based on swarm algorithm; iii) minimizing the route randomization and risks for achieving the requirement of compliance and privacy. The proposed scheme works on history, as it collects previous packets for the purpose of training and then data packets are efficiently routed. For risk identification, the K-means clustering algorithm is used. It identifies k-centroid objects for finding the risk ratio, and it is processed offline. The risk is analyzed and then for routing data packets the online method is used to make a real-time decision. Ant colony optimization is used for making real-time decisions with low complexity level.

Kumar *et al.* [112] explore the applicability of ML algorithms for selecting the least congested route for routing traffic in SDN. The proposed method of route selection provides a list of possible routes based on the network statistics dynamically provided by the SDN controller. The authors propose two ML methods: a K-means clustering algorithm

TABLE 5. Comparison of unsupervised learning techniques for routing

Ref.	Techniques	Objective	Implementation & Evaluation	Advantage	Disadvantage
Budhraj et al. [111]	K-means	Minimum privacy risk, while achieving compliance requirements of data transmission	Ryu controller + Risk simulations in Python	Provides privacy and risk compliance with low complexity	Delays in communication
Kumar et al. [112]	K-means / Vector Space Model (cosine similarity)	Least congested route for routing traffic	Ryu controller + Mininet	Best Round Trip Time in comparison to Dijkstra	Simplistic network layout

and the Vector Space Model with cosine similarity. The proposed methods are tested in Mininet using the Ryu controller and they made a comparison with Dijkstra's routing algorithm. The experiments shows that the best Round Trip Time (RTT) measurement of the traffic flows is achieved by the implemented K-means closely followed by Vector Space model, surpassing the times obtaining by Dijkstra.

C. REINFORCEMENT LEARNING

Lin et al. [116] emphasize the urgent need to define a reliable QoS routing mechanism for large-scale SDN-based networks. To solve this issue, they propose QoS-aware adaptive routing in multi-layer SDN. The architecture of hierarchical distributed control planes is introduced by combining the work of Kandoo [154] and Xbar [155]. Levels of this distributed control plane are Super Domain (master), switch subnets and slave controllers. Thanks to a RL, the authors achieve a reliable SDN infrastructure and minimum signal delay, later on expanded with time efficiency, and QoS aware of packet forwarding. This QoS-adaptive routing outperforms conventional Q-learning.

Rischke et al. [150] consider addressing diverse and varying traffic loads implies the utilization of complex model, hence they focus on achieving a model-free RL scheme. Their proposal, QR-SDN, creates multiple paths between source and destination, which achieves substantially lower flow latencies. However, they devise additional research efforts are needed to conceive a scalable approach as the network size increases.

Casas-Velasco et al. [151] introduce a routing approach entitled Reinforcement Learning and Software-Defined Networking for Intelligent Routing (RSIR), which is founded on the need of adding a Knowledge Plane, as mentioned in Section III.B, to the network, which is fed by data gathered by the Management Plane. In particular, they define a proactive RL-based routing algorithm based on link-state metrics and implement it in a prototype with real traffic matrices. RSIR is compared against the classic Dijkstra's algorithm, which is leverage by most routing protocols. Results show that RSIR obtains more shortest paths and is able to better balance the load, hence reducing the overall latencies. As future work, they envision the evolution of their approach to DRL.

Fang et al. [117] consider that Dijkstra-based routing algorithms might have problems, particularly when data streams

are combined by selecting the same forwarding path, which greatly reduces the use of network connections and leads to network congestion. As SDN is not constrained to any particular routing algorithm, the authors consider the application of RL, with a Q-learning-based routing algorithm, specifically for comparison against the RIP protocol. Additionally, by combining RL and NNs, which means the Q-table in Q-learning is replaced by a NN, the authors present a Deep Q-learning-based routing algorithm as well. Both algorithms are simulated and exhibit good performance results.

Sendra et al. [152] presents a solution to enhance network performance based on QoS and security concerns. The solution is implemented in a distributed manner only with Mininet and no controller, to facilitate testing a proof-of-concept. Their solution involve the application of reinforcement learning over the traditional OSPF routing protocol, using Quagga, which permits modifying the routing algorithms. It is tested and compared against the conventional OSPF routing protocol and results show that it enhances OSPF, obtaining more stable routes, with lower loss rates and better jitter and delay.

Valadarsky et al. [153] focus on data-driven routing and present some preliminary results in the context of intra-domain traffic engineering. They perform an analysis applying both supervised and reinforcement learning in a complementary way (reinforcement learning takes past values from the traffic demands and trains the values, while it assumes the future values or traffic demands with the help of supervised learning). However, no specific effort is performed to integrate this idea in SDN scenarios, although the authors leave it as future work.

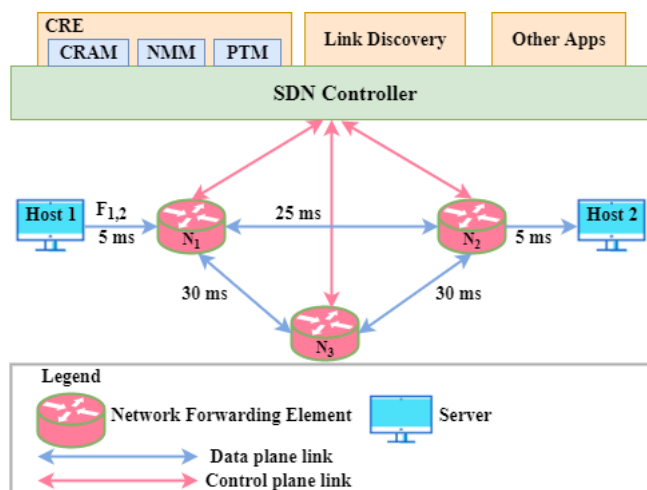
1) Deep Reinforcement Learning

Francois et al. [156] propose a new routing application called Cognitive Routing Engine (CRE) that enhances the efficiency of the processing and gathering of network states, and provides the best routing path that according to QoS requirements. The authors particularly consider the cloud provider use case, which typically needs dynamic re-routing for the different tenants, and focus on the design of the CRE module as an SDN application, as depicted in Fig. 8, in which the CRE application sits at the same level of the link discovery service. CRE is based on RNNs and tested in a

TABLE 6. Comparison of reinforcement learning techniques for routing

Ref	Techniques	Objective	Implementation & Evaluation	Advantage	Disadvantage
Lin <i>et al.</i> [116]	SARSA	Time-efficient and QoS-aware routing in large scale SDN	Simulated scenario + OpenFlow-compliant. It outperforms conventional Q-learning	Time-efficient, minimum signal delay	Missing experiments with fully SDN-compliant networks, including a controller
Rischke <i>et al.</i> [150]	Q-learning	Model-free proposal. Routing paths in its state-action space	Ryu SDN controller + Mininet	Multipath routing and reduced latencies. Comprehensive evaluation	Still lacks scalability in large scenarios
Casas-Velasco <i>et al.</i> [151]	Q-learning	Use of the Knowledge Plane concept. Best throughput, loss ratio, and delay + Obtaining best set of shortest paths	Ryu controller + Mininet. GÉANT network topology and traffic	Best metrics and enhanced set of shortest paths in comparison with Dijkstra. Very complete implementation and evaluation	Application to commercial SDN solutions (e.g. ONOS) would be desirable
Fang <i>et al.</i> [117]	Q-learning + Deep Q-learning	Improved network performance based on QoS	Simulated scenario + RIP protocol. After a certain training period, the algorithm can find a route with better QoS efficiency with almost 100 per cent accuracy	Better QoS connection and stronger link selection trend	The specific features must be designed manually, which is not trivial. No integration in real SDN scenarios
Sendra <i>et al.</i> [152]	Unspecified	Improved network performance with decision-making based on QoS and security	No controller (distributed, OSPF) + Mininet. Better jitter performance than delay results	Routing based on the open source tool Quagga, hence easily reproducible	Missing experiments with fully SDN-compliant networks, including a controller
Valadarsky <i>et al.</i> [153]	Data-driven model	Development of a data-driven model for routing optimization	Theoretical analysis + Results via simulation	Minimizes routing link utilization.	No integration with SDN or additional discussion about it.

Mininet scenario, but not exhaustively compared with other approaches. **Francois *et al.*** [113] updated their previous work by a practical scenario based on specific data center locations, plus the use of the Floodlight SDN controller.

**FIGURE 8.** Francois *et al.* [113], [156] present the CRE architecture that enhances the processing efficiency by gathering the network states according to the QoS requirements

Sun *et al.* [114], [157] combine the Recurrent Neural Network (not to be confused with RNN) with Deep Deterministic Policy Gradient (DDPG) [181] to model TIDE,

which proves to reduce network delay, as compared to standard shortest path routing schemes, like OSPF. In TIDE, the network model is represented as traffic data sequences in the router. The evaluated is performed via a realistic scenario based on Pica8 switches (well-known commercial SDN-capable hardware switches) and the POX SDN controller. In this experiment, 1000 training steps are present in each RNN-DDPG, and for performance measurement the average transmission delay is added in the total. After some time, it is observed that RNN-DDPG performs better as compared to shortest path. Although the results are promising, the authors foresee scalability issues in bigger scenarios. For this reason, a new work by **Sun *et al.*** [158], [159], entitled SINET, is presented afterwards specifically focused on scalability, in which partial control is applied together with DRL. SINET is evaluated via the OMNeT++ packet-based simulator, showing very good preliminary results. Finally, **Sun *et al.*** [160] present an updated solution for enhanced and scalable traffic engineering (similarly to their previous work), entitled ScaledDRL, in which they leverage the idea from the pinning control theory to select a subset of links in the network (set as critical links) and provide decisions based on them, hence fostering scalability. Their implementation is performed just with the OMNeT++ simulator, which might seem limited.

Stampa *et al.* [161] focus on the KDN concept to design a DLR agent to minimize network delay. The RL agent uses

TABLE 7. Comparison of deep reinforcement learning techniques for routing (1/2)

Ref	Techniques	Objective	Implementation & Evaluation	Advantage	Disadvantage
Francois et al. [113], [156]	Random Neural Networks (RNNs)	Secure traffic engineering based on QoS for cloud providers with low monitoring	Floodlight controller + Mininet. GÉANT network topology and traffic. Compared to IP (shortest path)	Better round trip time than IP. Reduced monitoring overhead	Relatively small evaluated network. More QoS parameters should be measured to prove the approach
Sun et al. [114], [157]	Recurrent Neural Network + DDPG	Reduced delay	POX controller + Pica8 switches. OS3E network topology	Reduced delay in comparison with shortest path. Realistic evaluation scenario	Poor scalability
Sun et al. [158], [159]	DDPG	Enhancing overall scalability in comparison to other DRL approaches	OMNeT++ simulator. OS3E, NSF and BRITE-generated network topologies	Partial control shows very good preliminary results	Evaluated only via simulation
Sun et al. [160]	DDPG	Traffic engineering via combination of DRL and pinning control theory focused on scalability	OMNeT++ simulator. OS3E, NSF and BRITE-generated network topologies	Improves delay	Throughput is not tested. Traffic workload is not real. Evaluated only via simulation
Stampa et al. [161]	DDPG	Reduced network delay via a DRL agent for routing optimization	OMNeT++ simulator. Scale-free network topology	One-step, model-free, black-box optimization	Evaluated only via simulation. Few details about the design
Yu et al. [162] & Maheswari et al. [163] & Xu et al. [164]	DDPG	Enhanced throughput and delay, while keeping reduced convergence time	OMNeT++ simulator. Sprint backbone network. Compared against OSPF	DROM dynamically adjusts the reward function, it does not rely on specific network states and achieves better results than OSPF	DROM requires the definition of a strategy, which cannot be defined automatically (and requires human intervention)
Yao et al. [165]	DDPG	Enhanced routing based on a hybrid approach (distributed+centralized)	OMNeT++ simulator	Quick average delivery time. Promising architecture	Evaluated only via simulation. Huge amount of data and training iterations
Zhang et al. [166]	DDPG	Content-aware traffic engineering for SDN	Event-driven simulator. GÉANT, NSFNET and BRITE-generated network topologies	Best throughput and bandwidth utilization compared to classic algorithms (e.g. shortest path)	Evaluated only via simulation
Nahar et al. [167]	DDPG	Enhanced cluster stability and route selection method for routing in VANETs	OMNeT++ simulator + SUMO simulator	Improves delay, throughput and computational overhead.	Evaluated only via simulation. Lacks study including effects like driver behaviour, road conditions, and real-world scenarios.
Tu et al. [115]	DDPG + LSTM	Enhanced throughput and delay, focused on topology changes in space-ground integration networks	OMNeT++ simulator. CERNET+NSFNET topologies + 3-layer satellite network. Compared to OSPF	Better results than OSPF in terms of throughput and delay	Evaluated only via simulation
Quang et al. [168]	DDPG + Convolutionary Neural Networks	Reduced latency and packet loss rate	OMNeT++ simulator. BtEurope network	It admits diverse configuration as input parameters	Evaluated only via simulation. No comparison with other approaches is performed
Swain et al. [169]	DDPG + Convolution layer	Reduced latency and packet loss rate	OMNeT++ simulator. Compared to OSPF	It outperforms OSPF in terms of latency and packet loss	Evaluated only via simulation
Lu et al. [170]	DDPG-EREP	Optimized routing (no specific parameters involved)	Ryu controller + Mininet	Improves the original DDPG algorithm.	Slow reading of information on complex topologies. More tests should on more topologies and traffic workloads.

TABLE 8. Comparison of deep reinforcement learning techniques for routing (2/2)

Ref	Techniques	Objective	Implementation & Evaluation	Advantage	Disadvantage
Liu <i>et al.</i> [171], [172]	DDPG + Deep Q-network (DQN)	High performance routing in data center networks	OMNeT++ simulator. Fat-Tree topology. Compared to OSPF and TIDE [157]	It outperforms OSPF and TIDE in terms of throughput, flow completion time and load balance	Evaluated only via simulation
Fu <i>et al.</i> [173]	Deep Q-network (DQN)	High performance routing in data center networks, differentiating mice and elephant flows	Ryu controller + Mininet. Fat-tree data center topology. Compared to ECMP [174] and SRL+FlowFit [175]	It outperforms ECMP and SRL+FlowFit in terms of throughput, delay and packet loss	Missing a wider range of topologies. No detail about traffic matrices
Jalil <i>et al.</i> [176]	Dueling Deep Q-learning (Dueling DDQN)	Computing path based on multiple QoS metrics (delay, bandwidth, loss, cost)	Ryu controller + Mininet. NSFNet and 10-node topologies. Compared to other greedy routing	Good results in terms of cost, loss and bandwidth, with acceptable delay	Overall gain is low. Missing detail about traffic matrices
Chen <i>et al.</i> [177]	Dueling Double Deep Q-learning (Dueling DQN)	Enhanced throughput and delay	Ryu controller + Mininet. Fat-tree, NSFNet and ARPANet topologies. Compared to OSPF and LL	Good results in terms of reward, file transmission time, and utilization rate metrics	Missing analysis of monitoring cost
Etengu <i>et al.</i> [27]	Deep Q-learning + SARSA	Energy-efficient routing and guaranteed QoS	N/A (Only architectural design)	Detailed explanation of the architecture	Missing synthetic or real experiments and comparison
Jha <i>et al.</i> [178]	Deep Q-learning + LSTM	Optimized multipath routing in DCNs (DRL to compute links weight and Dijkstra's to select optimal paths)	POX controller + Mininet. Fat-tree topology	Improves ECMP	Evaluated only with a few tests and not using DC-based workloads. Missing in-depth design details
Srivastava <i>et al.</i> [179]	Bio-inspired Restricted Boltzmann Machine (RBM)	Optimized load balancing	C++/WILL API. Fixed mesh topology	Better results than OSPF and DL	Evaluated with a few tests and not considering the usual performance metrics
Babayigit <i>et al.</i> [180]	Unspecified	Optimized load balancing in DCNs	Floodlight controller + Mininet. Fat tree topology. Traffic generated with Iperf	Compared with other ML techniques such as: ANN, SVM and logistic regression (all worse than the author's proposal)	Missing details of the DRL technique implemented. Limited to DCNs

three signals that are state, action and reward, to provide a near optimal solution. The RL agent is an off-policy, actor-critic, deterministic policy gradient algorithm that exchanges these three signals for interacting with the network.

Yu *et al.* [162] propose the DDPG Routing Optimization Mechanism (DROM). DROM is based on neural networks, not Q-tables, which saves time and storage, and works in continuous time with effective black-box optimization. The evaluation is focused on delay and throughput, in comparison with the well-known OSPF protocol, and the authors additionally measured convergence time, obtaining good simulation results. **Maheswari *et al.*** [163] and **Xu *et al.*** [164] present a very similar work to DROM, following the same approach.

Yao *et al.* [165] exploit a hybrid ML paradigm that combines a distributed intelligence, based on units called "AI routers", with a centralized intelligence, called the "network mind", to provide different network services. Using this paradigm, the authors deploy centralized AI control for connection-oriented tunneling-based routing protocols, such

as, multiprotocol label switching and segment routing, to guarantee a high QoS. Besides, for hop-by-hop IP routing, the authors shift the intelligent control responsibility to each AI router to ease the overhead imposed by centralized control and use the network mind to improve the global convergence. The work provides a DRL-based algorithm for an effective routing policy generation. The authors apply a DDPG approach for policy generation [182]. A DDPG agent has two main components: a deterministic policy network, the called actor, which attempts to improve the current policy; and a Q-network, the called critic, which evaluates the quality of the current policy. An iterative alternation between both actors reach the optimum policy. The authors simulate their proposal with OMNeT++. Experiments prove that with increasing load intensity, the AI-based routing achieves better performance than shortest path routing.

Zhang *et al.* [166] apply deep neuronal networks for content-awareness and exploit DRL for traffic engineering decisions. They provide a parallel online learning mechanism

to use DRL that has trial-and-error nature. They improve network performance in terms of total network throughput, bandwidth utilization, and load balance.

Nahar *et al.* [167] apply SDN-enabled spectral clustering-based routing together with DDPG to define SeScR. The special thing about this proposal is that the objective are not packet-based networks, but Vehicular Ad-Hoc Networks (VANETs) instead. For evaluation, they used OMNeT++ together with SUMO, a popular traffic simulator.

Tu *et al.* [115] highlight the existing challenge for optimized routing in space-ground integration networks, particularly when changes occur in the topology and link status. For that purpose, they define the ML-SSGIN framework, which uses the DDPG algorithm and a neural network that integrates LSTM and Dense layers. They compared their proposal with OSPF, obtaining better results in terms of throughput and delay.

Quang *et al.* [168] also leverage the concept of KDN to apply the ML principles in SDN environments. In order to improve the performance of QoS-aware routing, the author exploit a DRL agent with Convolutionary Neural Networks in the KDN context to improve latency and packet loss rate. The results obtained show that even in complex networks, the proposed approach can significantly improve the performance of the routing configurations. By proposing a DDPG algorithm, the authors address the continuous control needs. The OMNeT++ discrete event simulator (v5.4.1) was used to obtain the latency and packet loss rate.

Swain *et al.* [169] propose the Convolutional Deep Reinforcement Learning (CoDRL) model, consisting of a DDPG agent coupled with a Convolution layer. The authors simulate the environment with OMNeT++ and show that CoDRL clearly outperforms OSPF in terms of delay and packet loss.

Lu *et al.* [170] design an enhanced version of DDGP entitled DDPG-EREP, and they evaluate it with an emulated network (composed by the Ryu SDN controller and Mininet), instead of using a simulator (as the previous works). However, their evaluation is limited to a single execution of a fixed topology and additional tests should be performed to prove the benefits of their approach.

Liu *et al.* [171], [172] particularly emphasize on the need for optimized routing in data center networks. Their approach focus on the specific needs of these types of networks and how resource allocation and routing affects the overall performance of software-defined data center networks. For this purpose, they employ Q-network (DQN) and DDPG to build their model, DRL-R. After an extensive evaluation performed via simulation in OMNeT++, their results outperform those of traditional OSPF and TIDE (another DRL-based routing model previously mentioned).

Fu *et al.* [173] propose a routing strategy based on deep Q-learning (DQL) specifically designed for data center networks. In particular, the authors consider that mice and elephant flows (usual types of flows in data center networks) have different requirements: both need low packet loss, but reduced delay is more important in mice flows, while high

throughput is more relevant for elephant flows. Their proposal outperforms ECMP [174], the classic routing algorithm for data center networks, and SRL+FlowFit [175], which is an improved routing algorithm in comparison to ECMP and focuses on balancing the network load in folded-Clos data center topologies.

Jalil *et al.* [176] present Deep Q-Routing (DQR), which uses dueling deep Q-network with prioritised experience replay to compute a path for any source-destination pair request in the presence of multiple QoS metrics, such as delay, bandwidth or loss. They compare their approach with other existing learning methods for greedy online routing, showing better results in terms of loss and path cost, while keeping the best bandwidth most of the times and a reasonable delay.

Chen *et al.* [177] comprehensively analyze the need for optimized routing in SDN and present RL-Routing. After an extensive evaluation based on a real SDN controller and networks, RL-Routing proves to offer better results than other routing algorithms like OSPF and Least Loaded (LL).

Etengue *et al.* [27] propose a DNN-based approach in a hybrid SDN/OSPF network deployment. The SDN controller performs energy-efficient routing and enhanced performance with QoS guarantees. It is composed by both the SDN-enabled supervised ML module and the DRL module. The hybrid SDN-enabled supervised ML is formed by an LSTM to perform traffic flow prediction using time-series datasets, which extracts short-term network data traffic variabilities and periodicities to ensure traffic flow prediction and energy-efficient routing with guaranteed QoS performance. The DRL module performs learning from the existing historical data and iteratively from the interfacing with the defined network setting.

Jha *et al.* [178] focus on multipath routing in Data Center Networks (DCNs) and, for that reason, they directly try to compete against Equal-Cost Multi-Path (ECMP), which is one of the most popular protocols in those scenarios. In their design, they use DRL to compute the links weight and, afterwards, they apply Dijkstra's algorithm (as other traditional approaches). Although their evaluation is performed via an SDN-based environment, it does not consider typical traffic patterns from DCNs (such as elephant/mouse traffic), the tests are not comprehensive, and in-depth details from their implementation are missing for reproducible research.

Srivastava *et al.* [179] present a bio-inspired RBM algorithm to optimize load balancing. However, their analysis and evaluation seems limited, as they do not consider the measurement of standard metrics, the network topology is a fixed mesh (which is not common in practical networks) and they do not provide any additional thoughts on the actual SDN deployment.

Babayigit *et al.* [180] focus on DCNs and evaluate and compare a DRL technique with others like ANN, SVM and logistic regression. The results show that their approach is very efficient for load balancing, outperforming all the rest in diverse evaluated parameters. However, the authors do not provide specific details of the technique implemented, which

makes it hard to reproduce.

D. LEARNED LESSONS AND RESEARCH TREND OVERVIEW

After examining the works that apply ML together with SDN for optimal routing, several conclusions arise at first sight:

- Since the publication of the KDN concept four years ago, there is a huge tendency to apply ML and AI in SDN environments (particularly towards 6G) and, in the case of routing, DRL is particularly relevant in the last two years, as most published works fall in this type of ML technique.
- Most works compared their proposal with shortest path algorithms in terms of latency and/or throughput, and either use OMNeT++ for simulation, which might not be realistic enough, or leverage the Ryu SDN controller, which is very easy and good for prototyping, but it does not follow the requirements of the industry (e.g. bad performance, as it is written in Python).
- Selected topologies and datasets are often very specific and differ among authors. Only a few works use several types of topologies and datasets to guarantee comprehensive and homogeneous evaluations.
- Few efforts have been made to create synergies or even compare the different ML works in relation with routing in SDN. Most evaluations performed just compare their approaches with classic routing protocols and no competing proposals (probably because implementations are usually not publicly available), which hinders the attainment of actual conclusions.
- Most proposals lack design and/or implementation details, which makes it a hard task to reproduce results or produce comprehensive comparisons. For example, DDN-based proposals do not detail their architectures and the parameters used in their networks.

Apart from these four main learned lessons, there are some other trends observed in our analysis. For example, most designs propose a centralized architecture, following the idea of classic SDN, while distributed or hybrid SDN approaches are set aside. In the case of evaluation, most proposals agree on the use of topologies like GÉANT, NSFNET and BRITe-generated, which are consistent with practical implementations, although almost all are wired networks. These topologies are usually deployed with Mininet via Open vSwitches (we assume, as most works omit this specific –yet important– detail). As for datasets and traffic pattern generation, there is a huge heterogeneity of approaches: some leverage existing datasets, some others directly generate their own traffic based –or not– in current literature analysis, while many directly omit to provide details about this technical aspect.

Finally, the majority of works agree that future research efforts should be made regarding three aspects, namely: (1) scalability enhancement, (2) evaluation with more types of (real) datasets and (3) automatic fine-tuning of the system (which needs some manual configuration in the very first stages).

As a conclusion, following the definitions, descriptions, and evaluation of the different proposals presented, we believe the most complete and/or promising approaches are the following:

- Sacco *et al.* [134], as they realize a comprehensive analysis with a testbed close to practical scenarios, including real traces, and application and comparison of different techniques.
- Hardegen *et al.* [108], because they leverage P4 programmable switches, which might have the best performance over other implementations.
- Casas-Velasco *et al.* [151], since they present a very complete implementation and evaluation and leverage the KDN concept.
- Fu *et al.* [173], because they particularly focus on a type of scenario (data center networks) and carefully design their approach around it.
- Chen *et al.* [177], as their implementation and evaluation is very complete, and close to real scenarios.

Therefore, we recommend to follow the work from these research teams in case of interest in the field. Additionally, just out of curiosity, all of these five research items were published in 2020, which shows the very recent trend in the field.

VI. FUTURE RESEARCH DIRECTIONS

ML and AI have already influenced almost every field of human life [183]. Although ML algorithms are mostly leveraged for robotics, image and signal processing, they are playing an undeniable role in network control and management as well [184]. In particular, ML has been applied to routing problems in computer networks as early as in 1994 [185] and rapidly evolving everyday [186].

Recently, SDN has emerged in the field to provide a wider range of possibilities in the field of routing optimization with ML, as seen in previous sections. Nevertheless, this field still demands immense research efforts towards full-fledged ML-based networking environments, which we discuss in detail in the following sections. Though these challenges could be considered a burden, we believe they indeed illustrate an opportunity towards real and practical next-generation networks. For this reason, for each of the five sections, we will summarize the envisioned future research directions, together with the overall goal, in case these could hopefully serve as inspiration for the research community.

A. WHAT IS OPTIMAL ROUTING?

Though it might seem trivial, this is the first question that should arise when trying to design optimized routing algorithms based on ML for SDN environments. Networking scenarios are vast and heterogeneous and, for sure, not limited to be assessed by latency and throughput. Hence, when asked about the definition of optimal routing, the initial answer should be *it depends*.

For instance, first of all, in physical terms, networks could be divided into two main types: wired and wireless, and

they have different routing protocols to start with. As an example, latency and throughput could be valid parameters to measure routing quality in wired environments, but some wireless scenarios, like Low-power and Lossy Networks (LLNs) [187], might require low power consumption or high-robustness instead. Additionally, network topologies also vary depending on the specific use case. Optimized routing in data center networks might drastically differ from what it is expected in large service-provider networks, which could even follow business-based directives. Finally, networks are dynamic and change (not only because of updates, but also because of failures) and this should be taken into account as a factor as well.

All of these ideas are just a few considering the physical media aspect, but many more could be evoked considering other aspects, like types of communication (unicast, multicast, broadcast), or applications. This is particularly relevant for 5G networks and beyond [188] for example, in which new types of requirements and applications are still flourishing.

Nevertheless, after our analysis of the state of the art, we found out that most research works simply consider a very limited subset of networks: wired, unicast, and considering latency and throughput as main drivers. Only a few mention specifically the application to data center or wireless scenarios. For that reason, we devise the following research directions:

- Efforts should be made to apply ML in routing in wireless scenarios and, particularly, constrained scenarios.
- Broadcast and multicast optimal routing would be very valuable to assess.
- Traffic patterns, topologies and network changes should be considered in future analysis.
- Additional metrics should be evaluated as part of optimal routing, such as: node energy consumption, resilience or business-based metrics.

Overall goal: A ML-based routing algorithm for SDN should be customizable based on a diverse set of parameters (latency, throughput, CPU usage, energy-efficiency), media (wired and wireless), types of communication (unicast, multicast, broadcast), applications (traffic patterns) and topologies (DCNs, IoT, etc.). Additionally, apart from typical performance evaluations, proposals should also encompass long-term and multidisciplinary objectives, such as sustainability, hence tackling challenges envisioned by the Sustainable Development Goals (SDGs). If not feasible, the authors should at least justify the use case scenario and the evaluation method, to be consistent.

B. SECURITY AS A CROSS-CUTTING FEATURE

Possibly related with the previous aspect, security is an orthogonal aspect in networking [189], which affects all types of scenarios and should also be evaluated as part of any type of optimal routing. As many works already exist that apply ML and SDN for network intrusion detection, we would like to particularly focus on two aspects: data acquisition

and routing policy population. In particular, we envision the following research directions:

- ML-based proposals should consider the possibility that data acquisition could be hampered or modified to obtain faulty results, hence either a secure mechanism should be defined or a ML-based method to filter these attacks should be part of the overall designed ML method.
- Similarly to data acquisition, installation of routing entries could be affected as well by security attacks and this should be alleviated or, at least, proven to be safer than traditional and/or distributed approaches.

Overall goal: Security should be assessed as a cross-cutting parameter when evaluating the application of ML in SDN environments. The definition of an overall secure ML framework for SDN would be extremely valuable for the whole research community.

C. ARCHITECTURAL APPROACHES AND DATA MODELING

Though the classic definition of SDN presents a logically centralized architecture, it is not the only architectural approach to follow when applying ML-based approaches and, more importantly, it could even be not the most beneficial either. Researchers aiming at the application of AI and ML in SDN and, more generally, in programmable networks, should consider alternative architectural approaches like hybrid SDN (either vertically or horizontally [42]) or in-band SDN communication [190], as they could enhance and optimize the behavior of their proposals, including the monitoring side and data acquisition, or the potential security breaches that might be more severe in strictly centralized environments. To achieve this initiative, researchers could still leverage Mininet, but using BOFUSS switches [191] instead of (by-default) Open vSwitches, as the former can be easily modified. Alternatively, technologies like P4 [192] and XDP [193] have already demonstrated enhanced network programmability capabilities [149].

Additionally, alternative architectures could also provide deeper knowledge-based environments related with data modeling. So far, most data is directly obtained from the network, like CPU usage, packets received and sent, etc. Nevertheless, instead of this type of raw data, ML could profit from the use of advanced and high-level architectures like ontology-based [194] or even described by data bases [195], in which data is collected, merged and could provide an enhanced vision of the network. While it is true that these SDN architectures are more immature, some thoughts about potential applications with ML could be worth it.

Accordingly, the related research directions are the following:

- Proposals of ML-based SDN frameworks should consider the possibility of following non-centralized architectures, hence analyzing its benefits in comparison with centralized architectures. The simplest approach

would be redesigning one existing framework into a non-centralized scheme.

- Although more incipient, it would be nice to assess to what extent ML can benefit from using high-level data models.

Overall goal: To evaluate the advantages (security, scalability, etc.), and even disadvantages, of using non-centralized SDN architectures in ML-based frameworks.

D. IN THE NEED OF OPEN DATASETS AND IMPLEMENTATIONS

The need of open datasets and implementations is probably the most important of the five types of research directions. Although solutions based on ML for networking are growing more rapidly everyday, these frameworks not only rely on the specific developed code, but they also need input data to train and/or test their models. Such data is scarce and barely shared [196]. Most times, this is because the collection of network data involves individual privacy issues [166]. Although this could initially have a high cost (for the first researchers following this idea), it would benefit the whole community tremendously in the long term, because it would permit other to reproduce, compare and enhance the existing solutions, hence increasing their impact. Recent initiatives are appearing in this regard, like the Softwarized Network Data Zoo (SNDZoo) [197], which intends to start an open ecosystem for dataset collections in the networking domain, based on a specific methodology to achieve homogeneous collection and publication.

Alternatively, open implementations is another, and probably easier, method to foster the merging efforts in the field. Whilst most surveyed works have used open platforms to implement their ideas (like the Ryu controller or the OM-NeT++ simulator), most of them omit publishing them in public repositories like GitHub, which is a simple and very effective way to promote the merging of efforts from different proposals and research groups.

In conclusion, we envision the next research directions:

- To build upon existing open data ecosystems like SNDZoo and define the requisites to make it grow faster.
- To evaluate what is the most beneficial method for implementation replication, i.e., what open platforms and tools should be prioritized for later publication and reutilization.
- To develop some type of framework or community to compete based on specific AI & ML *challenges* based on homogeneous datasets and topologies, which would foster evolution and replication of results.

Overall goal: To foster open datasets and implementations to achieve more valuable results and ideas for the research community. At least, all frameworks should have a public link to their implementations.

E. INTO THE FOG

As previously mentioned, the current evolution of networks is every day more focused on the edge of networks, where

IoT devices –and users– reside. This clear trend [17], [68] is moving step by step the intelligence of the network far from the core, towards what is called *edge computing*, *fog computing* and, even, *mist computing* [198]. When checking these names anybody can clearly visualize that the future of the ML approaches should be based on federated approaches, as the ones referenced before [75], [76]. However, these paradigms are still incipient and many challenges still need to be tackled. An example of these challenges are LLNs (previously mentioned), in which nodes are constrained in memory and battery and, therefore, routing is –per se– a challenge for them. This type of networks would benefit from this architectural approach as stand-alone devices cannot cope with the whole computational requirements of a centralized ML approach.

In particular, we envision the next research directions:

- To determine the minimum computational requirements of network nodes to act as federated ML nodes.
- To define a negotiation and/or communication framework to allow efficient, secure and scalable communication among nodes.
- To align the previous two points with specific SDN and NFV architectural concepts and technologies (e.g. leverage SDN in-band communication for federated ML approaches).

Overall goal: While this survey focuses on ML for its application to networking, some research efforts should be directed to networking for ML too, as they are both complementary.

F. TOWARDS INDUSTRY-BASED PRACTICAL SCENARIOS

Finally, we would like to mention an objective directly related with the previous ones: working on implementations close to industry-based practical scenarios. Now that most network innovation in companies is based on open source software, we, as part of the research community, should profit from it and leverage the same platforms and tools for a more effective adoption by industry. Alternatively, merging efforts with other big projects like Pronto [199], [200] would be clearly beneficial. Additionally, considering the application of ML in routing is usually foreseen as a step towards automatized network management, we should continuously monitor to what extent is ML trusted by network operators. Moving from a traditional (almost manual) management to another based on ML might imply severe changes and even unexpected outcomes. Therefore, the benefits of applying ML in these environments should be proven and clear or, otherwise, the potential impact might be too low.

In summary, some research directions could be the following:

- To implement scenarios based on the ONOS controller, which is the one most supported by the ONF and industry. Alternatively, OpenDaylight could also be a good choice.

- To create a communication channel with industry to check their needs and propose initiatives, which could also be feasible via the ONF (they provide the mechanisms to do so).

Overall goal: Implementation and evaluations should be as close to real scenarios as possible for effective adoption by industry. To this purpose, using platforms leveraged for commercial solutions (like ONOS) and communicating with standardization bodies (ONF) is pivotal.

VII. CONCLUSION

In this paper we surveyed the use of ML in SDN for routing optimization, classified into three types (SL, UL and RL), which are first introduced and defined, together with some of the associated techniques. According to our analysis, during the last three years, the works using ML for routing optimization in SDN have rapidly flourished, and particularly those leveraging DRL. Nevertheless, most research works are based on simple prototypes and for very specific network scenarios (wired, centralized SDN, and compared to distributed routing algorithms based on latency and throughput) and are hard to reproduce and compare. Thus, their evaluations are not completely meaningful and conclusive. We believe a sustained effort is needed to create an open ecosystem in which the different works support each other, instead of being proposed independently. Otherwise, most research efforts might never be implemented in practice. To this purpose, we finalize the survey with six sections including specific research directions for this field.

REFERENCES

- [1] S. Salsano, P. L. Ventre, L. Prete, G. Siracusano, M. Gerola, and E. Salvadori, "OSHI-Open Source Hybrid IP/SDN networking (and its emulation on Mininet and on distributed SDN testbeds)," in *2014 Third European Workshop on Software Defined Networks*. IEEE, 2014, pp. 13–18.
- [2] F. Alam, I. Katib, and A. S. Alzahrani, "New networking era: software defined networking," *Int J Adv Res Comput Sci Softw Eng. Computer Science Department, Faculty of Computing & IT Kind Abdulaziz University Jeddah, Saudi Arabia*, 2013.
- [3] F. Bannour, S. Souihi, and A. Mellouk, "Distributed SDN control: Survey, taxonomy, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 333–354, 2017.
- [4] ETSI, "Network Functions Virtualisation (NFV)," 2020.
- [5] A. A. Antonov, "From artificial intelligence to human super-intelligence [J]," *Artificial Intelligence*, vol. 2, no. 6, p. 3560, 2011.
- [6] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 3–10.
- [7] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, "Machine Learning in Software Defined Networks: Data collection and traffic classification," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, 2016, pp. 1–5.
- [8] T. V. Phan, S. T. Islam, T. G. Nguyen, and T. Bauschert, "Q-DATA: Enhanced Traffic Flow Monitoring in Software-Defined Networks applying Q-learning," in *2019 15th International Conference on Network and Service Management (CNSM)*, 2019, pp. 1–9.
- [9] S. I. Kim and H. S. Kim, "Dynamic Service Function Chaining by Resource Usage Learning in SDN/NFV Environment," in *2019 International Conference on Information Networking (ICOIN)*, 2019, pp. 485–488.
- [10] J. Xu, J. Wang, Q. Qi, H. Sun, and B. He, "IARA: An Intelligent Application-Aware VNF for Network Resource Allocation with Deep Learning," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018, pp. 1–3.
- [11] Q. Schueller, K. Basu, M. Younas, M. Patel, and F. Ball, "A Hierarchical Intrusion Detection System using Support Vector Machine for SDN Network in Cloud Data Center," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–6.
- [12] P. Somwang and W. Lilakiatsakun, "Computer network security based on Support Vector Machine approach," in *2011 11th International Conference on Control, Automation and Systems*, 2011, pp. 155–160.
- [13] G. Kaur and P. Gupta, "Hybrid Approach for detecting DDOS Attacks in Software Defined Networks," in *2019 Twelfth International Conference on Contemporary Computing (IC3)*, 2019, pp. 1–6.
- [14] A. Prakash and R. Priyadarshini, "An Intelligent Software defined Network Controller for preventing Distributed Denial of Service Attack," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 585–589.
- [15] R. Mijumbi, J. Serrat, J. Rubio-Loyola, N. Bouten, F. De Turck, and S. Latré, "Dynamic resource management in SDN-based virtualized networks," in *10th international conference on network and service management (CNSM) and workshop*. IEEE, 2014, pp. 412–417.
- [16] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A Roadmap for Traffic Engineering in SDN-OpenFlow Networks," *Comput. Netw.*, vol. 71, p. 1–30, Oct. 2014. [Online]. Available: <https://doi.org/10.1016/j.comnet.2014.06.002>
- [17] A. Mourad, R. Yang, P. H. Lehne, and A. de la Oliva, "Towards 6G: Evolution of Key Performance Indicators and Technology Trends," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5.
- [18] A. K. Singh and S. Srivastava, "A survey and classification of controller placement problem in SDN," *International Journal of Network Management*, vol. 28, no. 3, p. e2018, 2018.
- [19] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019.
- [20] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.
- [21] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.
- [22] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.
- [23] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, "A Survey on the Contributions of Software-Defined Networking to Traffic Engineering," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 918–953, 2017.
- [24] M. Karakus and A. Dursesi, "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)," *Computer Networks*, vol. 112, pp. 279–293, 2017.
- [25] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.
- [26] A. Binsahaq, T. R. Sheltami, and K. Salah, "A Survey on Autonomic Provisioning and Management of QoS in SDN Networks," *IEEE Access*, vol. 7, pp. 73 384–73 435, 2019.
- [27] R. Etengu, S. C. Tan, L. C. Kwang, F. M. Abbou, and T. C. Chuah, "AI-Assisted Framework for Green-Routing and Load Balancing in Hybrid Software-Defined Networking: Proposal, Challenges and Future Perspective," *IEEE Access*, vol. 8, pp. 166 384–166 441, 2020.
- [28] Y. Qian, J. Wu, R. Wang, F. Zhu, and W. Zhang, "Survey on Reinforcement Learning Applications in Communication Networks," *Journal of Communications and Information Networks*, vol. 4, no. 2, pp. 30–39, 2019.
- [29] Z. Mammeri, "Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches," *IEEE Access*, vol. 7, pp. 55 916–55 950, 2019.
- [30] S. Jamshidi, "The Applications of Machine Learning Techniques in Networking," 2019.

- [31] Y. Zhang, J. Xin, X. Li, and S. Huang, "Overview on routing and resource allocation based machine learning in optical networks," *Optical Fiber Technology*, vol. 60, p. 102355, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S106852002030345X>
- [32] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 16, 2018.
- [33] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393–430, 2018.
- [34] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A Survey of Networking Applications Applying the Software Defined Networking Concept Based on Machine Learning," *IEEE Access*, vol. 7, pp. 95 397–95 417, 2019.
- [35] H.-N. Quach, S. Yoem, and K. Kim, "Survey on Reinforcement Learning based Efficient Routing in SDN," in *The 9th International Conference on Smart Media and Applications (SMA 2020)*, 2020.
- [36] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Computer Networks*, vol. 81, pp. 79–95, 2015.
- [37] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 623–654, 2015.
- [38] O. S. Al-Heety, Z. Zakaria, M. Ismail, M. M. Shakir, S. Alani, and H. Alsariera, "A Comprehensive Survey: Benefits, Services, Recent Works, Challenges, Security, and Use Cases for SDN-VANET," *IEEE Access*, vol. 8, pp. 91 028–91 047, 2020.
- [39] M. D. Hatagundi and H. Kumaraswamy, "A Comprehensive Survey on Different Attacks on SDN and Approaches to Mitigate," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2019, pp. 624–627.
- [40] J. C. C. Chica, J. C. Imbachi, and J. F. Botero, "Security in SDN: A comprehensive survey," *Journal of Network and Computer Applications*, p. 102595, 2020.
- [41] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, p. 69–74, Mar. 2008. [Online]. Available: <https://doi.org/10.1145/1355734.1355746>
- [42] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: A survey of existing approaches," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3259–3306, 2018.
- [43] N. S. Pawar, A. Arunvel, G. N. Kumar, and A. K. Sinha, "Securing network using software-defined networking in control and data planes," in *Computing and Network Sustainability*. Springer, 2019, pp. 433–443.
- [44] Y.-D. Lin, P.-C. Lin, C.-H. Yeh, Y.-C. Wang, and Y.-C. Lai, "An extended SDN architecture for network function virtualization with a case study on intrusion prevention," *IEEE Network*, vol. 29, no. 3, pp. 48–53, 2015.
- [45] P. L. Consortium, "P4 Language and Related Specifications," 2020. [Online]. Available: <https://p4.org/specs/>
- [46] E. Rojas, R. Doriguzzi-Corin, S. Tamurejo, A. Beato, A. Schwabe, K. Phemius, and C. Guerrero, "Are We Ready to Drive Software-Defined Networks? A Comprehensive Survey on Management Tools and Techniques," *ACM Comput. Surv.*, vol. 51, no. 2, Feb. 2018. [Online]. Available: <https://doi.org/10.1145/3165290>
- [47] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu et al., "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.
- [48] B. Davie, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. Gude, A. Padmanabhan, T. Petty, K. Duda, and A. Chanda, "A database approach to sdn control plane design," *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 1, pp. 15–26, 2017.
- [49] M. Filer, J. Gaudette, M. Ghobadi, R. Mahajan, T. Issenhuth, B. Klinkers, and J. Cox, "Elastic optical networking in the Microsoft cloud," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, no. 7, pp. A45–A54, 2016.
- [50] T. Y. Yang, A. Dehghantanha, K.-K. R. Choo, and Z. Muda, "Windows instant messaging app forensics: Facebook and Skype as case studies," *PloS one*, vol. 11, no. 3, 2016.
- [51] J. Ungerman, "SDN v praxi," Cisco Connect, Tech. Rep., Mar. 2015, last accessed April 19, 2018. [Online]. Available: https://www.cisco.com/c/dam/assets/global/CZ/events/2015-ciscoconnect/pdf/TECH-SP-1-SDN_v_praxi-Ungerman.pdf
- [52] D. Zheng, "Huawei Enterprise Business: Four-Dimensional SDN Deployment," Huawei Techn. Co. Ltd., Tech. Rep., Jul. 2013, last accessed April 19, 2018. [Online]. Available: http://e.huawei.com/en/publications/global/ict_insights/hw_314355/-feature%20story/HW_311109
- [53] NEC White Paper, "SDN Component Stack and Hybrid Introduction Models," NEC, Tech. Rep., 2014, last accessed April 19, 2018. [Online]. Available: <https://www.necam.com/docs/?id=c2e5a040-cdf1-4fd7-b63e-6eea4b1f7a7b>
- [54] Verizon Network Infrastructure Planning, "SDN-NFV Reference Architecture, Version 1.0," Verizon, Tech. Rep., Feb. 2016, last accessed April 19th, 2018. [Online]. Available: http://innovation.verizon.com/content/dam/vic/PDF/Verizon_SDN-NFV_Reference_Architecture.pdf
- [55] Hewlett-Packard, "HP Technical white paper: HP SDN hybrid network architecture: Scalable, low-risk network deployments using hybrid SDN," HP, Tech. Rep., Apr. 2015, last accessed April 19, 2018. [Online]. Available: <http://arubanetworks.com/aruba/attachments/aruba/SDN/43/1/4AA5-6738ENW.PDF>
- [56] R. Honnachari, "Understanding and Embracing SDN and NFV-Based Network Solutions to Drive Operational Efficiency—An Executive Brief Sponsored by AT&T," Frost & Sullivan, Tech. Rep., Aug. 2015, last accessed April 19, 2018. [Online]. Available: <https://www.business.att.com/content/whitepaper/gc/frost-and-sullivan-nod-sdn-nfv-whitepaper.pdf?grantAccess>
- [57] J. W. Guck, A. Van Bemt, M. Reisslein, and W. Kellerer, "Unicast QoS Routing Algorithms for SDN: A Comprehensive Survey and Performance Evaluation," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 388–415, 2018.
- [58] D. Lopez-Pajares, E. Rojass, J. A. Carral, I. Martinez-Yelmo, and J. Alvarez-Horcajo, "The Disjoint Multipath Challenge: Multiple Disjoint Paths Guaranteeing Scalability," *IEEE Access*, vol. 9, pp. 74 422–74 436, 2021.
- [59] A. Shirmarz and A. Ghaffari, "Performance issues and solutions in SDN-based data center: a survey," *The Journal of Supercomputing*, pp. 1–49, 2020.
- [60] S. Badotra and S. N. Panda, "Experimental comparison and evaluation of various OpenFlow software defined networking controllers," *International Journal of Applied Science and Engineering*, vol. 17, pp. 317–324, diciembre 2020. [Online]. Available: [https://doi.org/10.6703/IJASE.202012_17\(4\).317](https://doi.org/10.6703/IJASE.202012_17(4).317)
- [61] F. Tomonori, "Introduction to Ryu SDN framework," *Open Networking Summit*, pp. 1–14, 2013.
- [62] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an Open, Distributed SDN OS," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1–6. [Online]. Available: <https://doi.org/10.1145/2620728.2620744>
- [63] P. Casas, "Two Decades of AI4NETS - AI/ML for Data Networks: Challenges Research Directions," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1–6.
- [64] A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, E. Alarcón, M. Solé, V. Muntés-Mulero, D. Meyer, S. Barkai, M. J. Hobbett, G. Estrada, K. Ma'ruf, F. Coras, V. Ermagan, H. Latapie, C. Cassar, J. Evans, F. Maino, J. Walrand, and A. Cabellos, "Knowledge-Defined Networking," *SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, p. 2–10, Sep. 2017. [Online]. Available: <https://doi.org/10.1145/3138808.3138810>
- [65] W. Kellerer, P. Kalmbach, A. Blenk, A. Basta, M. Reisslein, and S. Schmid, "Adaptable and Data-Driven Softwarized Networks: Review, Opportunities, and Challenges," *Proceedings of the IEEE*, vol. 107, no. 4, pp. 711–731, 2019.
- [66] P. Kalmbach, J. Zerwas, P. Babarczy, A. Blenk, W. Kellerer, and S. Schmid, "Empowering Self-Driving Networks," in *Proceedings of the Afternoon Workshop on Self-Driving Networks*, ser. SelfDN 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 8–14. [Online]. Available: <https://doi.org/10.1145/3229584.3229587>
- [67] J. Zerwas, P. Kalmbach, L. Henkel, G. Rétvári, W. Kellerer, A. Blenk, and S. Schmid, "NetBOA: Self-Driving Network Benchmarking,"

- in *Proceedings of the 2019 Workshop on Network Meets AI and ML*, ser. NetAI'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 8–14. [Online]. Available: <https://doi.org/10.1145/3341216.3342207>
- [68] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, “The Roadmap to 6G: AI Empowered Wireless Networks,” *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.
- [69] K. Sheth, K. Patel, H. Shah, S. Tanwar, R. Gupta, and N. Kumar, “A taxonomy of AI techniques for 6G communication networks,” *Computer Communications*, vol. 161, pp. 279–303, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140366420318478>
- [70] J. Gedeon, F. Brandherm, R. Egert, T. Grube, and M. Mühlhäuser, “What the Fog? Edge Computing Revisited: Promises, Applications and Future Challenges,” *IEEE Access*, vol. 7, pp. 152 847–152 878, 2019.
- [71] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.
- [72] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [73] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng, “Text detection and character recognition in scene images with unsupervised feature learning,” in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 440–445.
- [74] L. Deng and X. Li, “Machine learning paradigms for speech recognition: An overview,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 1060–1089, 2013.
- [75] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.
- [76] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, “A joint learning and communications framework for federated learning over wireless networks,” *IEEE Transactions on Wireless Communications*, 2020.
- [77] A. K. Jain, J. Mao, and K. M. Mohiuddin, “Artificial neural networks: A tutorial,” *Computer*, vol. 29, no. 3, pp. 31–44, 1996.
- [78] B. A. Pearlmutter, “Learning state space trajectories in recurrent neural networks,” *Neural Computation*, vol. 1, no. 2, pp. 263–269, 1989.
- [79] M. F. Alghifari, T. S. Gunawan, and M. Kartiwi, “Speech emotion recognition using deep feedforward neural network,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 2, pp. 554–561, 2018.
- [80] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [81] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [82] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [83] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [84] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [85] S. Timotheou, “The random neural network: a survey,” *The computer journal*, vol. 53, no. 3, pp. 251–267, 2010.
- [86] D. Chen and K. S. Trivedi, “Optimization for condition-based maintenance with semi-Markov decision process,” *Reliability engineering & system safety*, vol. 90, no. 1, pp. 25–29, 2005.
- [87] M. B. Ferraro, R. Coppi, G. G. Rodríguez, and A. Colubi, “A linear regression model for imprecise response,” *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 759–770, 2010.
- [88] W. Szeto, R. Wong, and W. Yang, “Guiding vacant taxi drivers to demand locations by taxi-calling signals: A sequential binary logistic regression modeling approach and policy implications,” *Transport Policy*, vol. 76, pp. 100–110, 2019.
- [89] R. Islam and M. A. Shahjalal, “Late breaking results: Predicting DRC violations using ensemble random forest algorithm,” in *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–2.
- [90] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [91] M. Lehsaini and M. B. Benmahdi, “An improved k-means cluster-based routing scheme for wireless sensor networks,” in *2018 International Symposium on Programming and Systems (ISPS)*. IEEE, 2018, pp. 1–6.
- [92] S. Patel, S. Sihmar, and A. Jatain, “A study of hierarchical clustering algorithms,” in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2015, pp. 537–541.
- [93] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [94] D. Reynolds, *Gaussian Mixture Models*. Boston, MA: Springer US, 2009, pp. 659–663. [Online]. Available: https://doi.org/10.1007/978-0-387-73003-5_196
- [95] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [96] E. O. Neftci and B. B. Averbeck, “Reinforcement learning in artificial and biological systems,” *Nature Machine Intelligence*, vol. 1, no. 3, pp. 133–143, 2019.
- [97] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.
- [98] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [99] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, “Rainbow: Combining improvements in deep reinforcement learning,” *arXiv preprint arXiv:1710.02298*, 2017.
- [100] B. Jang, M. Kim, G. Harerimana, and J. W. Kim, “Q-Learning Algorithms: A Comprehensive Classification and Applications,” *IEEE Access*, vol. 7, pp. 133 653–133 667, 2019.
- [101] H. V. Hasselt, “Double Q-learning,” in *Advances in neural information processing systems*, 2010, pp. 2613–2621.
- [102] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering Cambridge, UK, 1994, vol. 37.
- [103] M. Nazari, A. Oroojlooy, L. V. Snyder, and M. Takác, “Deep reinforcement learning for solving the vehicle routing problem,” *arXiv preprint arXiv:1802.04240*, 2018.
- [104] Y. Zhang, J. Yao, and H. Guan, “Intelligent cloud resource management with deep reinforcement learning,” *IEEE Cloud Computing*, vol. 4, no. 6, pp. 60–69, 2017.
- [105] B. Bakker, “Reinforcement learning with long short-term memory,” in *Advances in neural information processing systems*, 2002, pp. 1475–1482.
- [106] A. Azzouni, R. Boutaba, and G. Pujolle, “Neuroute: Predictive dynamic routing for software-defined networks,” in *2017 13th International Conference on Network and Service Management (CNSM)*. IEEE, 2017, pp. 1–6.
- [107] C. Chen-Xiao and X. Ya-Bin, “Research on load balance method in SDN,” *International Journal of Grid and Distributed Computing*, vol. 9, no. 1, pp. 25–36, 2016.
- [108] C. Hardegen and S. Rieger, “Prediction-based Flow Routing in Programmable Networks with P4,” in *2020 16th International Conference on Network and Service Management (CNSM)*, 2020, pp. 1–5.
- [109] S. Troia, A. Rodriguez, I. Martín, J. A. Hernández, O. G. De Dios, R. Alvizu, F. Musumeci, and G. Maier, “Machine-Learning-Assisted Routing in SDN-based Optical Networks,” in *2018 European Conference on Optical Communication (ECOC)*. IEEE, 2018, pp. 1–3.
- [110] L. Wang and D. T. Delaney, “QoE Oriented Cognitive Network Based on Machine Learning and SDN,” in *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*. IEEE, 2019, pp. 678–681.
- [111] K. K. Budhraj, A. Malvankar, M. Bahrami, C. Kundu, A. Kundu, and M. Singhal, “Risk-based packet routing for privacy and compliance-preserving SDN,” in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. IEEE, 2017, pp. 761–765.
- [112] S. Kumar, G. Bansal, and V. S. Shekhawat, “A machine learning approach for traffic flow provisioning in software defined networks,” in *2020 International Conference on Information Networking (ICOIN)*. IEEE, 2020, pp. 602–607.
- [113] F. Francois and E. Gelenbe, “Optimizing secure SDN-enabled inter-data centre overlay networks through cognitive routing,” in *2016 IEEE*

- 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, 2016, pp. 283–288.
- [114] P. Sun, J. Li, J. Lan, Y. Hu, and X. Lu, “RNN Deep Reinforcement Learning for Routing Optimization,” in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*. IEEE, 2018, pp. 285–289.
- [115] Z. Tu, H. Zhou, K. Li, G. Li, and Q. Shen, “A Routing Optimization Method for Software-Defined SGIN Based on Deep Reinforcement Learning,” in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.
- [116] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, “QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach,” in *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 2016, pp. 25–33.
- [117] C. Fang, C. Cheng, Z. Tang, and C. Li, “Research on Routing Algorithm Based on Reinforcement Learning in SDN,” in *Journal of Physics: Conference Series*, vol. 1284, no. 1. IOP Publishing, 2019, p. 012053.
- [118] T. Phan, S. Feld, and C. Linnhoff-Popien, “Artificial Intelligence—the new Revolutionary Evolution,” 2020.
- [119] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, “Machine learning in wireless sensor networks: Algorithms, strategies, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.
- [120] A. Forster, “Machine learning techniques applied to wireless ad-hoc networks: Guide and survey,” in *2007 3rd international conference on intelligent sensors, sensor networks and information*. IEEE, 2007, pp. 365–370.
- [121] C. Fiandrino, C. Zhang, P. Patras, A. Banchs, and J. Widmer, “A Machine Learning-based Framework for Optimizing the Operation of Future Networks,” *IEEE Communications Magazine*, 2020.
- [122] A. Sabeeh, Y. Al-Dunainawi, M. F. Abbod, and H. Al-Rawashidy, “A hybrid intelligent approach for optimising software-defined networks performance,” in *2016 6th International Conference on Information Communication and Management (ICIMC)*. IEEE, 2016, pp. 47–51.
- [123] Y.-J. Wu, P.-C. Hwang, W.-S. Hwang, and M.-H. Cheng, “Artificial Intelligence Enabled Routing in Software Defined Networking,” *Applied Sciences*, vol. 10, no. 18, p. 6564, Sep 2020. [Online]. Available: <http://dx.doi.org/10.3390/app10186564>
- [124] A. Azzouni and G. Pujolle, “NeuTM: A neural network-based framework for traffic matrix prediction in SDN,” in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–5.
- [125] F. Benamrane, M. Ali, D. K. Luong, Y. Hu, J. Li, and K. Abdo, “Bandwidth Management in Avionic Networks based on SDN Paradigm and ML Techniques,” in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, pp. 1–9.
- [126] K. Rusek, J. Suárez-Varela, A. Mestres, P. Barlet-Ros, and A. Cabellos-Aparicio, “Unveiling the Potential of Graph Neural Networks for Network Modeling and Optimization in SDN,” in *Proceedings of the 2019 ACM Symposium on SDN Research*, ser. SOSR ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 140–151. [Online]. Available: <https://doi.org/10.1145/3314148.3314357>
- [127] K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, “RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2260–2270, 2020.
- [128] W. Sun, Z. Wang, and G. Zhang, “A QoS-guaranteed intelligent routing mechanism in software-defined networks,” *Computer Networks*, vol. 185, p. 107709, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128620313050>
- [129] G. Choudhury, D. Lynch, G. Thakur, and S. Tse, “Two use cases of machine learning for SDN-enabled IP/optical networks: traffic matrix prediction and optical path performance prediction,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D52–D62, 2018.
- [130] L. EL-Garoui, S. Pierre, and S. Chamberland, “A New SDN-Based Routing Protocol for Improving Delay in Smart City Environments,” *Smart Cities*, vol. 3, no. 3, pp. 1004–1021, Sep 2020. [Online]. Available: <http://dx.doi.org/10.3390/smartcities3030050>
- [131] M. K. Awad, M. H. H. Ahmed, A. F. Almutairi, and I. Ahmad, “Machine learning-based multipath routing for software defined networks,” *Journal of Network and Systems Management*, vol. 29, no. 2, pp. 1–30, 2021.
- [132] A. Akbar, M. Ibrar, M. A. Jan, A. K. Bashir, and L. Wang, “SDN-Enabled Adaptive and Reliable Communication in IoT-Fog Environment Using Machine Learning and Multiobjective Optimization,” *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3057–3065, 2021.
- [133] A. I. Owusu and A. Nayak, “An Intelligent Traffic Classification in SDN-IoT: A Machine Learning Approach,” in *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2020, pp. 1–6.
- [134] A. Sacco, F. Esposito, and G. Marchetto, “RoPE: An Architecture for Adaptive Data-Driven Routing Prediction at the Edge,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 986–999, 2020.
- [135] D. Todorov, H. Valchanov, and V. Aleksieva, “Load Balancing model based on Machine Learning and Segment Routing in SDN,” in *2020 International Conference Automatics and Informatics (ICAI)*, 2020, pp. 1–4.
- [136] B. Man and C. Li, “Routing control method in software defined networking and openflow controller,” Mar. 19 2019, uS Patent 10,237,181.
- [137] K. Perera, U. Gunaratne, B. Chaturanga, C. Ramanayake, and A. Pasqual, “Hybrid software defined networking controller,” in *DCNET*, 2017, pp. 77–84.
- [138] D. Pamucar and G. Ćirović, “Vehicle route selection with an adaptive neuro fuzzy inference system in uncertainty conditions,” *Decision Making: Applications in Management and Engineering*, vol. 1, no. 1, pp. 13–37, 2018.
- [139] A. Hiassat, A. Diabat, and I. Rahwan, “A genetic algorithm approach for location-inventory-routing problem with perishable products,” *Journal of manufacturing systems*, vol. 42, pp. 93–103, 2017.
- [140] B. Yao, B. Yu, P. Hu, J. Gao, and M. Zhang, “An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot,” *Annals of Operations Research*, vol. 242, no. 2, pp. 303–320, 2016.
- [141] A. Azzouni and G. Pujolle, “A long short-term memory recurrent neural network framework for network traffic matrix prediction,” *arXiv preprint arXiv:1705.05690*, 2017.
- [142] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O’Connor, P. Radoslavov, W. Snow, and G. Parulkar, “Onos: Towards an open, distributed sdn os,” in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1–6. [Online]. Available: <https://doi.org/10.1145/2620728.2620744>
- [143] L. Guillen, S. Izumi, T. Abe, and T. Suganuma, “SAND/3: SDN-Assisted Novel QoE Control Method for Dynamic Adaptive Streaming over HTTP/3,” *Electronics*, vol. 8, no. 8, p. 864, 2019.
- [144] A. Rajagopal and M. Balmakhtar, “Data service policy control based on software defined network (SDN) key performance indicators (KPIs),” Oct. 9 2018, uS Patent 10,097,421.
- [145] W.-K. Jia, X. Dong, Y.-C. Chen, and F. Chen, “A Survey on All-Optical IP Convergence Optical Transport Networks,” in *2019 7th International Conference on Information, Communication and Networks (ICIN)*. IEEE, 2019, pp. 114–119.
- [146] J. Kundrát, O. Havlíš, J. Jedlinský, and J. Vojtěch, “Opening up roadms: Let us build a disaggregated open optical line system,” *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4041–4051, 2019.
- [147] G. Leduc, H. Abrahamsson, S. Balon, S. Bessler, M. D’Arienzo, O. Delcourt, J. Domingo-Pascual, S. Cerav-Erbas, I. Gojmerac, X. Masip, A. Pescapè, B. Quoitin, S. Romano, E. Salvadori, F. Skivée, H. Tran, S. Uhlig, and H. Ümit, “An open source traffic engineering toolbox,” *Computer Communications*, vol. 29, no. 5, pp. 593–610, 2006, networks of Excellence. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140366405002124>
- [148] I. Tomkos, D. Klonidis, E. Pikasis, and S. Theodoridis, “Toward the 6G Network Era: Opportunities and Challenges,” *IT Professional*, vol. 22, no. 1, pp. 34–38, 2020.
- [149] D. Carrascal, E. Rojas, J. Alvarez-Horcajo, D. Lopez-Pajares, and I. Martínez-Yelmo, “Analysis of P4 and XDP for IoT Programmability in 6G and Beyond,” *IoT*, vol. 1, no. 2, pp. 605–622, 2020.
- [150] J. Rischke, P. Sossalla, H. Salah, F. H. P. Fitzek, and M. Reisslein, “QR-SDN: Towards Reinforcement Learning States, Actions, and Rewards for Direct Flow Routing in Software-Defined Networks,” *IEEE Access*, vol. 8, pp. 174 773–174 791, 2020.
- [151] D. M. Casas-Velasco, O. M. C. Rendon, and N. L. S. da Fonseca, “Intelligent Routing based on Reinforcement Learning for Software-Defined

- Networking,” *IEEE Transactions on Network and Service Management*, pp. 1–1, 2020.
- [152] S. Sendra, A. Rego, J. Lloret, J. M. Jimenez, and O. Romero, “Including artificial intelligence in a routing protocol using Software Defined Networks,” in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017, pp. 670–674.
- [153] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, “A machine learning approach to routing,” *arXiv preprint arXiv:1708.03074*, 2017.
- [154] S. Hassas Yeganeh and Y. Ganjali, “Kandoo: a framework for efficient and scalable offloading of control applications,” in *Proceedings of the first workshop on Hot topics in software defined networks*, 2012, pp. 19–24.
- [155] J. McCauley, A. Panda, M. Casado, T. Koponen, and S. Shenker, “Extending SDN to large-scale networks,” *Open Networking Summit*, pp. 1–2, 2013.
- [156] F. Francois and E. Gelenbe, “Towards a cognitive routing engine for software defined networks,” in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.
- [157] P. Sun, Y. Hu, J. Lan, L. Tian, and M. Chen, “TIDE: Time-relevant deep reinforcement learning for routing optimization,” *Future Generation Computer Systems*, vol. 99, pp. 401–409, 2019.
- [158] P. Sun, J. Li, Z. Guo, Y. Xu, J. Lan, and Y. Hu, “SINET: Enabling Scalable Network Routing with Deep Reinforcement Learning on Partial Nodes,” ser. SIGCOMM Posters and Demos '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 88–89. [Online]. Available: <https://doi.org/10.1145/3342280.3342317>
- [159] P. Sun, J. Lan, Z. Guo, Y. Xu, and Y. Hu, “Improving the Scalability of Deep Reinforcement Learning-Based Routing with Control on Partial Nodes,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3557–3561.
- [160] P. Sun, Z. Guo, J. Lan, J. Li, Y. Hu, and T. Baker, “ScaleDRL: A Scalable Deep Reinforcement Learning Approach for Traffic Engineering in SDN with Pinning Control,” *Computer Networks*, vol. 190, p. 107891, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128621000554>
- [161] G. Stampa, M. Arias, D. Sánchez-Charles, V. Muntés-Mulero, and A. Cabellos, “A deep-reinforcement learning approach for software-defined networking routing optimization,” *arXiv preprint arXiv:1709.07080*, 2017.
- [162] C. Yu, J. Lan, Z. Guo, and Y. Hu, “Drom: Optimizing the routing in software-defined networks with deep reinforcement learning,” *IEEE Access*, vol. 6, pp. 64 533–64 539, 2018.
- [163] D. N. Maheswari, C. Sujitha, and K. Ramana, “Routing optimization in SDN using deep reinforcement learning,” *Journal of Engineering, Computing and Architecture*, 2020.
- [164] C. Xu, W. Zhuang, and H. Zhang, “A Deep-Reinforcement Learning Approach for SDN Routing Optimization,” in *Proceedings of the 4th International Conference on Computer Science and Application Engineering*, ser. CSAE 2020. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: <https://doi.org/10.1145/3424978.3425004>
- [165] H. Yao, T. Mai, C. Jiang, L. Kuang, and S. Guo, “AI Routers & Network Mind: A Hybrid Machine Learning Paradigm for Packet Routing,” *IEEE Computational Intelligence Magazine*, vol. 14, no. 4, pp. 21–30, 2019.
- [166] Q. Zhang, X. Wang, J. Lv, and M. Huang, “Intelligent Content-Aware Traffic Engineering for SDN: An AI-Driven Approach,” *IEEE Network*, vol. 34, no. 3, pp. 186–193, 2020.
- [167] A. Nahar and D. Das, “SeScR: SDN-Enabled Spectral Clustering-Based Optimized Routing Using Deep Learning in VANET Environment,” in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, 2020, pp. 1–9.
- [168] P. T. A. Quang, Y. H. Aoul, and A. Outtagarts, “Deep Reinforcement Learning Based QoS-Aware Routing in Knowledge-Defined Networking,” in *Quality, Reliability, Security and Robustness in Heterogeneous Systems: 14th EAI International Conference, Qshine 2018, Ho Chi Minh City, Vietnam, December 3-4, 2018, Proceedings*, vol. 272. Springer, 2018, p. 14.
- [169] P. Swain, U. Kamalia, R. Bhandarkar, and T. Modi, “CoDRL: Intelligent Packet Routing in SDN Using Convolutional Deep Reinforcement Learning,” in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2019, pp. 1–6.
- [170] X. Lu, J. Chen, L. Lu, X. Huang, and X. Lu, “SDN Routing Optimization Based on Improved Reinforcement Learning,” in *Proceedings of the 2020 International Conference on Cyberspace Innovation of Advanced Technologies*, ser. CIAT 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 153–158. [Online]. Available: <https://doi.org/10.1145/3444370.3444563>
- [171] W. Liu, “Intelligent Routing based on Deep Reinforcement Learning in Software-Defined Data-Center Networks,” in *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–6.
- [172] W. xi Liu, J. Cai, Q. C. Chen, and Y. Wang, “DRL-R: Deep reinforcement learning approach for intelligent routing in software-defined data-center networks,” *Journal of Network and Computer Applications*, p. 102865, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804520303313>
- [173] Q. Fu, E. Sun, K. Meng, M. Li, and Y. Zhang, “DQR: Deep Q-Learning for Routing Schemes in SDN-Based Data Center Networks,” *IEEE Access*, vol. 8, pp. 103 491–103 499, 2020.
- [174] M. Chiesa, G. Kindler, and M. Schapira, “Traffic Engineering With Equal-Cost-MultiPath: An Algorithmic Perspective,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 779–792, 2017.
- [175] W. Seher and T. Charles Clancy, “Load balancing in data center networks with folded-Clos architectures,” in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 2015, pp. 1–6.
- [176] S. Q. Jalil, M. Husain Rehmani, and S. Chalup, “DQR: Deep Q-Routing in Software Defined Networks,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.
- [177] Y. R. Chen, A. Rezapour, W. G. Tzeng, and S. C. Tsai, “RL-Routing: An SDN Routing Algorithm Based on Deep Reinforcement Learning,” *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.
- [178] A. Jha, K. Kunal Singh, K. Vimala Devi, and V. Manjula, “Reinforcement learning based weighted multipath routing for datacenter networks,” *Materials Today: Proceedings*, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214785321003412>
- [179] V. Srivastava and R. S. Pandey, “Machine intelligence approach: To solve load balancing problem with high quality of service performance for multi-controller based Software Defined Network,” *Sustainable Computing: Informatics and Systems*, vol. 30, p. 100511, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210537921000044>
- [180] B. Babayigit and B. Ulu, “Deep learning for load balancing of SDN-based data center networks,” *International Journal of Communication Systems*, vol. 34, no. 7, p. e4760, 2021.
- [181] J. N. Witanto and H. Lim, “Software-defined networking application with deep deterministic policy gradient,” in *Proceedings of the 11th International Conference on Computer Modeling and Simulation*, 2019, pp. 176–179.
- [182] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.
- [183] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, “Automatic differentiation in machine learning: a survey,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5595–5637, 2017.
- [184] S. Wang and R. M. Summers, “Machine learning and radiology,” *Medical image analysis*, vol. 16, no. 5, pp. 933–951, 2012.
- [185] J. A. Boyan and M. L. Littman, “Packet routing in dynamically changing networks: A reinforcement learning approach,” in *Advances in neural information processing systems*, 1994, pp. 671–678.
- [186] T. T. Nguyen and G. Armitage, “A survey of techniques for internet traffic classification using machine learning,” *IEEE communications surveys & tutorials*, vol. 10, no. 4, pp. 56–76, 2008.
- [187] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” Internet Requests for Comments, RFC Editor, RFC 6550, March 2012, <http://www.rfc-editor.org/rfc/rfc6550.txt>. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6550.txt>
- [188] R. Vannithamby and S. Talwar, *Towards 5G: Applications, requirements and candidate technologies*. John Wiley & Sons, 2017.
- [189] G. A. Marin, “Network security basics,” *IEEE Security Privacy*, vol. 3, no. 6, pp. 68–72, 2005.
- [190] E. Rojas, “From Software-Defined to Human-Defined Networking: Challenges and Opportunities,” *IEEE Network*, vol. 32, no. 1, pp. 179–185, 2018.
- [191] E. L. Fernandes, E. Rojas, J. Alvarez-Horcajo, Z. L. Kis, D. Sanvito, N. Bonelli, C. Cascone, and C. E. Rothenberg, “The road to BOFUSS: The basic OpenFlow userspace software switch,” *Journal of Network and*

- Computer Applications*, vol. 165, p. 102685, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804520301594>
- [192] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming Protocol-Independent Packet Processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, p. 87–95, jul 2014. [Online]. Available: <https://doi.org/10.1145/2656877.2656890>
- [193] T. Høiland-Jørgensen, J. D. Brouer, D. Borkmann, J. Fastabend, T. Herbert, D. Ahern, and D. Miller, "The EXpress Data Path: Fast Programmable Packet Processing in the Operating System Kernel," in *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 54–66. [Online]. Available: <https://doi.org/10.1145/3281411.3281443>
- [194] R. Quinn, J. Kunz, A. Syed, J. Breen, S. Kasera, R. Ricci, and J. Van der Merwe, "KnowNet: Towards a knowledge plane for enterprise network management," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 249–256.
- [195] A. Wang, X. Mei, J. Croft, M. Caesar, and B. Godfrey, "Ravel: A Database-Defined Network," in *Proceedings of the Symposium on SDN Research*, ser. SOSR '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: <https://doi.org/10.1145/2890955.2890970>
- [196] M. Peuster, S. Schneider, and H. Karl, "The Softwarised Network Data Zoo," in *2019 15th International Conference on Network and Service Management (CNSM)*, 2019, pp. 1–5.
- [197] U. Paderborn, "Software Network Data Zoo (GitHub)." [Online]. Available: <https://github.com/sndzoo/>
- [198] S. Ketu and P. K. Mishra, "Cloud, Fog and Mist Computing in IoT: An Indication of Emerging Opportunities," *IETE Technical Review*, pp. 1–12, 2021.
- [199] N. Foster, N. McKeown, J. Rexford, G. Parulkar, L. Peterson, and O. Sunay, "Using Deep Programmability to Put Network Owners in Control," *SIGCOMM Comput. Commun. Rev.*, vol. 50, no. 4, p. 82–88, Oct. 2020. [Online]. Available: <https://doi.org/10.1145/3431832.3431842>
- [200] "Pronto Project." [Online]. Available: <https://prontoproject.org/>

...