Journal of King Saud University - Computer and Information Sciences 36 (2024) 102069

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University - Computer and Information Sciences

journal homepage: www.sciencedirect.com

Review article

A look into smart factory for Industrial IoT driven by SDN technology: A comprehensive survey of taxonomy, architectures, issues and future research orientations

Nteziriza Nkerabahizi Josbert, Min Wei *, Ping Wang, Ahsan Rafiq

Automation School, Key Lab of Industrial IoT and Networked Control, Ministry of Education, National Industrial IoT International S&T Cooperation Base, Chongqing University of Posts and Telecommunications, Chongqing, PR China

ARTICLE INFO

Keywords:
Software-Defined Networking (SDN)
Industrial Internet of Things (IIoT)
SDN-IIoT architecture
Software-Defined Industrial Network (SDIN)
SDN solutions

ABSTRACT

The Internet of Things (IoT) provides a major contribution to the innovation of smart manufacturing and industrial automation. Due to IoT, network devices and intelligent machines exchange information through different types of Internet connection and processes are predominantly automated. This reduces significantly the need for more human intervention and supports high performance. Nevertheless, the utilization of IoT in industrial automation called Industrial IoT (IIoT) has several issues, including the management of applications and IIoT devices. Moreover, heterogeneous networks and tremendous devices deployed in the IIoT environment require flexible configuration and reconfiguration according to the change for ensuring dynamic performance. We argue that Software-Defined Networking (SDN) is one of the technologies that can be used to solve some of the previously mentioned issues. In this paper, we propose a survey for the implementation of SDN solutions in IIoT and discuss the pros and cons brought about by this synergy named "SDN-IIoT". We explore the current articles on SDN-IIoT by considering different crucial domains such as flow installation techniques, fault tolerance, traffic routing optimization, resource management, energy efficiency, real-time, and network security. Furthermore, we analyze Artificial Intelligence (AI)/Machine Learning (ML) tasks to improve the performance of SDN-IIoT and the deployment of different technologies like Network Function Virtualization (NFV) and Time-Sensitive Networking (TSN) in SDN-IIoT. After observing the limitations of existing SDN-IIoT architectures, we propose an improved candidate architecture for SDN-IIoT based on a hierarchical distributed control plane. The new SDN-IIoT architecture contains AI, Industrial Backhaul Network (IBN), Dynamic Hash Table (DHT), AdaptFlow protocol, and edge/cloud storages. This paper selects the five most used SDN controllers by the literature review and identifies the features of each SDN controller. In the end, we provide open challenges and future research orientations in SDN-IIoT. We hope that this paper will be helpful for engineers, organizations, and researchers on the innovation of IIoT and SDN technologies.

1. Introduction

The innovation of wireless communications over the past few decades has given rise to a new era named the Internet of Things of which the short form is IoT (Salih et al., 2022; Bansal and Kumar, 2020). The IoT technology was initially presented by Ashton Kevin in 1998 as a platform for interconnecting physical things through the Internet (Khan et al., 2020). The IoT is expected to offer multiple supports in different areas such as smart buildings, agriculture, mobility, healthcare, and so on. Additionally, it is considered to have a major positive impact on industrial manufacturing by reaching more automation, improving monitoring, communication, and control. This technology is believed to accelerate development and bring innovation

* Corresponding author.
E-mail addresses: L201710007@stu.cqupt.edu.cn (N.N. Josbert), weimin@cqupt.edu.cn (M. Wei), wangping@cqupt.edu.cn (P. Wang), L201710003@stu.cqupt.edu.cn (A. Rafiq).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2024.102069
Received 14 January 2024; Received in revised form 6 May 2024; Accepted 17 May 2024
Available online 1 June 2024

Table 1
A comprehensive comparison between IoT and IIoT (Khan et al., 2020).

Consideration	IoT	IIoT
Field of target	Overall applications	Industrial applications
Innovation concentration	Smart nodes/devices	Sophisticated machines and industrial systems
Risk and security measures	Utility-centric	Robust and Advanced
Programmability	Simple off-site programming	Remote on-site programming
Interoperability	Functionally independent	Combination with current legacy Operating Systems
Scalability	Low-scale network	Large-scale network
Reliability	Not required	High-reliability required
Accuracy and precision	Critically monitored	Synchronized to milliseconds
Maintenance	Consumer preferred	Planned and scheduled
Output	Utilization and convenience	Efficiency performance

to industrial performance leading to the IIoT platform (Li et al., 2023). The IIoT permits the industry to gather and scrutinize a vast amount of data that can be utilized to enhance system performance and different categories of services. The IIoT concept is also deemed to minimize the expenses in terms of Operating Expenses (OPEX) and Capital Expenditures (CAPEX) (Ren et al., 2020; Wójcicki et al., 2022). Closely resembling terminologies are considered to explain the deployment of IoT in industrial networks, for instance, Industry 4.0/5.0/6.0 and smart manufacturing (Haghnegahdar et al., 2022; Noor-A-Rahim et al., 2022). The main idea behind all these terminologies is the utilization of IoT and sometimes with other emerging technologies (e.g., 5G/6G (Long et al., 2019), cloud/fog/edge computing (Jawed and Sajid, 2022; Goudarzi et al., 2022), ML (Jayalaxmi et al., 2022), NFV (Mostafavi et al., 2021), TSN (Lázaro et al., 2022), etc.), specifically for optimizing industrial communication and operations. Furthermore, the IoT brings smartness into the production system and supports information sharing during the manufacturing process (Khan et al., 2020).

IIoT provides proper management of industrial resources and processes alongside predictive maintenance. The development of IIoT is also widely anticipated in the next generation of industrial networks. The IIoT will allow the Industry 5.0 concept to close the gap between machines and people and it will facilitate the accomplishment of the Industry 6.0 vision (Maddikunta et al., 2022; Chourasia et al., 2022). The current prediction displays the notable progression in the domain of IIoT and IoT. In accordance with this prediction, there will be 70 billion physical nodes/devices interconnected to the Internet by 2025 and the end of 2023 the IIoT shares in the international market will be around US$14.20 trillion (Khan et al., 2020). Table 1 summarizes the main differences between IIoT and IoT technologies.

Fig. 1 shows a general IIoT network architecture consisting of four utility layers: data perception layer, data forwarding layer, data analysis layer, and application layer (Tange et al., 2020). The data perception layer uses sensing applications and information acquisition to accumulate different types of data and transmit them to the data forwarding layer through different communication technologies such as Bluetooth, WiFi, Machine-to-Machine (M2M) communication, 5G mmWave, and so on Cäsar et al. (2022), Mazhar et al. (2022), Wu et al. (2021). The data forwarding layer as a middleware layer sends the received data to the data analysis layer through the use of Dial on Demand Routing (DDR) or the shortest path routing (or other routing methods). This transmission is based on determined QoS requirements (delay-sensitive, loss-sensitive, etc.) and constraints (minimum bandwidth, maximum costs, network lag, etc.) of the IIoT system. The data analysis layer applies the data computing technologies (such as distributed computing, edge/fog/cloud computing, datacenter, etc.) to manipulate the data with low latency. It can additionally communicate with the application layer through the Application Programming Interface (API), to the extent that the different types of industrial applications can be updated and reconfigured according to the present IIoT requirements and the dynamic network status.

Scalability and flexibility required by IIoT networks are normally achieved through the use of wireless communication. Long ago, wireless networks in the industrial environment were mainly based on ad

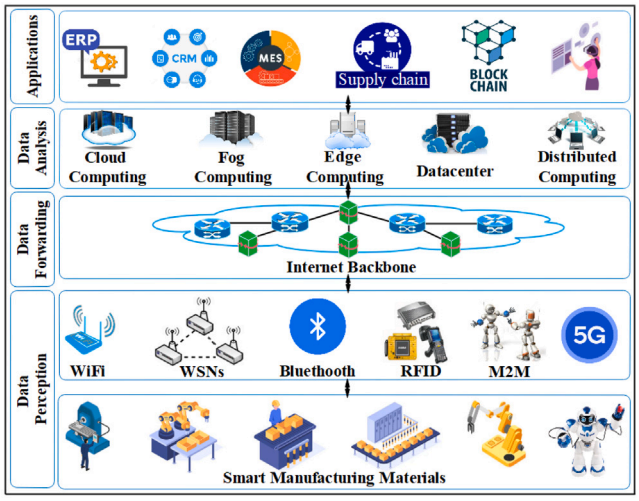

Fig. 1. IIoT network architecture.

hoc connections. A little while ago, standards protocols developed for optimizing industrial performance (e.g., WirelessHART (Devan et al., 2021), ISA100.11a (Padrah et al., 2021), WIA-PA (Tan et al., 2022), 6TiSCH (Kalita and Khatua, 2022), etc.) were available. Nonetheless, they have drawbacks relating to scalability and coverage in case there is a need to cover a large zone. While various versions of cellular networks such as 4G, 5G, and 6G technologies guarantee the connection of a huge number of network devices across a large area, they need licensed band and infrastructure support (Kutuzov et al., 2021). IIoT services usually require comparatively lower throughput per network device. Moreover, the demand of interlinking multiple nodes to the Internet at a low cost, with limited node capacities and energy resources (such as mini batteries) induces minimum cost, low-latency, and energy-efficient more desirable elements (Åkerberg et al., 2011).

Although there is an improvement in the protocols developed for IIoT and the benefits brought by this new technology, there are also various problems (Younan et al., 2020; Khan et al., 2020), a few of them are related to the connectivity of smart devices and network management. Especially, the system management of IIoT faces different issues, such as controlling and programming heterogeneous IIoT devices, scalability, dynamic performance based on network state changes, and scrutinizing a massive amount of data collected or generated by IIoT devices. In this survey, we argue that SDN technology can improve management and reduce the complexity of IIoT in the context of programmability, controllability, flexibility, and dynamicity.

SDN is a hot pattern that uncouples the control plane layer out of the data plane layer (Nisar et al., 2020; Yan et al., 2021; Khorsandroo et al., 2021; Hussain et al., 2022). Different from the traditional network, the switches deployed at the data plane level are only in charge of transmitting the data in accordance with the instructions from the control plane layer. The SDN controller is positioned in the control

plane layer as a network management entity, and targets to control every switch by using dedicated APIs (e.g., OpenFlow, OpenState, and LISP) (Pfaff et al., 2012). With the support of SDN, the management of IIoT applications can be improved, and network resources can be used more efficiently. As a matter of fact, in an IIoT environment with heterogeneous networks, the SDN platform can render the network with a consistent control plane, so that each node can be monitored in near real-time and the whole network can be regulated according to the predetermined rules based on the QoS requirements of IIoT (Naeem et al., 2020a; Xia et al., 2020). The common purpose of optimizing the IIoT communication performance can be reached via the deployment of SDN-based applications. Especially, industrial critical services, like the control of robots and Quality Control (QC) can take advantage of this platform in smart manufacturing (Singh Rajawat et al., 2021). However, a single SDN controller may not be appropriate for IIoT with multiple sections. On the other hand, when multiple controllers are applied in IIoT, complications of SDN controller placement happen (Isong et al., 2020). But, there is a possibility to choose the suitable distributed controllers design according to the services (Oktian et al., 2017; Zhang et al., 2021). Another issue for the utilization of SDN arises from its deployment in multi-hop architecture (Czachórski et al., 2021). To this end, the SDN paradigm may not be the best choice for some IIoT scenarios, since each technology has its own merits and demerits.

1.1. Motivation for this survey

The motivation for this survey lies in the appreciation of the significant augmentation of IoT/IIoT devices and their positive impacts on the innovation of industrial networks and smart manufacturing performance. IoT interlinked industrial apparatuses and nodes that are deployed to increase production rate, while minimizing operational expenses and manufacturing cycle time (Haghnegahdar et al., 2022). Moreover, multiple IIoT devices are currently interconnected to the industrial network, permitting the share of information, data gathering, and analysis. According to the Juniper report, there will be just over 37 billion IIoT network nodes/devices in 2025 (Mai et al., 2021a). Considering this large number of IIoT devices and different technologies, the IIoT networks are currently facing many issues like communication among heterogeneous IIoT devices, heterogeneous flows management, data inconsistency, high latency, data loss, cyber-attack, network element failure, and so on. These issues show that there is a need to design multiple frameworks so that the QoS requirements of IIoT and network users cannot be affected when things are connected (Atharvan et al., 2022).

Contrary to traditional networking services, IIoT services impose strict QoS requirements. Guaranteeing bandwidth, latency-sensitive, jitter-sensitive, loss-sensitive, high reliability, and network security are among those requirements (Mahmoodi et al., 2016). IIoT communication systems dictate constraints on latency and data loss. For instance, the M2M interface, smart manufacturing communication devices, and some IIoT processes require nearly 100% prosperous data delivery and high reliability with minimum latency (Zurawski, 2014). Particularly, latency requirements of smart factory applications vary from 0.3 ms (ms) to 10 ms, while the procedure automation can endure latency up to 100 ms (Schulz et al., 2017). Similarly, the loss of a few or many time-critical messages can disturb the protection functions as analyzed in Ali and Hussain (2017). The jitter enforced by IIoT and Industry 4.0 applications is lower or equivalent to 1 microsecond (μs) for motion control and lower or equivalent to 1 ms for cell control (Moutinho et al., 2019). Certain mobile robots may require 1–5 ms communication latency, a jitter to be under 50% of latency, and network reliability to be more than four nines (99.99%) (Bennis et al., 2018). Moreover, one of the main challenges in today's networks, especially in IIoT networks, is how to deploy the fault tolerance application in the system, that is, to provide the network configuration which is able to survive when a failure occurs. For example, between 2007 and 2013, cloud platforms from 28 cloud suppliers collected losses valued at 1600 h of network outages and U.S.$ 273 million due to multiple failures of applications, devices, and links (Fonseca and Mota, 2017). Furthermore, energy efficiency is one of the most important metrics in IIoT, as an insufficient resource can considerably jeopardize the lifetime of sensors and actuators. Some devices deployed in IIoT continue consuming a significant quantity of energy and augment the carbon footprint. Besides, there is considerable articles on AI and ML, which remarkably optimize the performance of SDN-IIoT (Lv et al., 2020; Foukalas and Tziouvaras, 2021). AI and ML offer convincing solutions to different automation systems and monitoring challenges. Hence, AI/ML can improve network management, analyze network traffic, automate complex processes and tasks within networks, and quickly identify network faults, etc.

It is evident that IIoT services may require one/double or several QoS requirements cited above, as well as fault tolerance, energy efficiency, and AI/ML. For that reason, it is necessary to review the fault tolerance domain, energy efficiency, multiple QoS requirements imposed by different IIoT services, and various flow installation techniques applied to guarantee the QoS requirements of IIoT, particularly latency-sensitive and loss-sensitive. Additionally, the applicability of the AI/ML algorithms in IIoT and different technologies such as TSN and NFV are considered due to their positive impacts on IIoT operations. TSN is a perfect candidate in promising low-latency for smart factories, and NFV utilizes hypervisor software to build virtual network resources that can simultaneously run numerous applications. One of the great advantages of NFV is network slicing (dividing the physical network into multiple small-scale logical networks). Thus, NFV is a better candidate for assuring efficient resources management for the IIoT.

Considering the above points, the need for systematic network management platforms and the suitable dynamic reconfiguration according to the network state change is increasing in IIoT. Furthermore, many IIoT nodes cannot be programmed to address complex policies and customize routing traffic because of the memory limitations. Thus, it is extremely difficult for the side of traditional networking to offer viable solutions regarding critical applications deployed in the IIoT system. Traditional networking also suffers modularity and scalability challenges, whereas SDN provides a global IIoT network management, dynamic reconfiguration, programmability, and it can also increase the security and resilience.

1.2. The objective of this survey

This survey analyzes multiple domains that are the most important in industrial networks, especially those following the IIoT platform. Some of them are adaptive transmission, fault tolerance, network security, traffic routing, resource management, flow installation techniques, real-time, and energy efficiency. SDN can be used to optimize these domains in terms of programmability, dynamicity, scalability, etc. Therefore, the objective of this survey is to determine and analyze a set of key points to be addressed in IIoT networks, and then consider SDN solutions to improve them. For instance, the IIoT applications face more serious security risks compared to traditional network applications due to heterogeneity on the side of IIoT (Serror et al., 2020). Then, we render a general investigation of network security approaches provided by SDN in IIoT, including the technologies used in network security such as blockchain, Manufacturer Usage Description (MUD), etc. We illustrate the merits of combining SDN-IIoT with other technologies, especially NFV and TSN, and analyzing a detailed background behind this combination. Additionally, this survey deals with how AI/ML algorithms can optimize the SDN-IIoT functionality in different scenarios. The survey also includes the comparison with different works, making it profitable to highlight the strengths and weaknesses of some models applied in the analyzed works.

An SDN-IIoT architecture with a hierarchical distributed control plane is presented based on three layers: the infrastructure layer, the control layer, and the application layer. This architecture includes our proposed IBN (Wang et al., 2019a), DHT (Josbert et al., 2021c), as well as adopted AdaptFlow API (Aujla et al., 2019) and AI. The existing works indicate that the research in this field is still in its inchoate level. We hope that another important contribution of this survey is defined by a thorough analysis of future research orientations related to SDN-IIoT. Thus, we have outlined sixteen open challenges including the improvement of simulation software and APIs, IIoT workstation safety, SDN-IIoT datasets, deployment of Wireless-TSN (WTSN) in SDN-IIoT, the utilization of wireless networks in SDN-IIoT, mobility management, and so on. To this end, we trust that this survey can render broad guidance for new researchers who would like to delve into this fascinating field.

1.3. Comparison with related survey papers and contributions from this survey

Most of the existing surveys on integrating SDN with other technologies focused on IoT and NFV. For instance, an SDIoT-Edge mechanism offers a comprehensive survey on the combination of edge computing, SDN, and IoT (Rafique et al., 2020). The authors in Imran et al. (2021) talking about the feasible alternatives for building SDN-IoT applications and IoT challenges solved through SDN solutions. In Li et al. (2020b), the authors discuss the perspectives of knowledge-driven SDN for IoT applications and demonstrate how to improve IIoT performance through the IoT architecture. Ref. Chaudhary et al. (2022) provides an extensive survey on the use of SDN in different intelligent applications. The survey includes details on SDN infrastructure, OpenFlow switches, programming languages, multiple SDN controllers, simulation software, and the current open issues in the implementation of SDN with different technologies like microservice architecture and 5G. The authors in Rahouti et al. (2020) propose a survey on SDN security communication infrastructures and analyze the applicability of SDN in smart city components such as energy, health, and transportation. Ref. Bekri et al. (2020) emphasizes on functionalities of IoT networks and the correlation between SDN and IoT, and the support of SDN to achieve the efficient management of IoT networks. The authors in Babiker Mohamed et al. (2022) investigate the adoption of NFV, SDN, and blockchain to improve the security of IoT networks. Ref. Alam et al. (2020) offers a survey for NFV and SDN platforms and their deployment in IoT networks. The authors in Shah et al. (2021) analyze network slicing and its related technologies like NFV, Multi-Access Edge Computing (MEC), SDN, and cloud-native 5G core. Ref. Abid et al. (2022) highlights the traditional IoT concepts, their limitations, and how they can be optimized via NFV and SDN for designing smart IoT networks. The authors in Khorsandroo et al. (2021) provide a survey of hybrid SDN in which SDN practicalities are leveraged while the present infrastructures of the traditional network are acknowledged. In Manguri and Omer (2022), the authors review different IoT aspects and domains, including IoT management, network security, cellular networks, wireless sensors, smart city mechanisms, and the general pros and cons of SDN-IoT are outlined. The authors in Kafetzis et al. (2022) propose a review of SDN and Software-Defined Radio (SDR) with cross-layer optimization, as well as analyze the limitations related to the synergy of SDR and SDN technologies. Ref. Ja'afreh et al. (2022) presents a review of IoT and SDN, and discusses the collaboration of these technologies, architectures, and open issues. Ref. Yungaicela-Naula et al. (2022) presents the literature review on security automation in multiple SDN scenarios. In Barakabitze et al. (2020), the authors offer a survey by considering the current solutions related to 5G network slicing and its application in NFV and SDN. In Wu et al. (2022), the authors provide comprehensive research on network slicing management in common use cases and analyze IIoT components such as smart energy, smart transportation, autonomous vehicles, and smart factory. As noted above, the authors specify that in smart factories, supporting use cases such as motion control, closed-loop control, emergency stops for a mobile robot, and sensor networks during the monitoring process, all required communication latencies between 1 ms and 10 ms. In Etxezarreta et al. (2023), the authors propose a survey on the security of Industrial Control Systems (ICSs) and the role of SDN in developing intrusion response mechanisms to mitigate unauthorized users.

Few surveys specifically focus on SDN-IIoT networks. For instance, Ref. Urrea and Benítez (2021) presents the solutions provided by SDN in the industrial network/IIoT and displayed the SDN-IIoT architecture with distributed controllers, as well as highlighting the controllers which are appropriate for the industrial network. However, different essential domains for SDN-IIoT are not considered, such as the deployment of AI/ML in SDN-IIoT, real-time, advanced and standard flow installation techniques, fault tolerance management, TSN, and energy efficiency. The authors in Jiang et al. (2022) investigate how SDN combined with AI can improve functionalities and network security of IIoT. Nevertheless, the authors did not take into account the following domains for SDN-IIoT: real-time, advanced and standard flow installation techniques, NFV, fault tolerance management, and energy efficiency. Therefore, the observed gaps from the previous surveys in the field of SDN-IIoT are deeply investigated in this survey. Furthermore, the domains that have already been analyzed in previous surveys are not neglected in this survey (e.g., traffic routing optimization, resource management, network security, and the synergy of NFV with SDN-IIoT). Only three survey papers introduce the SDN-IIoT architecture. However, the existing SDN-IIoT architectures (Urrea and Benítez, 2021; Jiang et al., 2022; Li et al., 2020b), missing the IBN, DHT, edge storage, cloud storage, AdaptFlow, and IIoT controller as the SDN application. Consequently, our proposed SDN-IIoT architecture includes the missing technologies observed in the existing architectures to improve the performance and guarantee QoS requirements imposed by IIoT. After exploring all the above-discussed proposals, a comparative analysis is shown in Table 2. It is relevant to mention that none of the proposals discussed above can take into account any aspect related to advanced flow installation techniques.

The main contributions of this survey are:

- We propose a candidate architecture for SDN-IIoT based on a hierarchical distributed control plane structure to reduce latency and simplify management, as well as avoid a single point of failure. This new architecture comprises the IBN, DHT, AdaptFlow, edge/cloud storage, AI technology, etc.
- We discuss, analyze, and compare the current articles on industrial networks, especially those related to the IIoT paradigm leveraging SDN technology.
- We review the current articles considering different main aspects such as flow installation techniques, fault tolerance, traffic routing optimization, resource management, energy efficiency, real-time, and network security.
- Dual emerging technologies named NFV and TSN, and their deployment in SDN-IIoT are discussed.
- In order to provide an intelligent network, the analysis of how AI/ML algorithms can optimize the functionality of SDN-IIoT from the viewpoint of different scenarios is taken into account.
- In the literature review, different SDN controllers have been used in the simulation to test the performance of the proposed solutions. In this survey, we analyze the five most applied SDN controllers in the literature review due to their capability and suitability of managing IIoT networks. Moreover, several characteristics are outlined to choose the most appropriate SDN controllers based on several Multi-Criteria Decision Analysis (MCDA) approaches.
- Finally, current open research challenges are displayed to provide a guideline for future research directions in the SDN-IIoT domain.

Table 2
A comparison of the previous surveys focused on SDN-IIoT and SDN-IoT.

Ref.	Year	SDN	IIoT	SIF	SIA	AI/ML	Real-time	SAFIT	TSN	NFV	NS	FT/R	EC/CC
Khorsandroo et al. (2021)	2021	✓	×	×	×	✓	×	×	×	✓	✓	✓	✓
Rafique et al. (2020)	2020	✓	×	×	×	✓	✓	×	×	✓	✓	✓	✓
Imran et al. (2021)	2021	✓	×	×	×	✓	✓	×	×	✓	✓	×	✓
Li et al. (2020b)	2020	✓	✓	×	✓	✓	×	×	×	✓	✓	×	✓
Chaudhary et al. (2022)	2022	✓	×	×	×	×	✓	×	×	✓	✓	✓	✓
Rahouti et al. (2020)	2020	✓	×	×	×	×	✓	×	×	✓	✓	×	✓
Bekri et al. (2020)	2020	✓	×	×	×	×	✓	×	×	✓	✓	×	✓
Babiker Mohamed et al. (2022)	2021	✓	×	×	×	✓	×	×	×	✓	✓	×	✓
Alam et al. (2020)	2020	✓	✓	×	×	✓	✓	×	×	✓	✓	×	✓
Shah et al. (2021)	2021	✓	×	×	×	✓	✓	×	×	✓	×	×	✓
Abid et al. (2022)	2022	✓	✓	×	×	✓	×	×	×	✓	✓	×	✓
Manguri and Omer (2022)	2022	✓	×	×	×	×	×	×	×	×	✓	×	✓
Kafetzis et al. (2022)	2022	✓	×	×	×	✓	×	×	×	✓	×	×	✓
Ja'afreh et al. (2022)	2021	✓	×	×	×	×	✓	×	×	✓	✓	×	✓
Yungaicela-Naula et al. (2022)	2022	✓	×	×	×	✓	×	×	×	✓	✓	×	×
Barakabitze et al. (2020)	2020	✓	×	×	×	×	×	×	×	✓	✓	×	✓
Urrea and Benítez (2021)	2021	✓	✓	✓	✓	×	×	×	×	✓	✓	×	✓
Jiang et al. (2022)	2022	✓	✓	✓	✓	✓	×	×	✓	×	✓	×	✓
Wu et al. (2022)	2022	✓	✓	×	×	✓	✓	×	✓	✓	✓	×	✓
Etxezarreta et al. (2023)	2023	✓	×	×	×	✓	✓	×	×	✓	✓	×	×
Our Survey	2024	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Ref.: Reference; SIF: SDN-IIoT Framework; SIA: SDN-IIoT Architecture; SAFIT: Standard and Advanced Flow Installation Techniques; NS: Network Security; FT/R: Fault Tolerance/Resilience; EC/CC: Edge Computing/Cloud Computing; ✓: Considered; ×: Not considered.

1.4. Organization of the survey

The structure of this survey is given in Fig. 2 and organized as follows: Section 2 provides the background including the introduction of SDN, OpenFlow, IIoT technology, IIoT/industrial network standard protocols, and computing/networking in IIoT. Section 3 illustrates a new SDN-IIoT architecture with its three layers. Section 4 presents advanced and standard flow installation techniques in SDN-IIoT. Section 5 renders fault tolerance mechanisms in SDN-IIoT. Section 6 analyzes the traffic routing optimization while taking into account the QoS requirements of IIoT and resource management. Section 7 demonstrates the essential domains in SDN-IIoT, including IIoT leveraging the SDN-based testbed, energy efficiency, and real-time. Section 8 presents the synergy of different technologies (TSN and NFV) in SDN-IIoT. Section 9 introduces the applicability of ML/AI in SDN-IIoT. Section 10 discusses the network security in SDN-IIoT. Section 11 selects the suitable SDN controllers for IIoT. Section 12 illustrates current limitations, challenges, and future research orientations in SDN-IIoT. Finally, Section 13 concludes the survey and covers topics for future work.

2. Background

In this section, there is an overview of the ordinary SDN architecture and its comparison with the traditional networking architecture. Moreover, we provide a brief introduction of OpenFlow, IIoT, and IIoT/industrial network standard protocols, as well as computing and networking in IIoT.

2.1. SDN technology

SDN has recently attracted more attention due to handling many networking problems, and its main idea is to decouple the control plane from the implementation devices to simplify network programmability (Klimis, 2021; Ali et al., 2023). This decoupling allows network administrators to dynamically direct traffic flows and simplify network management by using automation and centralized control. SDN renders various advantages, including improved network agility, flexibility, scalability, and cost-efficiency. One of the main benefits of SDN is its capability to abstract policies and network intelligence from the physical hardware. This abstraction facilitates network management tasks, such as troubleshooting, monitoring, and provisioning, while also offering a high level of orchestration and programmability. Furthermore, SDN supports network virtualization and enables dynamic allocation of resources, on-demand service provisioning, and effective utilization of network resources. By leveraging the SDN controller and APIs, network administrators can adjust network configurations based on real-time conditions and service requirements, impose security measures, and optimize traffic routing. Again, the SDN allows the control of operations using a software entity named the SDN controller. The Open Networking Foundation (ONF) (Open Networking Foundation, 2023) is a non-commercial consortium committed to the implementation, standardization, and advancement of the SDN architecture.

Fig. 3 depicts an SDN architecture made up of three planes as follows: application plane, control and data planes (Haleplidis et al., 2015a). The data plane is made up of transmission devices (e.g., OpenFlow switches) that distribute the incoming data packets in the entire system. The data plane is located at the bottom layer of the SDN architecture, comprising forwarding nodes like switches and routers, whether virtual or hardware. Virtual switches are software-based switches that execute on a general Operating System. Some examples of virtual switches are Indigo (Yang et al., 2017), Pantou (Anon, 2024), and Open vSwitch (Open vSwitch, 2024). Hardware switches are physical-based switches, developed by using open network hardware like NetFPGA (Chu et al., 2018), or on a merchant switch from networking hardware sellers. Examples of hardware switches based on NetFPGA include switchBlade (Yan et al., 2020) and ServerSwitch (Zeng et al., 2020). Hardware sellers implement their commercial switches with the support of SDN protocols. Virtual switches are more compatible and flexible with SDN protocols compared to hardware switches. However, hardware switches have a high rate of flow forwarding in comparison

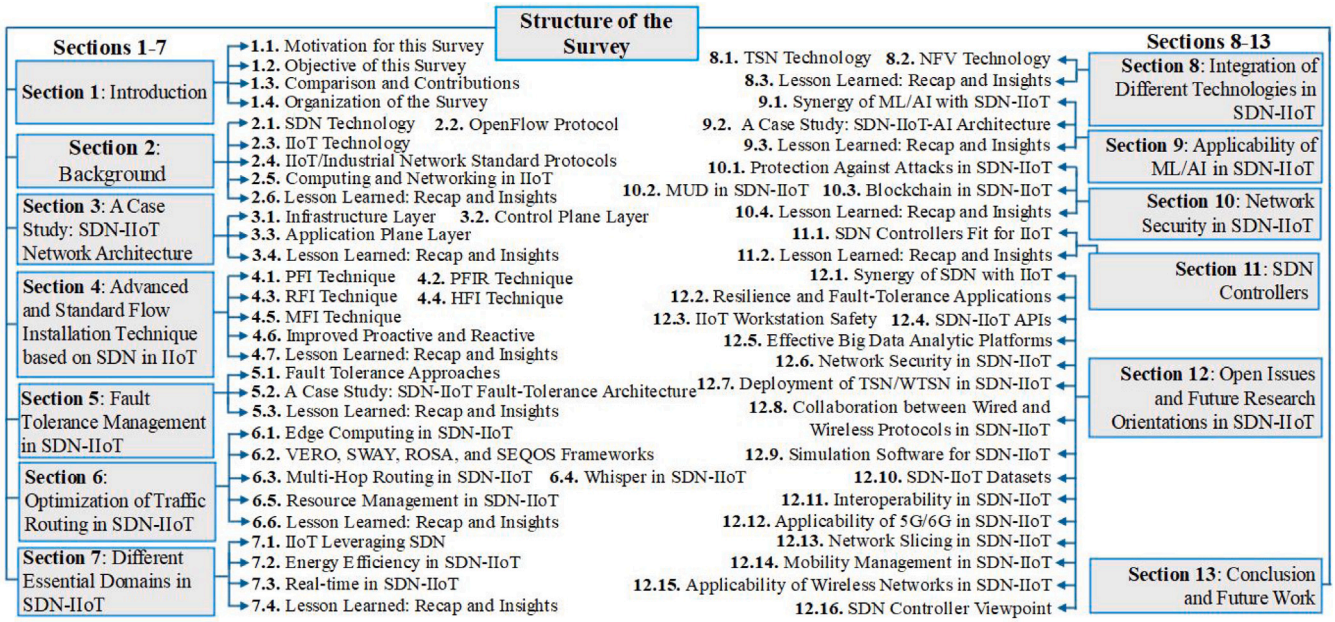

Fig. 2. Structure of the survey.

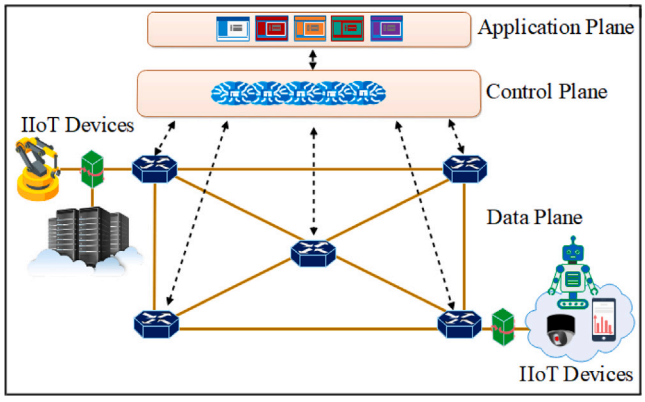

Fig. 3. SDN architecture.

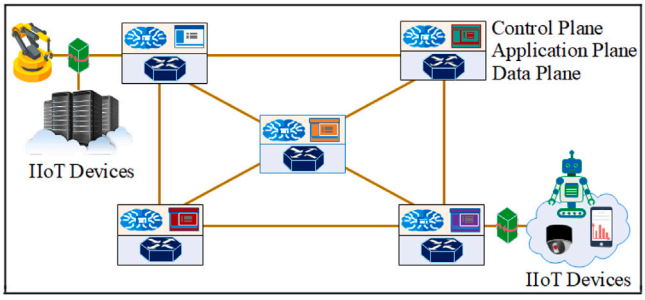

Fig. 4. Traditional network architecture.

to virtual switches. Both are utilized for modifying, dropping, and forwarding data packets based on the rules from the control plane logic. These devices are managed by the control plane logic through the southbound API.

The control plane is designated as the intelligence part of the networks that regulate diverse types of data planes through southbound API and it acts as the intermediate between the application plane and data plane devices. In SDN, the control plane works as the brain that performs different tasks; for example, it uses flow rules to deal with the incoming ethernet frames and determine their destination ports. The control plane translates application plane requirements into regulations and inserts these regulations inside data plane devices. The control plane duties include network monitoring, configuration/reconfiguration of devices, computing shortest path routing, etc. The control plane used southbound APIs to communicate with data plane devices and exchange control policies based on the current network state information. In distributed control plane architecture, westbound and eastbound interfaces are utilized to provide communication between SDN controllers. the exchange of information is vital between controllers to offer a global network view to the applications. Westbound and eastbound interfaces are private and cannot communicate with each other.

The application plane layer at the top of the architecture permits the development of different applications and this layer brings openness, innovation, and flexibility to network suppliers. These applications render management and optimization of business services based on the current network state information received from the SDN controller to update the network behavior. The application plane layer is connected with the control plane layer through northbound API.

In the traditional network architecture, the above three planes are combined inside the devices (e.g., switches and routers) as shown in Fig. 4. Network services such as firewalls, load balancers, and routing were offered by vendor-specific hardware, and network configuration/reconfiguration and other updates across different devices required to be done manually. The complexity resulting from this architecture does not only increase delay, but it also increases OPEX. Moreover, troubleshooting while installing this network requires extra materials which, consequently, further enlarges CAPEX (Haji et al., 2021).

In SDN, respective northbound APIs as the intermediate between the application plane layer and the control plane layer have been

developed to offer prominent abstractions to the applications deployed in the control plane layer. One of the paramount northbound APIs is REST API. On the other hand, standard OpenFlow (Pfaff et al., 2012) is the main southbound API that allows the network to be controlled efficiently. However, SDN cannot be limited to OpenFlow as other additional interfaces exist like POF (Li et al., 2017), NETCONF (Kunz and Muthukumar, 2017), OVSDB (Pfaff and Davie, 2013), OpenState (Bianchi et al., 2014), ForCES (Haleplidis et al., 2015b), OpFlex (Smith et al., 2014), and LISP (Rodriguez-Natal et al., 2015). OpenFlow conducts all the intelligence elements at the control plane level, which allows the decoupling of the control plane from the transmitting plane. The next subsection presented more details about the OpenFlow protocol.

2.2. OpenFlow protocol

The OpenFlow protocol offers an interface-based communication between the controller layer and the infrastructure layer (Pfaff et al., 2012). Furthermore, it provides a way for regulating nodes (e.g., OpenFlow switches) without caring about the source code developed by the suppliers. Shortly, OpenFlow renders a line to immediately access and manage the forwarding nodes. Moreover, it allows access to flow tables and inserts the rules into these flow tables under the SDN controller guidance to conduct network traffic in a secure channel. OpenFlow enables the control plane to update and modify the flow network in a short span of time (Mondal et al., 2019). According to the existing literature, there are dual categories of OpenFlow switches. The first is hybrid-OpenFlow and the second is only-OpenFlow. Hybrid-OpenFlow is compatible with OpenFlow switches and standard Ethernet switches, whereas only-OpenFlow is compatible only with OpenFlow switches. A switch comprises multiple flow tables to perform packet tracing and forwarding operations. Every flow table comprises flow entries; and every flow entry comprises the following essential elements: priority, cookie, instructions, match fields, counters, and timeouts. Priority provides matching preeminence of the flow entry. Cookie opaque the value of data according to the choice from the control plane. It can be utilized by the control plane to filter flow deletion, flow modification, and flow statistics but is not utilized in the case of packet processing. Instruction offers information about manipulating and packet matching; their transmitting action is also defined, such as being transmitted to ports or SDN controller or occasionally deleted. The goal of the match fields is to match against data packets based on details regarding ID, VLAN, source port, IP address, destination port, etc. There are counters for holding details regarding the number of packets and updating when packets are matched. Timeout provides the maximum period or idle time before the instructions are expired inside the flow table (Zhao et al., 2019).

The OpenFlow channel renders an interface for linking the controller and switches. Through this interface, the controller can manage and configure the switches easily. Firstly, when the incident is acquired, multiple messages are transmitted through this channel, comprising asynchronous messages that contain messages to inform the controller about the update related to network state change and new incident. The second interaction involves controller-to-switch messages for the purpose of controlling and checking the switch status. Finally, symmetric messages are introduced by the controller or the switch and are transmitted without request (Li et al., 2016a). To build communication, the OpenFlow channel can be connected to one or several controllers known as the distributed control plane architecture. The distributed control plane architecture can enhance reliability when one of the controllers fails (Priya and Radhika, 2019). The P4 interface is also utilized for establishing the connection between the data plane and the control plane as it can manage the functionality of the data plane effectively (Liatifis et al., 2023).

2.3. IIoT technology

IIoT technology interconnects industrial apparatuses and nodes communicating in real-time with manufacturing process monitoring to attain high productivity while minimizing operating costs. The IIoT is a subsection of IoT that requires efficient communication and high security due to coping with mission-critical services, which are intended to operate with high reliability and without disruption of real-time communication (Atharvan et al., 2022). An IIoT system includes IoT devices to permit everything at any time to be connected in the smart factory in order to improve productiveness, functionality, protection, and cleverness. The IIoT system cannot be complete without Wireless Sensor Networks (WSNs) and middleware software that can monitor them (Aazam et al., 2018). Fig. 5 shows the application areas of IIoT. More specifically, the industrial and manufacturing factory areas were considered in this survey.

2.4. IIoT/Industrial network standard protocols

In industrial settings, protocols allow connected apparatuses/nodes to interact with each other. It is better when the deployed protocols are able to connect apparatuses/nodes made by different companies with different functions and internal structures so that sub-systems can share information and communicate automatically to build an intelligence network. Human activities are decreased when industrial processes are automated. In this subsection, we provide a summary of the following IIoT/industrial network protocols: WIA-PA, 6TiSCH, 6LoWPAN, OPC-UA, HART/WirelessHART, PROFIBUS/PROFINET, Modbus, and ISA100.11a.

2.4.1. WIA-PA

WIA-PA (Wireless Networks for Industrial Automation-Process Automation) is an industrial wireless networking protocol developed to support communication in industrial automation. In 2008, the International Electrotechnical Commission (IEC) approved WIA-PA and joined the list of other industrial wireless network standards that existed at the time, such as the WirelessHART protocol. WIA-PA offers a wireless connection service for automation nodes to carry out measurement, monitoring, and loop control for industry procedures (Tan et al., 2022; Wei et al., 2020). Based on the testbed, the synergy between the WIA-PA network and SDN technology is demonstrated to improve data flow scheduling and dynamic management of industrial communication (Wei et al., 2021). However, WIA-PA and wirelessHART protocols failed to support the IPv6 network. It is profitable to make these protocols access IPv6 to guarantee the deterministic communication required in many industrial applications. To address this issue, we propose a framework for industrial wireless networks that allows the WIA-PA and WirelessHART to be compatible with the IPv6 network, where each deployed gateway offers IPv6-enabled interfaces. Based on WIA-PA and WirelessHART nodes accessing the IPv6 network, a confirmation testbed in an assembly manufacturing line that can fulfill the remote real-time monitoring of an Automated Guided Vehicle (AGV) is illustrated (Wei et al., 2020).

2.4.2. 6LoWPAN

The 6LoWPAN protocol is a simplified form of IPv6 over a Low-Power Wireless Personal Area Network and it is compatible with several IEEE 802.15.4 protocols and the IPv6 network. Furthermore, 6LoWPAN is affordable, versatile, and consumes less energy (Das et al., 2022). These characteristics make it appropriate for many IIoT applications. The verification scenario and performance evaluation of the 6LoWPAN protocol in the SDN-IoT ecosystem through the Mininet-IoT simulator are well illustrated in Setiawan et al. (2021).

2.4.3. 6TiSCH

The IETF IPv6 over the TSCH form of IEEE802.15.4e working group has standardized a set of formalities to allow interoperability between the industrial level and IPv6 networks (Vilajosana et al., 2019). The

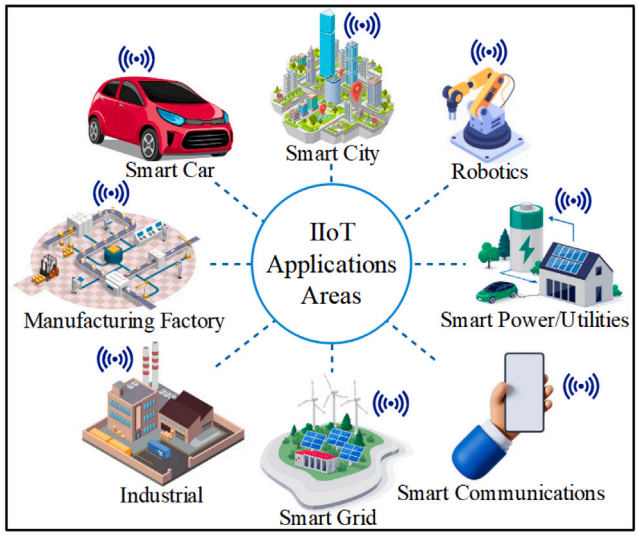

Fig. 5. IIoT applications areas.

primary focus of the 6TiSCH protocol is to establish guidelines for how nodes perform and facilitate the configuration of TSCH slot frames. This protocol facilitates default scheduling tasks for timeslots dynamic scheduling of IPv6 traffic and defines an interface to allow deterministic networking (Kalita and Khatua, 2022). 6TiSCH has emerged as one of the predominant IIoT protocols due to the energy efficiency achieved by amalgamating tools and manipulating the scenario of the IIoT scale. Also, 6TiSCH is capable of promising efficient allocation of nodes and distribution of radio resources, as well as enabling scalability using frequency diversity and fine-scheduled time. However, 6TiSCH contains multiple challenges that can be tackled by utilizing SDN technology. For instance, in Thubert et al. (2015), the authors proposed how deterministic networking and 6TiSCH can use a centralized control plane architecture to provide dynamic configuration and build the future of IIoT technology.

2.4.4. OPC-UA

The OPC-UA permits the standard for data exchange with protection and platform-independent. It facilitates M2M communication and IIoT connectivity that integrate automation systems, nodes, and apparatuses with software applications (Lee and Sung, 2022).

2.4.5. HART/WirelessHART

HART is a bi-directional network standard that offers access to information among host platforms and wise field devices. The WirelessHART protocol was implemented to extend the HART standard to be compatible with wireless networks. The WirelessHART protocol also maintains interoperability with current/legacy HART nodes (Devan et al., 2021).

2.4.6. PROFIBUS and PROFINET

PROFIBUS is a digital network in charge of facilitating communication among controllers or control systems and field sensors. The initial idea for the implementation of PROFIBUS was the solution to factory automation systems and industrial production processes (Vadi et al., 2022). PROFIBUS and PROFINET are both IEC standards designed by the same company but are not the same, PROFIBUS is a typical serial Fieldbus, whereas PROFINET is an industrial Ethernet protocol. Compared to PROFIBUS, PROFINET has a substantially higher transmission speed. PROFINET-enabled Ethernet communication has been designed and it has already been deployed in multiple technologies and domains such as Industry 4.0 and 5G (Ficzere et al., 2022).

2.4.7. Modbus

The Modbus network protocol is the most utilized in the manufacturing branch of industry. Some important responsibilities of this protocol are the exchange of information among Human–Machine Interface (HMI), Supervisory Control and Data Acquisition (SCADA), and Programmable Logic Controller (PLC). Furthermore, Modbus boosts a hardware-agnostic to permit compatibility among diverse automation devices (Elamanov et al., 2022).

2.4.8. ISA100.11a

The ISA100.11a protocol has been implemented and promoted by the International Society of Automation (ISA) to deal with some issues of wireless network communication in industrial backgrounds, such as manufacturing plants and petroleum refineries. The ISA100.11a protocol is founded on the IEEE 802.15.4 standard and it offers a reliable technique of data collection that is more cost-effective compared to traditional wired networks. Besides, as ISA100.11a accessing IPv6, it holds multiple advantages of IPv6 over IPv4 utilized in IP networks (Jecan et al., 2022).

2.5. Computing and networking in IIoT

The IIoT system contains multiple apparatuses and network devices such as smart machines, sensors, actuators, switches, gateways, computers, etc. All the steps of smart manufacturing are continuously monitored through the support of sensing technologies (Shu et al., 2017). Therefore, a large amount of data is complicated to manipulate, and different technologies like big data analytics, edge computing, fog computing, and cloud computing are utilized to improve data processing and help in decision-making. Again, cloud computing deals with the challenges of huge storage, high-performance computing, long-standing battery power, and so on. Cloud computing provides on-demand provisioning of diverse platforms, applications, and miscellaneous computing infrastructures such as storage and servers (Khan et al., 2024). Cloud computing consists of three service models as follows. Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). For SaaS, users can utilize different services and applications running on cloud infrastructure, as well as ready-to-use diverse application software through the Internet. Some of these application software are managed by a service provider at its data center, permitting customers to access them through a standard web browser. PaaS offers a platform that enables customers to develop, run, and manage various applications without the challenges of constructing and maintaining the cloud infrastructure. On the other hand, IaaS provides virtualized computing resources, self-service options for managing, monitoring, and accessing remote data center infrastructures, such as networking services and storage. Multiple big companies have built their own cloud platforms, such as Microsoft Azure for Microsoft Company (Microsoft Azure Cloud Computing Platform & Services, 2024), Google Compute Engine (GCE) for Google Company (Google Cloud Computing, Hosting Services & APIs |, 2024), Amazon Web Services (AWS) for Amazon Company (Gupta et al., 2021), etc. In this context, there are four types of cloud computing as follows: private cloud, public cloud, multi-cloud, and hybrid cloud. The private cloud is dedicated to one company, whereas the public cloud is operated and owned by third-party service providers. Multi-cloud and hybrid cloud approaches combine both private and public cloud resources.

Nonetheless, instead of totally depending on a single cloud entity in the form of a centralized network architecture, edge devices such as smartphones, tablets, and computers are located near the nodes and smart machines. Thus, some challenges such as high latency and packet loss can be efficiently solved (Qiu et al., 2020). Edge computing is a distributed computing technology that brings data storage and computation closer to where data is generated. In contrast to traditional cloud computing, which centralizes data processing in remote data

centers, edge computing allows data analysis and processing to happen near the data source, at the network edge. This proximity reduces the delay it takes for data to move between nodes and the computing infrastructure. Low delay is crucial for services/technologies that require real-time responsiveness, such as IIoT, industrial automation, autonomous vehicles, and Virtual Reality/Augmented Reality (VR/AR) experiences. Edge computing supports IIoT services by facilitating predictive maintenance, real-time monitoring, and process optimization in manufacturing plants. By processing sensor/actuator data at the edge level, IIoT can minimize downtime, enhance efficiency, and improve Overall Equipment Effectiveness (OEE). However, due to the limited computing and storage capability at the network edge, edge computing typically cooperates with cloud computing, known as edge–cloud orchestration (Taleb et al., 2017; Wu, 2020). Hence, edge nodes would just complement the cloud. In fact, edge computing focuses on determining which data has to be analyzed at a specific point (for example, deciding which data should be kept at the edge and which should be pushed to the cloud) (Cao et al., 2021; Arthurs et al., 2021). Therefore, the synergy between edge and cloud is essential. As each coin has two sides, edge computing has also several limitations including incomplete tasks for the management system, synchronization, data security and privacy, and the difficulty of dealing with heterogeneous technologies deployed in the same system (Yu et al., 2017; Ferrag et al., 2022).

Fog computing allows storage, data, and application services to be distributed nearer to the end-users and their devices, typically at the local network infrastructure or within the network edge. The concept of "fog" highlights the idea of bringing the cloud nearer to the ground, indicating proximity to network devices and their real-time data sources. Because of these, fog computing presents various advantages, such as minimized latency, enhanced bandwidth efficiency, reliability, as well as security and privacy for sensitive data. In Ahuja and Deval (2021), the authors discuss the connection between fog computing and cloud computing. They illustrated that fog and cloud are complementary technologies and none of them replaces the other. The distinction between fog and cloud depends on the sorts of data required and the speed of data processed. Local data can be handled by fog nodes, while global data can be handled by the cloud. The proximity to end users makes the fog technology more appropriate for IIoT services compared to the cloud technology (Basir et al., 2019). Furthermore, fog computing operates as an intermediate between the edge layer and the cloud layer for different targets such as data filtering, fast response time, etc. However, fog computing is not able to replace edge computing, whereas edge computing can function without fog computing in multiple applications (Sarkar et al., 2023). Fog computing plays a key role in supporting smart manufacturing in IIoT by facilitating quality inspection at the network edge, predictive maintenance, and real-time control. By scrutinizing sensor data and equipment telemetry locally, manufacturing procedures can become responsive, adaptive, and more efficient. However, managing a distributed fog infrastructure, which includes a variety of edge devices and gateways, poses difficulties in terms of resource allocation, orchestration, scalability, as well as QoS guarantees regarding capacity and storage. In this context, standardizing protocols and interfaces can promote synergy and interoperability among heterogeneous fog computing platforms, as well as achieve the complete prospects of fog computing (Srirama, 2024). Fig. 6 shows the three computing layers in the IIoT environment.

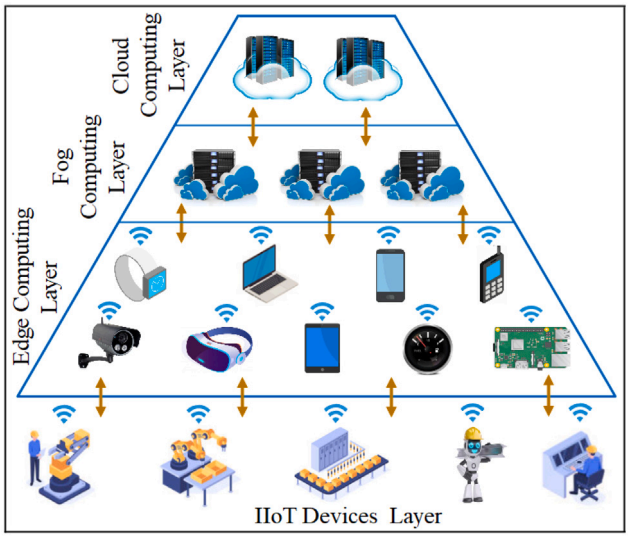

Fig. 6. Layers of computing in the IIoT environment.

2.6. Lesson learned: Recap and insights

The idea of SDN technology covers the concept that network nodes (e.g., switches) can be transformed to forwarding devices and updated like applications on a smartphone. In order to execute many network functions, these devices are programmed by an intelligent central controller in line with the network requirements from the application layer. This robust idea permits network decisions to be made based on the global network state, instead of taking them locally on the network infrastructure. However, in certain scenarios, using SDN may experience different challenges, especially when scaling up the network connectivity or handling a multitude of network traffic. It is critical to carefully design the network system and conduct the performance tests to guarantee that the use of SDN can efficiently address the desired workload. Furthermore, IIoT and smart manufacturing should consider the deployment of high-performance devices and choose the controller type based on their network requirements. As depicted in Fig. 3, the SDN network architecture is separated into three planes: data plane, control and application planes. In the traditional network architecture (see Fig. 4), all these three planes are combined inside the switches and configuration is independently completed in every switch across the network topology. This strategy increases complexity and takes considerable time in the configuration phase which is done manually. For the SDN network architecture, OpenFlow protocol is introduced as one of the standard southbound APIs. Nonetheless, OpenFlow may face issues when attempting to scale large networks with multiple hosts and switches. Performance challenges can arise when the centralized controller becomes a bottleneck. To address scalability issues, the distributed control plane or the hierarchical architecture can be adopted. This includes decoupling the network into smaller sections, each with its own local SDN controller, reducing the controller overhead and simplifying the management of tasks. On the other hand, the distributed control pane and the hierarchical architecture generate several other issues, for instance, SDN controller placement, synchronization of controller state, and east/westbound APIs security. Therefore, developing strong network security measures can help mitigate security threats. Monitoring, frequent network security audits, and patch management are also important to maintain a secure SDN-IIoT system.

As we presented in this section, IIoT connects industrial infrastructures, such as apparatuses, smart machines, and nodes, with information and control systems, as well as the engineers who are utilizing it for improving manufacturing and business operations. Based on connectivity, we can collect different types of data, then analyze it using modern data analysis applications to gain insights and provide optimum decision-making to manage industrial processes. However, with an increased communication between systems and devices in IIoT, the risk of data breaches and cyber-attacks increases. Malicious actors can take advantage of vulnerabilities in the network and obtain unauthorized access to critical applications. Therefore, it is essential to implement robust security mechanisms and regularly update and patch network nodes to prevent against external threats.

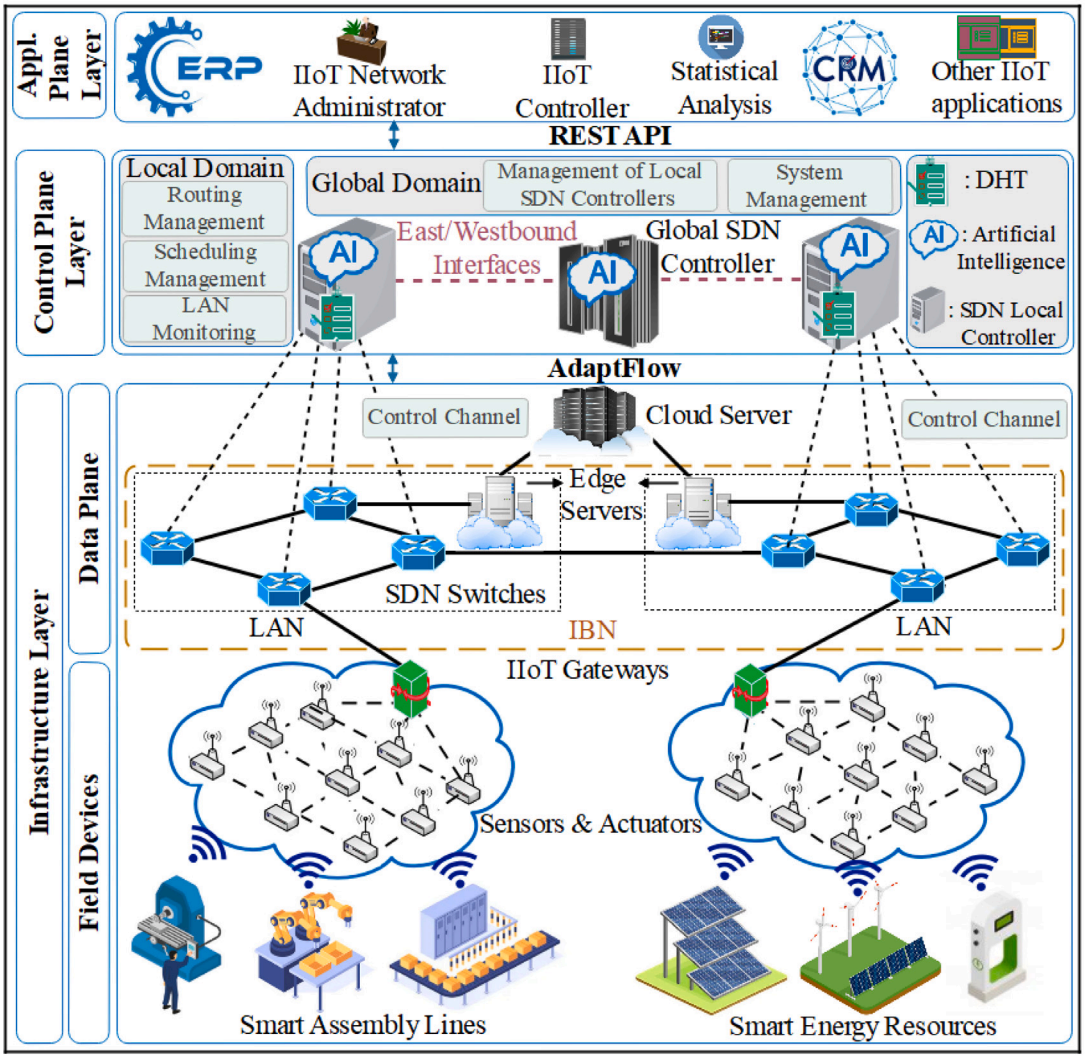

Fig. 7. SDN-IIoT architecture.

3. A case study based on SDN-IIoT architecture

Concerning this section, we discuss a case study based on an SDN-IIoT architecture, as displayed in Fig. 7. At the bottom of the architecture, there is an infrastructure layer that contains field devices (e.g., smart assembly lines, smart energy resource apparatuses, sensors, actuators, gateways, etc.) and data plane switches deployed inside IBN. A control plane layer as the mediator of the architecture contains local SDN controllers and a global SDN controller. Mainly, this layer controls the infrastructure layer, especially the data plane switches through an industrial southbound interface named AdaptFlow. Therefore, the AdaptFlow interface is adopted in the architecture due to minimized energy consumption and latency while boosting network capacity compared to the standard OpenFlow protocol (Aujla et al., 2019). Lastly, an application layer at the top of the architecture enables the development of different IIoT applications.

3.1. Infrastructure layer

3.1.1. Field devices

The concept of field devices makes Ethernet even more appealing. It is made up of several types of IIoT devices such as robots, smart machines, conveyor belts, solar panels, SolarEdge, sensor–actuator, etc. Field devices generate a large amount of data that can be sensed and gathered by sensor–actuator and other advanced devices. To be specific, the broadly used Radio Frequency Identification (RFID) technology for effective data sensing with effortless (Song et al., 2023). RFID and sensor–actuator transmit collected data to the switches through gateways as the mediators. Therefore, gateways support seamless connectivity to field devices through different protocols suitable for industries such as CoAP, IPv4/IPv6, HTTP, WIA-PA, MQTT, 6TiSCH, and 6LoWPAN (Zanella et al., 2014). In a case of larger communication, Long Range (LoRA) is the best choice, because it minimizes energy consumption and bandwidth over a long area (Li and Cao, 2022).

3.1.2. Data plane

The data plane layer is committed to interconnecting the network topology components, providing unified data forwarding for the efficient exchange of information in network connectivity (Kaljic et al., 2019). In this layer, Local Area Networks (LANs) are used for sharing resources and software and every LAN has an edge server to store and analyze data locally. A cloud server is used to save a large amount of data and aggregate all IIoT data from the edge servers. Each LAN incorporates a set of specific switches and every switch is connected to the local SDN controller through the AdaptFlow protocol. LANs are connected using dedicated industrial wired Ethernet protocols, comprising, for instance, one or more of the following protocols: PROFINET, Modbus, EtherNet/IP, and EtherCAT.

3.1.3. IIoT gateways

IIoT gateway nodes (Liu et al., 2022) are deployed at the infrastructure layer as middleware between field devices and switches. Gateways transform packets coming from actuators and sensors into frames with a format that is accessible and able to be manipulated by switches. Furthermore, the gateway processes each actuator–sensor flow differently based on the network reliability required and helps to identify a particular actuator–sensor data stream. Consequently, this provides an efficient performance of different actuator–sensor clusters having various requirements regarding packet loss and deadlines. In addition to the SDN switches, gateways can also be configured and managed by the SDN controller or IIoT controller. Therefore, different functions and control logic can be deployed inside the gateways to reduce the controller overhead and round-trip time. Moreover, gateways can keep local data or manipulate data according to the rules enforced by the controller.

3.1.4. IBN

IBN overcomes the routing issues that arise when the industrial field network accesses the IPv6/IPv4 communication protocol or Internet to enhance the cross-network speed. This was intended to help support deterministic routes for data forwarded in the industrial system. Once an industrial service establishment demand is received, the controller can deploy the industrial service instantly without requiring additional management applications. In order to flexibly manage the bandwidth allocation inside the IBN, the SDN controller can install additional bandwidth for reducing end-to-end latency and guarantee delay-sensitive for critical services. For non-critical services, the SDN controller can install lower bandwidth to keep more bandwidth for industrial critical services. Further details are described in our patent archived in Wang et al. (2019a).

3.2. Control plane layer

The control plane layer is an intermediate layer in the SDN architecture that needs a high interface to organize and facilitate information sharing among the other layers. The control layer cooperates with distributed network management (Firouzi and Rahmani, 2022; Golightly et al., 2023). It means, every local controller manages the network nodes installed on its related LAN, whereas the global controller focuses on managing the local controllers deployed in the system. This tactic is also applied in Edge-based SDN-6LoWPAN termed as SDN-6LE (Das et al., 2020), where the control plane layer is separated into dual sections, a global controller and an edge controller. In our SDN-IIoT architecture, the local controllers are applied to minimize the latency and facilitate the interoperability among devices. From this perspective, LAN switches do not always communicate or dispatch request to the global controller; rather, they acquire direct and fast feedback from the local controller. The global controller merely needs to participate in case the local controller acquires an unmatched flow at the first occasion and period to install the new rules inside the local controllers. This strategy primarily supports to minimize controller overhead and achieve delay-sensitive. Accordingly, the IIoT control system can remotely monitor different sections or departments with diverse communication requirements. For instance, one can consider a manufacturing factory with an assembly lines department and an energy resources department placed in different zones. Thus, dual LANs and their dual corresponding local SDN controllers would be applied, one for the assembly lines department and another for the energy resources department. In this way, many requests from switches deployed inside the LAN could be processed locally so as to minimize the latency and the number of round-trips at the global SDN controller. The global SDN controller installs the policies in the local controllers as instructed by the IIoT controller according to the QoS needed by IIoT applications and the requirements of each department. This could comprise dissimilar network rules for the assembly lines department and the energy resources department. SDN controllers communicate together through east/westbound APIs to fix the network performance. AI is installed inside the SDN controllers, to provide the intelligent decision-making capabilities to the SDN-IIoT system. Therefore, SDN controllers can be the smart entities, and complicated SDN-IIoT scenarios and services can be effectively managed (Alhilali and Montazerolghaem, 2023). In the same token, the AI modules can support the system to predict bottlenecks before they arise using a time-based analysis for incoming data. If the bottleneck is detected, the system would be capable to swap the hierarchy of SDN controllers in order to offload some of the responsibilities to the global SDN controller. In the SDN-IIoT architecture, DHT is deployed in the memory of every local SDN controller and acts as the internal flow table (Josbert et al., 2021c). DHT stores the flow rules corresponding to the optimum routes. The main advantages of utilizing DHT are the fastest lookup and rerouting according to the network state change, as well as minimizing the computation time of a new route.

3.3. Application plane layer

The application plane layer contains the IIoT controller and different IIoT applications that explicitly and accurately send their needs on network performance and diverse IIoT requirements to the control layer using a northbound API (e.g., REST API). Moreover, this layer offers suitable guidance to the control layer based on the current network status and statistics.

3.3.1. IIoT controller

IIoT controller decides the number of gateways to avoid data loss for mission-critical scenarios. For example, a scenario that requires loss-sensitive and high reliability will activate more gateways compared to a scenario that does not require high reliability. The IIoT controller is also capable to determine the energy consumption of the actuator–sensor in diverse forms of transmission, reception, waking, and sleeping. Therefore, the IIoT controller must set an appropriate schedule within the limits of the available resources for the actuator–sensor.

3.4. Lesson learned: Recap and insights

This section presents a case study based on an SDN-IIoT architecture consisting of three main layers: the infrastructure layer, the controller layer, and the application layer. Data from the field devices can be collected by smart nodes such as sensors, actuators, and RFID. In the communication, some of these smart nodes positioned in the field devices can utilize deterministic networking and low-power wireless protocols like WIA-PA, 6TiSCH, 6LoWPAN, etc. Data plane switches are deployed in the IBN, which is regulated by the control plane to achieve flexible configuration and systematic communication. These data plane switches are connected to the control plane through the AdaptFlow API. AdaptFlow is chosen due to the reduction of latency and energy utilization while increasing network capacity compared to the OpenFlow API. The control plane layer is based on the hierarchical distribution of SDN controllers to avoid the bottleneck and a single point of failure, as well as enhance the interoperability range. Hence, every local controller manages the network devices deployed on its corresponding LAN, while the global controller emphasizes the management of local controllers. DHT is installed in the memory of each local controller in order to minimize the calculation time of the new optimal route and the lookup time used to select the new flow rules at the control plane level. AI technology was used for the control plane side to make better decisions, minimize false positives, improve self-driving, detect network anomalies, and so on.

Distributed edge servers have limited resources compared to the central cloud server; thus, an AI decision-making model can be used to clarify the data, which can be manipulated at the edge level and

those that should be shifted at the cloud level. Moreover, data filtering can be used before shifting the data to the cloud in order to decrease resource consumption. Standardization of REST API and AdoptFlow API is necessary to enhance communication and interoperability of services. Furthermore, using multiple controllers may lead to high performance and minimize communication delay in SDN-IIoT. In this context, east/westbound APIs require standardization and further improvement.

An efficient synchronization between the control plane and the data plane is essential to boost the network control in the SDN-IIoT architecture. Additionally, the hierarchical control plane architecture may use different types of controllers. Due to this, the control plane layer may utilize multi-vendor solutions, and therefore, standardization of interfaces is a key area of research. Interoperability with other domain controllers can also be further analyzed for efficient communication. Ref. Yin et al. (2012) has done interesting work in this matter which may be a promising starting point.

4. Advanced and standard flow installation techniques based on SDN in the IIoT environment

In this section, we analyze seven flow installation techniques based on SDN technology, from which the IIoT network administrator can select the appropriate flow installation technique applied according to different scenarios and requirements of IIoT services. The selected technique can be inserted inside the SDN controller, and then the SDN controller can also insert this technique in the switches as flow rules.

4.1. Proactive flow installation (PFI) technique

Based on L1-Norm optimization to compute the shortest path, the authors in Ahmed et al. (2018) propose a PFI technique in industrial automation (see Fig. 8). In the PFI technique, the SDN controller inserts the predetermined flow rules to the switches across the main route immediately after identifying them. The flow rules used in the flow table of a switch have the timeout value or lifetime interval. Once the flow rules expired, the switch has to demand the SDN controller for new flow rules in order to transmit the received packets according to their type. This process corresponds to the `PACKET_IN/PACKET_OUT` action in OpenFlow (Pfaff et al., 2012). The SDN controller transmits new flow rules based on a defined interval (its default time is 30 s (Pfaff et al., 2012)). That is why the switches have to wait for the remaining time of the specified interval to receive the new flow rules so as to update the flow tables. If the network state changes after preconfiguring flow rules, the performance of the destination side may be affected by the change. For instance, if the cost of the shortest route increments or if a malfunction of any component across the shortest route occurs, the switch cannot make dynamic forwarding decision in accordance with the change at this point as long as the SDN controller resends the new flow rules.

4.2. Proactive flow installation redirecting (PFIR) technique

To solve the above challenge, the PFIR technique is proposed in our paper stored in Josbert et al. (2021c). In this PFIR technique (see Fig. 9), the new flow rules are reconfigured if the change is identified in the connectivity or if the SDN controller gets the demand to swap the traffic type (e.g., exchange the traffic type from delay-sensitive to delay-sensitive and loss-sensitive or vice-versa) rather than waiting for the timeout value to expire. At that moment, update flow rules are configured according to the change. For instance, after discovering that the cost of the optimum route is incremented, the SDN controller calculates the new optimum route and inserts the flow rules inside the flow tables corresponding to the switches along this new optimum route. Moreover, when the traffic type swaps, the new optimum route is chosen in the DHT which is deployed in the SDN controller.

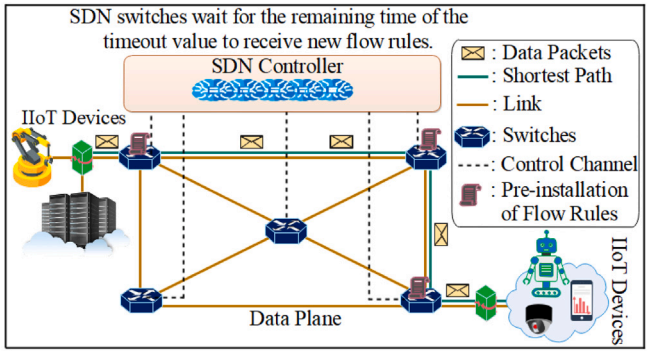

Fig. 8. PFI network scenario.

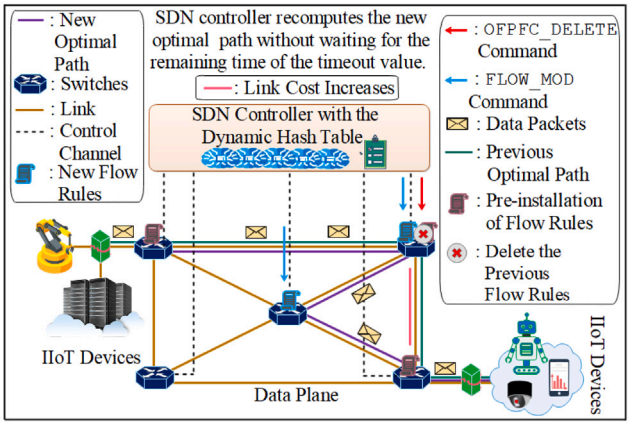

Fig. 9. PFIR network scenario.

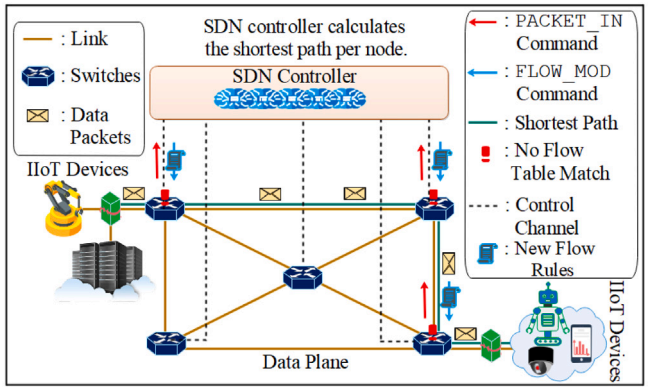

Fig. 10. RFI network scenario.

4.3. Standard reactive flow installation (RFI) technique

Fig. 10 shows the RFI technique in the industrial network. In this technique, flow rules can be installed on demand. Data can arrive at the source switch where no matching rule is found inside the flow table. As a result, the source switch is unable to forward the data on its own and it notifies the SDN controller through the `PACKET_IN` command. After that, the SDN controller computes the shortest route for the data and inserts the flow rules using the `FLOW_MOD` command to the switch (Pfaff et al., 2012). These processes are correspondingly repeated in each switch along the shortest route till the data arrive at the destination. As consequence, the RFI technique increases latency and cannot meet some of the requirements of IIoT, especially delay-sensitive.

Table 3
A comparison of flow installation techniques.

Ref.	Type	- Speed - Range (ms)	CD	DE/L	PLR	CI	PVR	SR	- CPA - CN	Evaluation	Performance	D-S	L-S	MC
PFI (Ahmed et al., 2018)	Proactive	High 0,1 – 2	×	✓	×	✓	×	✓	C Ryu	Simulation Implementation	Static	✓	×	Medium
PFIR (Josbert et al., 2021c)	Proactive	High 0,1 – 2	✓	✓	✓	✓	✓	×	C ODL	Simulation Testbed	Dynamic	✓	✓	High
RFI	Reactive	Low 10 – 80	–	–	–	–	–	–	D or C –	–	Dynamic	×	–	Low
HFI (Ahmed et al., 2018)	Hybrid	Medium 4 – 15	×	✓	×	✓	×	✓	C Ryu	Simulation Implementation	Dynamic	×	×	Medium
MFI (Josbert et al., 2021c)	Hybrid	Medium 3 – 10	✓	✓	✓	✓	×	×	C ODL	Simulation Testbed	Dynamic	✓	✓	High
IPFI (Josbert et al., 2021a)	Proactive	High 0,1 – 2	✓	✓	✓	×	×	×	C ODL	Testbed	Static	✓	✓	Medium
RPFI (Josbert et al., 2021a)	Hybrid	Medium 4 – 15	✓	✓	✓	×	×	×	C ODL	Testbed	Dynamic	×	✓	Medium

Ref.: Reference; CD; Cloud Domain; DE: Delay; L: Latency; PLR: Packet Loss Rate; CI: Confidence Interval; PVR: Packet Violation Rate; SR: Success Rate; CPA: Control Plane Architecture; CN: Controller Name; D-S: Delay-Sensitive; L-S: Loss-Sensitive; MC: Memory Consumption; ✓: Considered; ×: Not considered.

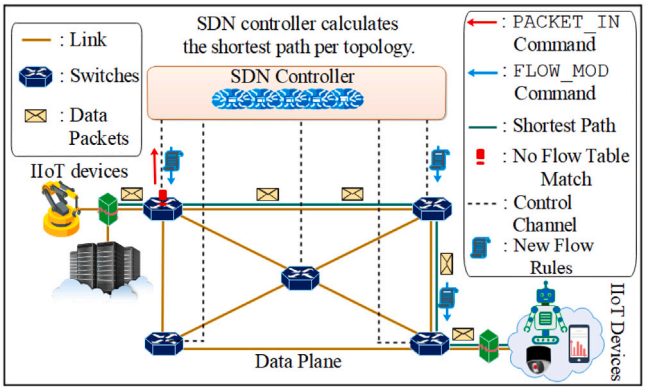

Fig. 11. HFI network scenario.

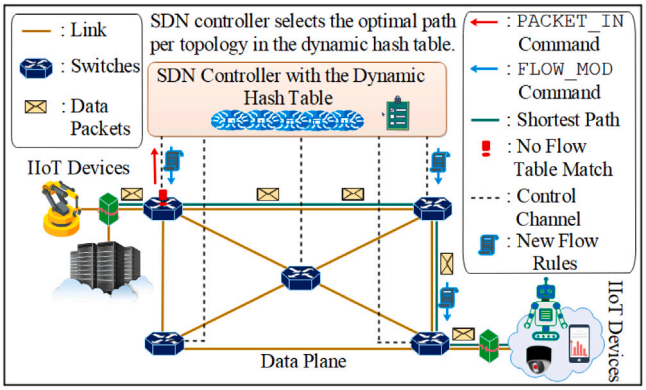

Fig. 12. MFI network scenario.

4.4. Hybrid flow installation (HFI) technique

Fig. 11 shows the HFI technique for industrial automation (Ahmed et al., 2018). In case a matching flow entry is found, the rules related to a precise flow entry are executed. If there is no match found in the switch flow table. At that moment, the controller receives a `PACKET_IN` command from the switch and then it calculates the shortest route to reach the destination; and later transmits simultaneously `FLOW_MOD` messages to every switch across this shortest route. The switches also keep the flow rules received inside the flow tables so that the next similar data streams are forwarded without notifying the controller. This tactic of caching flow rules in flow tables would reduce both controller overhead and latency.

4.5. Mixed flow installation (MFI) technique

Fig. 12 shows our proposed MFI approach to minimize the delay caused by RFI and HFI techniques, as well as taking into account dynamic performance according to the present network state (Josbert et al., 2021c). If there is no match in the switch, the SDN controller receives `PACKET_IN` command as a notification message from this switch. Consequently, a new optimum route is chosen inside the SDN controller based on the dynamic pre-determined flow rules stored in the DHT in order to achieve a fast lookup and minimize the computation time of the new optimum route.

4.6. Improved proactive and reactive flow installation techniques

In our paper stored in Josbert et al. (2021a), an optimal routing algorithm based on a polynomial-time model is applied to calculate the optimum path by considering both the delay-sensitive and the loss-sensitive. By making use of this optimum path to forward data packets, two flow installation solutions are put forth: Improvement of the PFI (IPFI) technique and the integration of Reactive and PFI (RPFI) techniques. Indeed, the main difference between IPFI and PFI is that IPFI considers both delay-sensitive and loss-sensitive while PFI considers only the delay-sensitive. This is also the main difference between RPFI and MFI. A comparison of flow installation techniques is provided in Table 3.

4.7. Lesson learned: Recap and insights

In this section, we provide seven types of flow installation techniques based on SDN technology. The first is the PFI technique which is founded on the preconfiguration of flow rules. PFI achieves the latency-sensitive but is static, since it cannot be changed until the timeout value expires. The second is PFIR which is also founded on the preconfiguration of flow rules. Contrary to the PFI technique, the PFIR technique is capable to reconfigure the network without awaiting the timeout interval to expire. However, it consumes additional memory for the controller side. The third is RFI which is based on dynamic performance and installed flow rules on demand. The main drawback of RFI is that it consumes additional time and cannot meet the latency-sensitive requirement. The fourth is the HFI technique which relies on the combination of PFI and RFI techniques. The fifth is MFI which relies on caching flow rules inside the DHT deployed in the controller memory. The sixth is IPFI which extends PFI by considering both loss-sensitive and delay-sensitive requirements. The last one is RPFI which also extends HFI by taking into account both loss-sensitive and delay-sensitive requirements. A comparison of all flow installation techniques

is provided in Table 3 where it is clear that more considered metrics than others are delay/latency and PLR. Thus, there are still different metrics useful for IIoT that should be considered in future work such as jitter-sensitive and energy consumption. Currently, the majority of SDN switches are not pure OpenFlow switches. Instead, most of them are typically hybrid switches, which are just traditional networking nodes with the capacity of communicating via OpenFlow protocol. However, the existing flow installation techniques consider pure SDN switches for installing flow rules. Therefore, there is a need for developing new flow installation techniques well compatible with a hybrid network consisting of both SDN switches and legacy networking nodes. Moreover, an effective realization of flow installation techniques needs to consider the requirements of both IIoT and SDN to significantly enhance the performance of SDN-IIoT infrastructures.

5. Fault tolerance management in SDN-IIoT

This section describes fault tolerance or resilience approaches that can be used to recover network connectivity when a failure occurs in SDN-IIoT. Furthermore, it presents a case study based on an SDN-IIoT fault tolerance architecture and its simulation results.

5.1. Fault tolerance approaches

IIoT services demand diverse requirements to ensure efficient performance. For instance, by forwarding data under conditions of high reliability, loss-sensitive, and latency-sensitive. The capability to deal with these requirements relies on suitable support from the network configuration approach, dynamic reconfiguration according to the change, and fault-tolerance applications. As the communication network becomes an important element of SDN-IIoT, its high availability and resilience must always be guaranteed (Josbert et al., 2021b). Thus, it is necessary to design network connectivity that includes the fault tolerance approach to detect and recover from any broken network link or device (Rehman et al., 2019). The SDN-IIoT system must identify any type of network connectivity failure and the problem should be reported to the SDN controller (Zurawski, 2014). A network connectivity failure can be either unintentional (e.g., unplanned) for several reasons, and some of them include natural disasters, hardware failures, overloading interface, software errors, cable/wire cuts, and human fallacies are part of the daily life of operational teams; or intentional (e.g., planned) generated by the network maintenance procedure (Rinaldi et al., 2018). In general, there are three types of fault tolerance approaches. The first is the restoration approach which relies on the reactive flow installation technique. The second is the protection approach which relies on the proactive flow installation technique. Lastly, the hybrid approach which is based on the combination of restoration and protection approaches (bin Salleh et al., 2023).

5.1.1. Restoration approach

For instance, Ref. Al-Rubaye et al. (2017) proposes an SDN-based resilience paradigm to guarantee smart grid reliability. In line with this, dynamic path restoration has been utilized if an intermediate link fails. Diverse scenarios such as network expansion and failure are presented to evaluate the practicality of the designed paradigm. Aloe as a distributed and auto-scalable orchestration mechanism is proposed (Chattopadhyay et al., 2020). Aloe analyzes in-network performance-based mechanism through several lightweight controller instances rather than service-grade controller apps. This mechanism guarantees reliability and minimizes the latency of flow-setup by embedding instances in the surrounding area of the resource-constraint IoT nodes. Aloe supports resilience and recovery from node/link failure by using a self-stabilizing controller placement model. Besides, Aloe is capable to preserve resources for micro-controllers in order to assure the QoS. In Jhaveri et al. (2021), an SDN resilience scheme is proposed based on self-reconfigurability and self-optimization to manage the dynamic bandwidth requirements of various routing flows and ensure network recovery when a failure occurs in the wired industrial network. The authors in Zhou et al. (2022) present a resilience approach that was designed for SDN-IIoT by considering fiber wireless networks based on parallel transmission and network coding. Dual adaptive algorithms: Parallel Transmission with Sparse Network Coding (PT-SNC) and Parallel Transmission with Random Linear Network Coding (PT-RLNC) are used to attain high-availability and reduce data loss posed by the malfunction of a network component and inadequate channel quality.

5.1.2. Protection approach

In Vestin et al. (2018), the authors develop a *FastReact* scheme where the control logic is deployed in the switch devices in order to do some tasks of the SDN controller. Because of that, the control loop between actuators and sensors can be minimized, since data transmitting decisions are generated closer to actuators and sensors. This tactic decreases the delay and the number of round-trips between data plane devices and the control plane. The control logic is defined through conditionals containing Boolean Logic, which are then installed at the data plane level using the SDN controller, as directed by the industrial controller. *FastReact* uses a protection approach to recover from a failure. The same authors (Vestin et al., 2015) also utilize packet duplication coordinated by the control plane to decrease data loss with minimum latency when a link failure occurs. However, transmitting duplicate packets consumes additional bandwidth and can cause network congestion. Ref. Babiceanu and Seker (2019) introduces a fault-tolerant approach that contains both resilience and cybersecurity ontology to provide the security and reliability requirements of manufacturing applications.

The theory of Shared Risk Link Groups (SRLGs) supports the management of several types of failures, such as single and multiple links/switches failure. SRLG is especially useful to develop path protection techniques because it clarifies rerouting scenarios. In this way, this theory is applied to calculate redundant paths as the primary and backup non-overlapping paths among nodes in IoT networks (Bakhshi Kiadehi et al., 2021; Kiadehi et al., 2021).

5.1.3. Hybrid approach

The route restoration technique increases the recovery time dissimilar to the route protection technique which attains fast recovery. However, the route protection utilizes more flow entries in the switches, which, in return, increments the lookup delay used to discover a matching flow entry. This may affect the end-to-end latency when there is no breakdown (normal circumstances). In order to balance the two techniques, we propose a Mixed Fast Resilience (MFR) technique to ensure the quick recovery of the main route without any negative impact on the end-to-end latency if there is no failure (Josbert et al., 2021d). This work is extended in Josbert et al. (2021b) by considering both delay-sensitive and loss-sensitive as the main metrics for industrial networks. Ref. Savaliya et al. (2021) uses different ML models including the K-nearest neighbors, decision tree, logistic regression, and support vector machine so as to predict network congestion and link breakdown, as well as to recover from the network failure. Tables 4 and 5 show comparisons of current research related to fault tolerance in SDN-IIoT.

5.2. A case study based on an SDN-IIoT fault-tolerance architecture

An SDN-IIoT fault tolerance architecture is made up of three layers as depicted in Fig. 13. The SDN-IIoT infrastructure layer contains field devices and data plane components. The SDN-IIoT control layer contains two SDN controllers. Finally, the SDN-IIoT application layer enables the development of various IIoT applications, especially the fault tolerance application.

Table 4
A summary of current papers focused on fault tolerance.

Ref.	Type	Evaluation	Environment	FRS	CD	Delay/L	PLR	Throughput	RT	LT	CPA	CN	Performance
Al-Rubaye et al. (2017)	Restoration	Simulation	SDN-IIoT	Low	×	✓	×	×	×	×	C	ODL	Dynamic
Chattopadhyay et al. (2020)	Restoration	Testbed	SDN-IoT	Medium	✓	✓	×	✓	✓	×	D	ODL, ONOS, ZeroSDN, Ryu	Dynamic
Vestin et al. (2018)	Protection	Prototype Testbed	SDIN	High	×	✓	×	×	✓	×	C	SDNC (NS)	Static
Vestin et al. (2015)	Protection	Simulation	SDIN	High	×	✓	✓	×	×	×	C	SDNC (NS)	Static
Jhaveri et al. (2021)	Hybrid	Testbed	SDIN	Medium	×	✓	×	✓	×	×	C	Ryu	Dynamic
Babiceanu and Seker (2019)	Protection	×	SDN-IIoT	–	×	×	×	×	×	×	×	×	Static
Bakhshi Kiadehi et al. (2021)	Protection	Simulation	SDN-IoT	High	×	✓	✓	×	×	×	D	ODL	Static
Zhou et al. (2022)	Restoration	Simulation	SDN-IIoT	Low	✓	✓	×	✓	×	×	C	SDNC (NS)	Dynamic
Josbert et al. (2021b)	Hybrid	Simulation Testbed	SDIN	Medium	×	✓	✓	×	×	✓	D	ODL	Dynamic
Savaliya et al. (2021)	Hybrid	Simulation	SDN-ICPS	Medium	×	×	×	×	×	×	C	Ryu	Dynamic

Ref.: Reference; L: Latency; FRS: Failure Recovery Speed; CD: Cloud Domain; PLR: Packet Loss Rate; RT: Response Time; LT: Lookup Time; CPA: Control Plane Architecture; CN: Controller Name; D: Decentralized; C: Centralized; ICPS: Industrial Cyber–Physical System; SDNC: SDN Controller; NC: Not Specified; ✓: Considered; ×: Not considered.

Table 5
Advantages and limitations of current papers focused on fault tolerance.

Ref.	Advantages	limitations
Al-Rubaye et al. (2017)	- Dynamic rerouting. - Network expansion.	- Increases the recovery time. - The paradigm is not deeply demonstrated.
Chattopadhyay et al. (2020)	- No performance bottleneck near the control plane. - Reduces the flow-setup delay.	- Complicated to manage the deployment of multiple micro-controllers at the control plane level.
Vestin et al. (2018)	- Local fast control actions. - Deployed the control logic inside switches.	- Many functions at the data plane level.
Vestin et al. (2015)	- Fast recovery. - It guarantees loss-sensitive.	- It requires more flow rules inside the switches. - The duplicate packets strategy leads to network congestion.
Jhaveri et al. (2021)	- Adapts to different network situations according to the change.	- Self-reconfigurability is not demonstrated. - Self-optimization is not demonstrated.
Babiceanu and Seker (2019)	- It combines network security and resilience.	- Does not consider the implementation or the simulation of the suggested framework.
Bakhshi Kiadehi et al. (2021)	- Fast failover.	- It requires configuring more flow rules in switches.
Zhou et al. (2022)	- Considers the fault tolerance of both the wired and wireless segments.	- Higher cost and complicated due to the use of parallel transmission.
Josbert et al. (2021b)	- Dynamic fast recovery. - No negative impact on the end-to-end delay before any failure occurs.	- Consumes additional memory in the SDN controller.
Savaliya et al. (2021)	- Uses several ML algorithms to improve decision-making in the SDN controller.	- Consumes more TCAM memory in order to store all the paths that have the lowest probability of experiencing link failure or congestion.

5.2.1. SDN-IIoT infrastructure layer

Field devices as one of the parts that comprise the infrastructure layer can use the 6LoWPAN IoT protocol to improve communication with low energy consumption. 6LoWPAN is practicable in SDN simulation software as this IoT protocol works with the Mininet-IoT simulator (Setiawan et al., 2021). Many architectures in the literature review lack a global clock to perform the latency measurements. Network Time Protocol (NTP) can be used to maintain the network devices synchronized (Hou et al., 2022). The synchronization time is set to a comparatively slight number to decrease the effect of clock drift. Jitter is also reduced by giving the highest priority to NTP transactions. Moreover, the switches deployed in the IBN comprise the internal addresses. Every source switch uses a Network Address Translation (NAT) to connect the interface with the external network. It is also vital to utilize IPTable policies to share the data between the external network and the internal network. The File Transfer Protocol Server (FTP Server) can be applied to forward data packets to the cloud server. In order to reduce the lookup time, it is important to hold a specific number of sensor–actuator IDs in the memories of the gateways. So, the gateways know how many sensor/actuator IDs are activated.

5.2.2. SDN-IIoT control layer

This layer is made of double SDN controllers. One of them is controller A which is responsible to control the wireless part of the network. As instructed by the IIoT controller, controller A decides on how many gateways are activated to avoid data loss in case of network malfunction. On the other hand, controller B is in charge of controlling the data plane switches and wired routing performance according to the change, as well as dealing with fault management in this part. Besides, this controller calculates the main routes and the backup routes from gateways to the destination. In the experiment (see Fig. 14) (Josbert et al., 2021b), using two main routes with two gateways decreases the data loss than one main route with one gateway when a network failure occurs. Correspondingly, the data transmitted from the source to the destination via both gateways (A and B), where A and B refer to separate routes and AB is their collection, the packet was transmitted

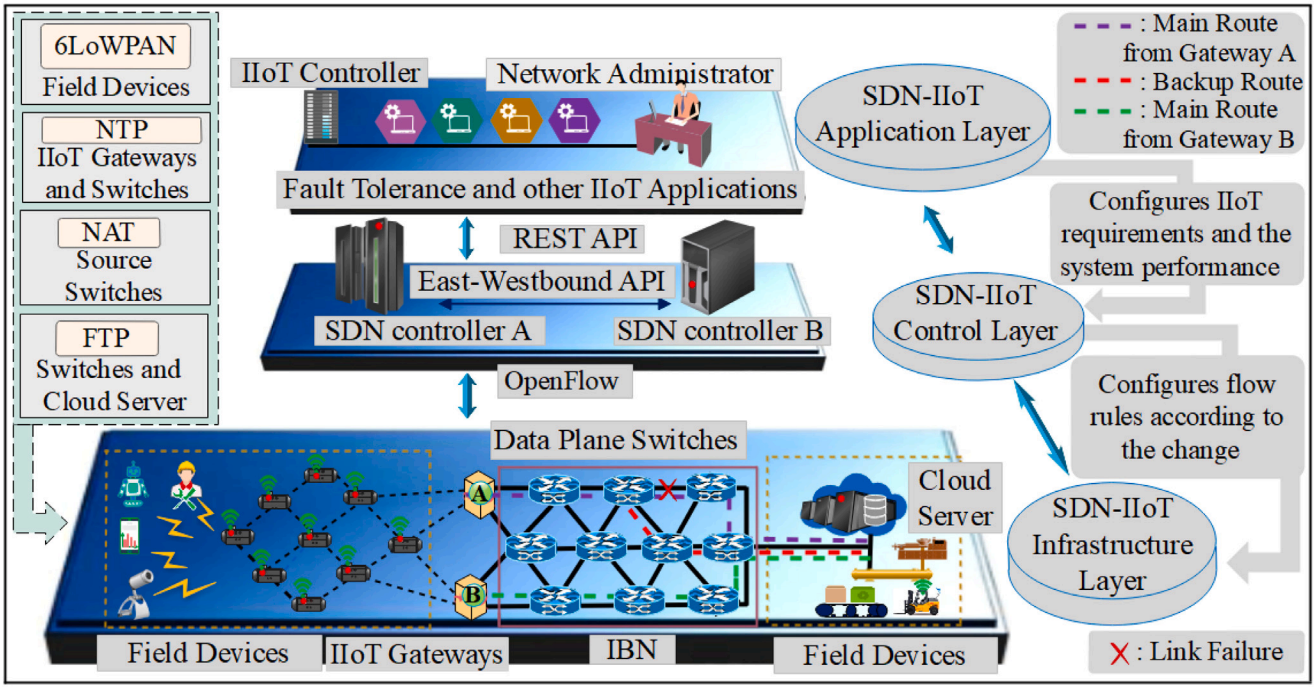

Fig. 13. SDN-IIoT fault tolerance architecture.

more successfully than through a single gateway with a single main route.

5.3. Lesson learned: Recap and insights

This section analyzes fault tolerance management which typically relies on three types of fault tolerance methods. The first is the restoration method, the second is the protection method and the third is the hybrid method. The restoration method runs dynamically based on the current network status. However, it increases the recovery time due to the calculation of the backup route after a network failure. On the contrary, the protection method achieves fast recovery by preconfiguring both primary and backup routes inside the switches before a failure occurs. Nonetheless, this method utilizes more flow entries in the switch which consume more memory and affect the lookup time in a normal situation. The hybrid method is the combination of the restoration method and the protection method. For instance, some hybrid methods store the primary route inside the switches and store the backup route in the SDN controller in order to reduce the recovery time without affecting the performance when there is no breakdown, while others use ML algorithms to predict the failure and recover network connectivity. Many fault tolerance methods are concentrated on the wired network of the data plane layer. Thus, further research efforts are needed to develop fault tolerance methods based on the wireless network, especially for the side of field devices. Additionally, many works consider a single link/switch failure. However, an SDN-IIoT environment is naturally vulnerable to external issues like fires, storms, and other unavoidable natural disasters. In this situation, there may be multiple link/switch failures with or without disjoint backup routes which require a resilience recovery solution. Therefore, there is a need to consider the occurrence of multiple concurrent link/switch failures, both for the network connectivity with and without backup disjoint routes. The work referenced in Balasubramanian et al. (2023) has made significant contributions to this issue and could serve as a starting point.

Normally, SDN-IIoT consists of heterogeneous nodes, which are mobile or static in nature. The design of resilience frameworks for the side of mobile nodes should be more addressed, while considering efficient and fast recovery in the presence of mobile nodes failure. In this regard, handover, path reconfiguration, and scalability issues should be further investigated. Table 4 displays a comparison of current research related to the fault tolerance management in SDN-IIoT. As shown in this table, none of the fault tolerance methods attempted to consider the energy consumption metric. Thus, future research can be directed in implementing new resilience approaches by considering energy efficiency. In this section, we also provide a case study based on an SDN-IIoT fault tolerance architecture (see Fig. 13). As depicted in the simulation results of this architecture (see Fig. 14), using double main routes with double gateways greatly reduces the PLR when a failure occurs.

6. Optimization of traffic routing in SDN-IIoT

The section here, describes traffic routing optimization using SDN technology to guarantee QoS requirements of IIoT. Based on different technologies (e.g., edge/fog/cloud computing, adaptive computing, etc.), we also present routing optimization by considering resource management.

To begin, routing is one of the main functionalities of communication. The SDN controller regulates the routing performance of traffic flow by inserting the rules in the flow tables of network devices like switches and routers. The SDN controller can instruct the network devices to forward the data through computed or selected routes, or it can also make a decision to drop a certain traffic type. Moreover, incompetent choices related to traffic routing can hurt communication performance in terms of packet loss, high latency, and network overload (Zhang and Yahya, 2023). There are different categories of algorithms utilized for improving routing such as Shortest Path First (SPF), heuristic, and so on. SPF is simple to implement, however, it does not improve the utilization of network resources (Ahmed et al.,

Fig. 14. Results for Packet Loss Rate (PLR).

2018). Conversely, heuristic models make better use of network resources, but with a high computational cost (El-Hefnawy et al., 2022). Furthermore, ML is considerably applied to solve the challenge of routing optimization based on decision-making theories. Therefore, ML algorithms like Reinforcement Learning (RL) successfully facilitate routing optimization (Younus et al., 2021; Almasan et al., 2022).

6.1. Edge computing in SDN-IIoT

Concerning routing for network communication, the authors in Das et al. (2020) propose an SDN-6LE architecture for guaranteeing optimum routing of data among the network devices and reducing latency. It also utilizes an edge-based computational ability to enhance connectivity performance. To deal with heterogeneity, a hybrid-edge switch is designed to provide the interface between SDN entities and the 6LoWPAN protocol. In Desai et al. (2022), the authors suggest triple routing approaches for SDN-IIoT as follows: Minutest Interval-based Optimum Path Approach (MIOPA), Cluster-enabled Optimum Path Approach (COPA), and K-means-enabled Optimum Path Approach (KOPA). These approaches reduce the network graph to enhance the effectiveness of determining the best route between devices. Moreover, edge computing servers are integrated into the system to further minimize latency. This work outperforms the Dijkstra's algorithm (Jiang et al., 2014) and the Adaptive Transfer-based energy Optimization (ATOP) mechanism (Li et al., 2018) due to the reduction of the network graph using clustering models.

6.2. VERO, SWAY, ROSA, and SEQOS frameworks

6.2.1. VERO

An SDN solution for the IoT platform called VERO-SDN is proposed to offer programmable routing control and decrease SDN controller overhead by using inherent flexibility to support an OpenFlow protocol and a long-range control channel. VERO-SDN meets the issue of unreliable communication with the SDN controller that exists in multiple Software-Defined-WSN (SD-WSN) domains. Furthermore, VERO-SDN provides an efficient exchange of information in a long-range of IoT network scenarios (Theodorou and Mamatas, 2020).

6.2.2. SWAY

SWAY (Saha et al., 2018), is a routing scheme based on a greedy model that uses the Yen's K-shortest route algorithm (Guck et al., 2017) to determine the optimum routes for transmitting the data and solve the NP-hard problem posed by the consideration of loss-sensitive metric. SWAY takes into account several QoS requirements of diverse traffic flows and the impact of SDN rule-capacity on switches. However, SWAY

Table 6
A summary of existing articles focused on traffic routing optimization.

Ref.	Algorithms and techniques	HN	5G	IIoT/IoT P	CD	Delay/L	PLR	EC	Thr.	Band.	CPA CN	Evaluation	Environment
Das et al. (2020)	- Path optimization - Routing path selection	✓	×	LoWPAN	Edge	✓	✓	×	✓	×	D ODL, Ryu, Floodlight, ZeroSDN	Testbed	SDN-IoT
Theodorou and Mamatas (2020)	- Dijkstra's shortest path - Topology discovery - Flow establishment	×	✓	RPL	×	✓	×	×	×	×	C VERO-SDN	Simulation	SDN-IoT
Desai et al. (2022)	- Dijkstra's shortest path - KOPA, - COPA, - MIOPA	×	×	×	Edge	✓	×	×	✓	×	C ODL, Ryu	Simulation	SDN-IIoT
Ouhab et al. (2020)	- Reinforcement learning (Q-routing), - DAG, - Multi-Hop Clustering	×	×	RPL	×	✓	×	✓	×	×	C ONOS	Simulation	SDN-IoT
Municio et al. (2020)	- Scheduling - Distributed routing	✓	×	6TiSCH, RPL, 6LoWPAN	×	✓	×	✓	×	×	D ONOS	Testbed	SDN-IIoT
Saha et al. (2018)	- Greedy heuristic - Yen's K-shortest path	✓	×	×	Fog	✓	×	×	×	×	C POX	Simulation	SDN-IoT
Bi et al. (2019)	- SPMC, - KPF - Hybrid network search path - OSPF mode	×	×	×	×	×	×	×	×	×	C SDNC (NS)	Simulation	SDIN
Njah and Cheriet (2021)	- Lagrangian relaxation - Dijkstra's shortest path - Deterministic networking-routing - Scheduled-routing	✓	×	×	×	✓	✓	✓	×	✓	C Floodlight	Simulation	SDN-IIoT
Naeem et al. (2020b)	- Max-flow min-cost - Yen's K-shortest path	✓	×	×	Cloud	✓	×	✓	×	×	C POX	Simulation	SDN-IIoT
Li et al. (2022b)	- Joint route - Power selection optimization - Multi-Armed Bandit	×	×	×	×	✓	×	✓	×	×	C SDNC (NS)	Simulation	SDN-IIoT

Ref.: Reference; HN : Heterogeneous Network; P: Protocol; CD: Cloud Domain; L: Latency; PLR: Packet Loss Rate; EC: Energy Consumption; Thr.: Throughput ; Band.: Bandwidth; CPA: Control Plane Architecture; CN: Controller Name; D: Decentralized; C: Centralized; SDNC: SDN Controller; NC: Not Specified; DAG: Directed Acyclic Graph; OSPF: Open Shortest Path First; ✓: Considered; ×: Not considered.

uses an additional number of link activation due to the calculation of optimum routes for both loss-sensitive and latency-sensitive flows. The rule-capacity constraints of the data plane switches also lead to incremented link activation. This challenge is addressed in Bi et al. (2019), where the maximum link activation is minimized while also considering both loss-sensitive and delay-sensitive. To enhance the QoS of industrial applications, this work developed a mixed network search route mechanism for intelligent traffic transmission, which uses a Single Path Minutest Cost (SPMC) model to determine the route crossing to the low-load area and a K-Path Forwarding (KPF) model to find several routes for data transmission.

6.2.3. ROSA

ROSA is an SDN framework-based parallel transmission optimization (Njah and Cheriet, 2021), which takes into account the traffic type, QoS requirements, energy efficiency, and network resource utilization. A centralized multi-program paradigm based on the Lagrangian relaxation algorithm is presented to solve the NP-hard problem. ROSA is composed of three models. One model is developed to guide a parallel routing method, while two other routing models are developed to control the deterministic networking and scheduled types of traffic flows.

6.2.4. SEQOS

Like in SWAY and ROSA, SEQOS (Naeem et al., 2020b) also takes into account different QoS requirements of IIoT traffic flows (latency-sensitive and loss-sensitive) and rule-capacity constraint of SDN switches and intermediate links. In addition, SEQOS considers the jitter-sensitive that is missing in the VERO, SWAY, and ROSA frameworks, as well as choosing the best paths to efficiently use bandwidth costs and energy consumption. The work defines the problem of multi-constrained QoS-aware for parallel transmission optimization as a min-cost max-flow problem from the viewpoint of the SDN-IIoT platform-based intelligent healthcare services. SEQOS adopted the Yen's K-shortest path model which was also used in SWAY.

6.3. Multi-hop routing in SDN-IIoT

The authors in Ouhab et al. (2020) propose a routing mechanism founded on the Routing Protocol for Low-Power and Lossy Networks (RPL), and a multi-hop clustering algorithm to arrange devices in clusters and provide energy efficiency. The adjustment of cluster heads and the view of network situations are provided by SDN and the Q-routing model is utilized because of its ability to find out optimum routing rules online in a dynamic ecosystem. In Li et al. (2022b), the authors propose a high-confidence bound founded on a power selection optimization model and joint path to improve the multi-hop collaboration mode and dynamically executes the best possible synergy by cooperating with the environment. Moreover, power options and paths are amalgamated to make a group of arms in a multi-armed bandit.

6.4. Whisper in SDN-IIoT

Ref. Municio et al. (2020) presented applicable cases illustrating the merit of a central control that collectively manages the complete IIoT communication area. In this context, a higher-level solution that leverages Whisper to control IIoT and wired segments is proposed, whereas 6TiSCH is used to orchestrate the whole network segments. The experience illustrated the strategy that shifted the Whisper range from the edge to the orchestrator and developed the Whisper southbound protocol that allowed its incorporation with a standard SDN controller. Whisper appeared to remain a compromise solution that combined the scalability and robustness, as well as reduced the overhead of distributed solutions. Moreover, it improves the programmability and pliability of centralized solutions. This happens without modifying the firmware of the devices deployed in the existing IIoT system. Lastly, Table 6 presents the current literature review focused on traffic routing optimization in SDN-IIoT.

6.5. Resource management in SDN-IIoT

Service providers and network operators utilize resources management methods to improve network functionalities (Mahmoudi et al.,

2022). In this context, the SDN platform minimizes resource utilization through network-based resources management in diverse fields such as 5G mobile networks (Tadros et al., 2020), edge/fog/cloud computing (Lv and Xiu, 2019; Bedhief et al., 2019), adaptive computing, resource provisioning, Industrial WSN (IWSN), smart manufacturing, and so on.

6.5.1. Fog-edge computing in SDN-IIoT

Resource management of the data plane and the SDN in general comprise networking, computing, and caching resources. Networking resources involve bandwidth and energy, which are utilized to accomplish the QoS requirements and user Quality of Experience (QoE). Caching mechanisms allow the most often demanded data to be retained at the end node in order to extract data redundancy and minimize data forwarding latency. New technologies like augmented reality and face recognition need high computation to improve QoS and QoE. Owing to scarce resources and battery capacity, the node resources are unable to carry out computational duties properly. One solution to offload such computational duties is to deploy computing resources close to the end-users through the use of edge computing. The authors in Huo et al. (2016) propose a scheme that incorporates computing, caching, and networking to logically support computing applications and data retrieval for green wireless networks including WiMAX, WLANs, and cellular networks. Based on dynamic management and programmability from the SDN controller, it integrates the concept of data-centricity generated in Information-Centric Networking (ICN). This hybrid scheme can allow dynamic orchestration of caching, networking, and computing resources to accomplish the demands of various services.

There are multiple issues that should be solved in the communication network due to the fast development of IoT/IIoT and a large number of nodes/apparatuses connected to the Internet, which increase the complexity to guarantee security, high reliability, and deterministic latency. Based on fog computing and SDN, the authors in Sharma et al. (2017), propose a distributed blockchain paradigm to address the principles needed to powerfully manage the raw data streams generated by a large number of IoT nodes and apparatuses in the edge network and distributed cloud. This paradigm contains the following four stages: choosing resource providers, offering services, recording of transactions, and payment.

6.5.2. Cloud-edge computing in SDN-IIoT

An SDN-IIoT application-based load-balancing model is proposed (Babbar et al., 2021). This application determines several clusters that balance the load across all clusters according to the number of requests received on every cluster. When the number of requests is beyond the threshold rate assigned, then the extra requests will be transmitted to the inter-cluster from the intra-cluster due to balancing the load inside the clusters. An SDN controller is in charge of handling incoming requests and directing them to the appropriate clusters of IoT devices. The SDN-assisted Resource Management (SDN-RM) approach is proposed (Okwuibe et al., 2021). The goal of SDN-RM is to speed up resource orchestration using cloud–edge solution for resource consumption and dynamic workload balancing in order to minimize the cost of IIoT services. The work designed a Constraint Satisfaction Problem scheme on Savil Row through Essence Prime language. Moreover, the SAT Solver was utilized to install the SDN-assisted cloud–edge network prototype in the IIoT ecosystem, whereas PureEdgeSim and CloudSimSDN are adopted in the implementation. The same authors (Okwuide et al., 2020) integrate Multi-access-based Edge Computing (MEC), SDN, and container technologies for improving resource management in IIoT. Moreover, this work takes into account an offloading method that provides effective and scalable resource management for edge-containerized applications.

6.5.3. Adaptive computing in SDN-IIoT

In Wang and Li (2018), the authors analyze the adaptive optimization computing in fog computing services based on the execution of different tasks and a Computing Mode Selection (CMS) approach deployed in the SDN controller. The SDN controller determines the task computation latencies for various computation models, and it can then choose the best computation model for every task. To guarantee the low-latency in the task processing for the fog computing system, a task execution sequence adjustment paradigm is used in accordance with the task priorities. In Li et al. (2018), an adaptive communication architecture for SDN-IIoT-based edge computing is proposed. Moreover, a coarse-grained forwarding route mechanism is developed for the low-deadline scenario. With the route difference degree, a balanced forwarding route is obtained for each candidate route, and a fine-grained technique comprising an adaptive power strategy is used for emergency circumstances.

6.5.4. SD-IWSN

Several works analyzed resource management approaches suitable for different domains, such as data center-based SDN (Paliwal and Shrimankar, 2019), 6G-SDN-based NFV (Barakabitze and Walshe, 2022), SD-IWSN (Bello et al., 2020), and IoT-assisted smart homes (Jang and Lin, 2019). Particularly, in Bello et al. (2020), TSCH scheduling is presented to deal with communication scheduling and mobility of network nodes in IWSN. A clustering algorithm was used to decrease the number of links needed for Mobile Nodes (MNs) with the purpose of enhancing scalability. Clustering helps to minimize both the overhead and the number of time slots needed in the system due to the network connectivity change only disturbs two clusters (for example, the one cluster that the MN joins and another from which the MN departs). However, multiple MNs may forward data in the same shared slot, which can lead to network collisions.

Dynamicity of the industrial system needs WSNs to be capable to adapt rapidly to logical and physical modifications, and to communicate through deterministic networking. In this way, an scheme named SDN-MMF (SDN with Mobile Multicast Forwarding) is proposed (Orozco-Santos et al., 2021), which leverages the integration of SDN, WSN, and TSCH as to facilitate efficient performance of MNs and guarantee the deterministic end-to-end latency in the data transmission with the same levels as those of fixed nodes. Additionally, SDN-MMF enables a parent node change to perform promptly and clearly. Due to this, it depends a bit on traffic control and uses less energy because there are fewer shared slots.

6.5.5. 6TiSCH protocol in SDN-IIoT

The integration of 6TiSCH and SDN in IIoT provides efficient resource allocation along constrained IoT/IIoT networks, high scalability, and dynamic configuration of IoT/IIoT devices. In this context, Ref. Baddeley et al. (2017) analyzes deterministic networking based on the 6TiSCH protocol and offers efficient communication between the SDN controller and IoT-constrained devices by tackling control channel unreliability issue. To continue, Ref. Thubert et al. (2015) analyzes the applicability of deterministic networking in the centralized technique of SDN technology, and how 6TiSCH can utilize and improve deterministic networking performance for low-power WSNs. The IETF 6TiSCH Working Group is in the process of standardizing the combination of TSCH and SDN (Thubert, 2019). The above studies concentrate on designing scheduling methods based on TSCH and SDN. However, they are unable to properly support changes in network connectivity caused by mobility.

6.5.6. Enterprise resource planning (ERP) and adaptive resource provisioning in SDN-IIoT

Ref. Sahoo et al. (2021) provides a dynamic and distributed pricing approach that enhances the utility of service providers. The approach

is based on the laws of supply and demand, where prices of ERP-IoT traffic flows are selected according to the current requirement of bandwidth and accessibility of resources. The authors in Becker et al. (2019) propose an adaptive resource provisioning framework for monitoring various thresholds which automatically activate the SDN controller to redistribute resources. This framework allocates resources to various traffic flows based on current requirements and network status. To this end, a summary of articles focused on resource management is presented in Table 7.

6.6. Lesson learned: Recap and insights

This section analyzes traffic routing optimization and resource management in SDN-IIoT. In Table 6, we summarize the current works, focusing specifically on optimizing traffic routing to achieve QoS requirements of IIoT. Besides, in Table 7, we summarize the works concentrated on resource management. As depicted in these Tables, some algorithms have been adopted repeatedly, such as Yen's shortest path, Dijkstra's shortest path, and clustering. Moreover, different important metrics for IIoT have been considered in performance evaluations such as latency, end-to-end delay, throughput, PLR, bandwidth utilization, energy consumption, and so on. For further improving routing optimization in SDN-IIoT, jitter-sensitive and lookup time metrics should be further investigated in future work, as additional jitter and additional lookup time affect end-to-end delay. SDN-IIoT routing must be flexible and adapt to the dynamic changes of QoS requirements. Therefore, future studies would be directed to develop adaptive routing schemes that can dynamically adjust QoS according to the network condition changes. Furthermore, it is crucial to focus on SDN-based big data analytics to determine traffic paths and install flow rules for large-scale traffic.

In routing optimization considering resource management, we analyze several fields and technologies applied to ensure effective resource management. In this subsection, many articles have focused on resource management for the side of the data plane layer and field devices with different computing layers (edge, fog, and cloud). Therefore, further research and improvement efforts are still needed for resource management on the side of the control plane layer. The relationship between resource adaptation and resource-awareness needs to be investigated in future research. Other technologies, like NFV-based network slicing, Content-Centric Networking (CCN), and Information-Centric Networking (ICN), can be further considered to reduce the consumption of network resources. For instance, one of the crucial features of ICN is the capability to cache information at network nodes and utilize it cooperatively for information sharing across the network. Thus, ICN brings support for efficient bandwidth utilization and decreases communication latency.

7. Different essential domains in SDN-IIoT

In this section, we discuss OPC UA protocol tasks to improve interoperability in SDN-IIoT and frameworks that consider hardware implementation in the performance evaluation of SDN-IIoT scenarios, as well as SDN-IIoT-based Digital Twin (DT). Lastly, we analyze the energy efficiency and real-time in SDN-IIoT.

7.1. IIoT leveraging SDN

7.1.1. Interoperability-based OPC UA in SDN-IIoT

Due to the increase of heterogeneous devices and different network protocols used in smart manufacturing, delay and interoperability become an open challenge that complicates data collection (Wu et al., 2022; Huo et al., 2022a). One solution is proposed to address this challenge by investigating interoperability in the IIoT architecture to provide the delay-sensitive in the data collection process using dual technologies, SDN and OPC UA. Nevertheless, the majority of network nodes do not support OPC UA. To deal with this incompatibility challenge, the work uses the Edge Information Layer relying on the smart edge gateway to build the semantic data model for describing the knowledge/information of network nodes in the legacy platform (Wang et al., 2022a). In general, OPC UA is attractive in the industrial manufacturing environment due to its straightforwardness and effectiveness (Nguyen et al., 2022). With the purpose of guaranteeing network interoperability and scalability based on SDN and OPC UA client–server architecture, a plugin named IoTDM is the dedicated IoT application particularly implemented to handle and keep data produced by IoT nodes based on oneM2M standard. IoTDM is capable to support ODL northbound and southbound APIs (Romero-Gázquez and Bueno-Delgado, 2018).

7.1.2. SDN-IIoT advanced testbed

In Wan et al. (2016), the authors investigate the applicability of SDN in the Industry 4.0 environment for controlling network devices and information sharing among them for improving network efficiency and reducing energy consumption compared with the traditional architecture. This paper noted three SDN-IIoT issues while outlining potential solutions. Those issues are system reliability and Information Security (InfoSec), practical implementation, and technology standardization. By considering the hardware implementation, the authors in Li et al. (2016c) propose a framework capable of adapting to dynamic modification in manufacturing orders. The framework ensures the delay-sensitive, permits plug-and-play connection, provides energy efficiency by improving data paths, and shuts down unnecessary manufacturing lines when orders are reduced. Moreover, the framework adopted the flow table cache to address the challenge of memory limitation of SDN switches. This flow table cache is developed based on queuing theory and uses the Jackson queuing method as a fundamental algorithm (Li et al., 2016b). Ref. Caiza et al. (2020) presents a testbed architecture that enables the development of SDN-IoT applications in the industry 4.0 ecosystem. The testbed architecture is composed of triple layers: process layer, SDN layer, and IoT application layer. The IoT server is connected through the MQTT protocol to support the sharing of information among layers. Furthermore, a graphical user interface is developed to provide the monitoring of industry 4.0 procedures.

7.1.3. SDN-IIoT-based DT

In Kherbache et al. (2021), the authors propose a holistic DT structure for IIoT where the network element is regarded in the adoption of Network DT (NDT) technology. In general, NDT improves the network performance by analyzing the data gathered through AI models, avoiding network breakdown and increasing the remaining lifetime of the network. However, utilizing the DT needs to reconsider and reinstall the basic system software, as well as the manufacturing line devices and their cloud/hardware interconnection. These consume more costs and could open the door for DT technology only for big enterprises with enough financial capital and human resources (Barricelli et al., 2019).

7.2. Energy efficiency in SDN-IIoT

7.2.1. Advantages of energy efficiency in IIoT

Through the interconnection of different smart devices and nodes, IIoT enables manufacturing firms to optimize performance and increase productivity with minimum costs (Wu et al., 2022; Boobalan et al., 2022). Irrespective of the challenging economic situation posed by the COVID-19 pandemic in the previous four years, investments in IIoT have shown a favorable trend, with additional growth anticipated over the coming years. In IIoT, energy efficiency is becoming a core aspect. The objective of the "Green Factory" is presently being sought not only to address the demands of sustainability and environmental, but to augment industry profitability (Tabaa et al., 2020). The increasing cost of energy consumption and its effect on the environment have

Table 7
A summary of existing papers focused on resource management.

Ref.	Algorithms and techniques	MEC	NV	IIoT/IoT P	CD	Delay/L	CU	EC	Thr.	Band.	CPA CN	Evaluation	Environment
Huo et al. (2016)	ICN	✓	✓	×	Edge Fog Cloud	✓	×	✓	×	×	C SDNC (NS)	Simulation	SDN-IoT
Sharma et al. (2017)	- Classified advertisement matchmaking - Scheduling (CLOUDRB) - Conventional symbolic	×	×	×	Edge Fog Cloud	✓	×	×	✓	×	D SDNC (NS)	Simulation	SDN-IoT
Becker et al. (2019)	- Threshold - Admission control	×	×	×	×	×	×	×	✓	×	C SDNC (NS)	Simulation	SDIN
Bello et al. (2020)	- Dijkstra's shortest path - Clustering	×	×	TSCH	×	✓	×	×	×	×	C SDNC (NS)	Testbed	SD-IWSN
Wang and Li (2018)	- CMS - Adaptive selection and task priority	×	×	×	Edge Fog Cloud	✓	×	×	×	×	C SDNC (NS)	Simulation	SDN-IIoT
Li et al. (2018)	- Clustering, - Fine-grained - Coarse-grained transmission path	×	×	×	Edge Cloud	✓	×	×	✓	×	C Open Mul	Simulation	SDN-IIoT
Babbar et al. (2021)	- Clustering	×	×	×	Cloud	×	✓	×	×	×	D POX	Simulation	SDN-IIoT
Sahoo et al. (2021)	- Nonlinear optimization - Linear optimization - Traffic flow queue scheduler	×	×	×	Cloud	×	×	×	×	✓	D Floodlight	Simulation	SDN-IIoT
Orozco-Santos et al. (2021)	- TSCH scheduling - Dijkstra's shortest path	×	×	TSCH	×	✓	×	✓	×	×	C SDNC (NS)	Simulation	SD-IWSN
Okwuibe et al. (2021)	- Weakest link approach - SAT solver	×	×	×	Edge Cloud	×	✓	×	×	×	C SDNC (NS)	Simulation	SDN-IIoT
Okwuide et al. (2020)	- Offloading	✓	✓	×	Edge Cloud	✓	×	✓	×	×	C ODL	Testbed	SDN-IIoT

Ref.: Reference; MEC : Mobile Edge Computing; NV: Network Virtualization ; P: Protocol; CD: Cloud Domain; L: Latency; CU: CPU Utilization; EC: Energy Consumption; Thr.: Throughput; Band.: Bandwidth; CPA: Control Plane Architecture; CN: Controller Name; D: Decentralized; C: Centralized; SDNC: SDN Controller; NC: Not Specified; ✓: Considered; ×: Not considered.

made energy efficiency one of the main goals of IIoT. The system of power management has the responsibility of real-time monitoring of the energy sources needed to perform industrial operations, offering valuable information corresponding to the parameters of power distribution and finding patterns (Mao et al., 2021). Energy efficiency in IIoT is essential for the following reasons: firstly, many IIoT devices are resource-constrained in terms of transmission and computation. Secondly, sensing, transmitting, and computing processes carried out by IIoT devices may also result in a rising carbon footprint from the panorama of the entire system (Xu et al., 2022). For instance, as published in GSMA Spec (2014), the number of carbon dioxide emissions generated by mobile networks is expected to reach 345 million tons by 2020, and increase in subsequent years. Furthermore, energy efficiency is more concerned due to its feature of cost reduction that a business can achieve by minimizing energy consumption.

SDN can significantly contribute to decreasing the energy consumption of network elements and data centers (Assefa and Özkasap, 2019; Tuysuz et al., 2017). Green Abstraction Layer (GAL) is one of the traffic-aware SDN paradigms used to reduce energy consumption. GAL allows internal communication between nodes to share energy-related data, like the energy consumption of a specific switch, route, or state. The data are subsequently transmitted to the control plane, which, in turn, analyzes the received data in real-time. Energy Aware States (EASs) mode in GAL is determined through the network logical entity and the energy optimizer module utilizes information from EASs to find out the appropriate state for achieving energy efficiency (Bolla et al., 2013).

7.2.2. Energy efficiency in SDN-IIoT-based edge–cloud computing

SDN-based edge–cloud interplay to manage the big data transmission in IIoT is presented (Kaur et al., 2018), in which SDN is a reliable middleware of different applications. A multi-objective evolutionary framework based on the Tchebycheff decomposition is used to improve the data transmitting and traffic scheduling. This work is evaluated by considering the following optimization goals, that is, the trade-off between delay and power-saving, and the trade-off between bandwidth and power-saving.

7.2.3. Energy efficiency in SD-WSN

In Masood et al. (2019), the Dolphin Echolocation Algorithm (DEA) is adopted in the meta-heuristic framework to determine forwarding paths in SD-WSN scenarios. DEA takes into consideration the residual energy of network devices to choose paths that can reduce energy consumption. Ref. Kipongo et al. (2022) presents an SDN framework based on the fuzzy topology discovery mode to provide energy efficiency in WSN. The authors in Moreno Escobar et al. (2020) introduce a light-diffusion mechanism in WSN to interconnect and share information among the sense devices of IIoT using the intelligence swarm and Peano fractal curve. This mechanism expands the lifespan of industrial networks if a small number of sense devices is activated.

7.2.4. Energy efficiency in SDN controller

Guarantying energy efficiency at the control plane level is a new method for energy-aware in SDN, where the majority of research has concentrated on the data plane layer. Due to the applications utilizing network programmability on the rise, the need for energy-saving on the control plane will follow. Accordingly, the work in Oliveira et al. (2021), provides a method to decrease energy consumption at the control plane level, which is complementary to the multiple current OpenFlow switches solutions. This method utilized the parallel processing abilities of current off-the-shelf multicore processors to divide different functions of the control plane among the cores. By splitting the functions among similar cores, one can reduce the frequency of processes, allaying the complete energy consumption while maintaining the same QoS level. To this end, a summary of existing papers focused on IIoT leveraging SDN has been presented in Table 8.

7.3. Real-time in SDN-IIoT

Ultra-low latency communication is essential to the tasks of a smart factory. The smart factory refers to an intelligent factory that uses information technology in all steps of the manufacturing procedure to boost productivity with lower defect rates. Robots are utilized to automate tasks and different actions are controlled remotely. To avoid malfunction caused by communication delays, real-time networking is required, whereas the rate of data packet transmission is also significant (Yan et al., 2023). IIoT sustains real-time monitoring and control

Table 8
A summary of existing articles focused on IIoT leveraging SDN technology.

Ref.	Algorithms and techniques	MEC	IIoT/IoT P	CD	Delay/L	Inter.	EC	PDR	Thr.	PLR	T/I Level	CPA CN	Environment
Wang et al. (2022a)	- Server-to-Server (S2S) - MNP communication	×	OPC UA	Edge Cloud	✓	✓	×	×	×	×	High	C Ryu	SDN-IIoT
Romero-Gázquez and Bueno-Delgado (2018)	- IoTDM plugin	×	OPC UA, CoAP, MQTT	Cloud	×	✓	×	×	×	×	Medium	C ODL	SDN-IIoT
Wan et al. (2016)	- MapReduce - Cluster resource management system (YARN)	×	×	Cloud	×	×	✓	×	×	×	High	D SDNC (NS)	SDN-IIoT
Li et al. (2016c)	- A real-time message transmission mode - Data backup and double route transmission	×	×	Cloud	✓	×	✓	✓	×	✓	High	D and C SDINC (NS)	SDIN
Caiza et al. (2020)	- Firewall application	×	MQTT	×	×	×	×	×	✓	×	Medium	C Ryu	SDN-IIoT
Kherbache et al. (2021)	- CSMA-CA - TASA	×	TSCH	×	✓	×	×	✓	×	✓	Medium	C SDNC (NS)	SDN-IIoT
Masood et al. (2019)	- DEA	×	×	×	×	×	✓	✓	✓	✓	Medium	C SDNC (NS)	SD-WSN
Kipongo et al. (2022)	- Fuzzy theory	×	×	×	×	×	✓	✓	×	✓	Medium	D SDNC (NS)	SD-WSN
Moreno Escobar et al. (2020)	- Peano fractal curve - Swarm intelligence - Labeled transition system	×	×	Cloud	✓	×	✓	×	×	×	Medium	D ESP8266 microcontroller	SDN-IIoT
Kaur et al. (2018)	- Multi-objective evolutionary-based TC - Workflow classification - Optimization-based selection and execution	✓	×	Edge Cloud	×	×	✓	×	×	×	Medium	C SDNC (NS)	SDN-IIoT

Ref.: Reference; MEC: Mobile Edge Computing; P: Protocol; CD: Cloud Domain; L: Latency; EC: Energy Consumption; PDR: Packet Delivery Ratio; Thr.: Throughput; PLR: Packet Loss Rate; T/I: Testbed/Implementation; CPA: Control Plane Architecture; CN: Controller Name; D: Decentralized; C: Centralized; SDNC: SDN Controller; SDINC: Software-Defined Industrial Network Controller; NC: Not Specified; MNP: Monitor-Notify-Publish; TASA: Traffic-Aware Scheduling Algorithm; TC: Tchebycheff Decomposition; ✓: Considered; ×: Not considered.

of the manufacturing system in order to decrease the maintenance time and allows the performance of devices to be made almost immediately using deployed applications (Salama et al., 2019). In IIoT, there is a significant issue of functional flexibility required to facilitate on-the-fly reinstallation of manufacturing cells and stations. At the network level, this flexibility comprises dynamic data scheduling, handling, and dispatching. SDN is able to provide flexibility, however, its utilization in IIoT has been hampered by limited support for real-time services.

7.3.1. Real-time OpenFlow (RTOF)

METRICS project (METRICS Project, 2023) addresses the limitation above by extending the OpenFlow protocol to provide an SDN interface with real-time reservation. Additionally, Refs. Silva et al. (2017b), Silva et al. (2017a) propose an OpenFlow extension called RTOF that contains the specification of real-time communication, while maintaining compatibility with the current OpenFlow protocol standard. Without any negative impact on traffic segregation, RTOF enables dynamic management of trigger-time and trigger-event with or without real-time flow. The RTOF API is extended by adding the admission control technique capable of executing a schedulability test whenever a request for a change in application flows is made to ensure that the available network resources are enable to fulfill the requirements (Moutinho et al., 2019). This extension allows the execution of a novel installation without endangering the existing installations. Furthermore, the extended RTOF API supports virtual data plane switches with real-time communication. However, every switch will need a particular mediator to interpret the RTOF API to its own communication interface such as NETCONF. An implementation of a special OpenFlow pipeline will be required, too. In order to improve the METRICS project, the work in Ribeiro et al. (2019) proposes a prototype that combines Linux Traffic Control, IEEE 802.11e, and RTOF to develop a new RTOF compatible with WiFi real-time data planes and provide flexible real-time reservations controlled by the SDN controller in the Industry 4.0 environment.

7.3.2. Real-time-based TSN

The authors in Nayak et al. (2016) propose a time-sensitive SDN architecture for time-sensitive scenarios, which ensures real-time for time-triggered flows through a routing schedule paradigm, while also applicable for non-time-sensitive scenarios. Moreover, ILP formulations were used to overcome the combined problem posed by scheduling time-triggered flows and routing.

7.3.3. Real-time in PROFINET

SDN technology is used in the PROFINET industrial automation to improve manufacturing performance. The SDN controller gathers network information and configures data plane switches according to PROFINET requirements. In PROFINET, a reactive method is used for non-real-time traffic while a proactive method is used for real-time traffic. However, the real-time mechanism is not deeply analyzed and evaluated as this work only concentrated on the integration of SDN in the industrial plant to simplify network management and remote automation (Ahmed et al., 2015).

7.3.4. MQTT-based real-time

Regarding communication, MQTT is one of the best messaging protocols utilized in IoT, with increasing popularity in IIoT, too. Its reputation comes from its simplicity, scalability, low footprint, and impressive publisher–subscriber messaging approach, which fit resource-constrained nodes. MQTT is generally employed over TCP/IP networks due to the systematic consistency and minimization of data loss through bi-directional channels. However, its QoS policy is limited to the transmission of message and missing real-time communication (OASIS Standard, 2019). This issue of lack of real-time is addressed by extending the MQTT protocol with a novel scheme that enables the system to state their requirements of real-time and interpret those requirements to network reservations. These network reservations are imposed by SDN, specifically the OpenFlow interface (Shahri et al., 2022). The real-time extension for MQTT Sensor Networks (MQTT-SN) is also proposed (Fontes et al., 2020). This extension enables network devices to allocate real-time attributes to topics and supports network interfaces to deal with the messages at the MAC level based on the related topics. However, this work relies on a specific MQTT-SN platform which is unavailable in the new version of MQTT such as MQTT V5.0. Several other papers analyze the applicability of MQTT in real-time services and integrate it with different technologies, such

Table 9
A summary of existing articles concentrated on real-time.

Ref.	Algorithms and techniques	NT	IIoT/IoT P	SAPI	CD	Delay/L	RT	SR	Jitter	RET	CPA CN	Evaluation	Environment
Moutinho et al. (2019)	- Admission control	Wired	×	RTOF	×	×	✓	×	✓	✓	C Ryu	Testbed	SDIN
Silva et al. (2017b)	- Flexible Time-Triggered	Wired	×	RTOF, OF	×	×	✓	×	✓	×	C OFC (NS)	Testbed	SDIN
Ribeiro et al. (2019)	- Linux Traffic Control - Admission control	Wireless	×	RTOF	×	✓	✓	×	×	×	C Ryu	Prototype	SDIN
Nayak et al. (2016)	- Erdős–Rényi (ER) - Barabási–Albert (BA) - Waxman, - RRG - ILP formulations	Wired	×	OF	×	✓	✓	✓	✓	×	C NETC (NS)	Simulation	SDIN
Ahmed et al. (2015)	- PFI - RFI	Wired	×	OF	✓	×	✓	×	×	×	D SDPC	Implementation	SDIN
Shahri et al. (2022)	- Depth-first search	Wired	MQTT	OF	×	✓	✓	×	×	×	C Ryu	Simulation Testbed	SDN-IIoT

Ref.: Reference; NT: Network Type; P: Protocol; SAPI: Southbound API; CD: Cloud Domain; L: Latency; RT: Real Time; SR: Scheduling Runtime; RET: Reconfiguration Time; CPA: Control Plane Architecture; CN: Controller Name; D: Decentralized; C: Centralized; OFC: OpenFlow Controller; NETC: Network Controller; SDPC: SDPROFINET Controller; NC: Not Specified; RRG: Random Regular Graphs; PFI: Proactive Flow Installation; RFI: Reactive Flow Installation; OF: OpenFlow; ✓: Considered; ×: Not considered.

as LoRaWAN (Rosli et al., 2020) and edge computing to meet network scalability and minimize bandwidth utilization (Park et al., 2018). Finally, Table 9 classifies the literature review concentrated on real-time in SDN-IIoT/SDIN.

7.4. Lesson learned: Recap and insights

This section shows different essential domains in SDN-IIoT, including IIoT leveraging SDN, energy efficiency, and real-time. In IIoT leveraging SDN, we provide the role of OPC UA protocol to improve interoperability among heterogeneous devices, the evaluation of experiments based on the advanced testbed, and the applicability of DT in SDN-IIoT. In the domain of energy efficiency, we introduce the benefits of energy efficiency for IIoT and how integrating SDN with other technologies like edge/cloud computing can decrease energy consumption in IIoT. It is clear that many articles have focused on the energy efficiency of field devices and the data plane layer compared to the control plane layer. Due to the increasing number of applications that use network programming and softwarization, further research efforts are needed to develop energy-efficient mechanisms that take into account the control plane layer. Additionally, we present real-time solutions for SDN-IIoT, where RTOF API, real-time-based TSN, and MQTT-based real-time are discussed. We have noticed that the OpenFlow southbound API has been extended to support real-time applications. However, future researches are needed to extend the other APIs, especially the northbound and east/westbound APIs so that they can also support real-time applications.

Several advanced testbeds have been proposed to illustrate the applicability of SDN in IIoT. However, standardized testbeds for SDN-IIoT experimentation are greatly needed, which can offer real results. In many SDN-IIoT testbeds and architectures, most IIoT devices are connected to data plane switches through gateway nodes. By virtualizing them, great improvement can be accomplished in the performance of various parameters. The potential advantages of virtualizing interfaces can also bring more benefits to SDN-IIoT testbeds and architectures. For instance, a solution for virtualizing gateways is proposed in Ojo et al. (2016). This is a fascinating solution, since the gateway does not always have sufficient capabilities. Thus, virtualizing its functions can be helpful. However, the main emphasis here is on the connectivity provided by edge/fog/cloud/layers, rather than the IoT network itself. In SDN-IIoT testbeds, it will be crucial to have controllers that can seamlessly integrate into access networks and effectively communicate with IIoT mobile devices. There are also other essential topics to further investigate in the future, such as synchronization and compatibility of different IIoT mobile devices in SDN-IIoT testbeds.

8. Integration of different technologies in SDN-IIoT

In this section, we firstly discuss the two famous technologies named TSN and NFV and their integration in SDN-IIoT. Moreover, we present the implementation of TSN in wireless networks (WTSN) as the current project and a case study based on an SDN-IIoT-NFV architecture.

8.1. TSN technology

8.1.1. Introduction to TSN

To enhance the real-time performance of IEEE 802 networking standards, the IEEE 802.1 TSN task group has implemented a set of technical standards called TSN (Anon, 2023). TSN emphasizes on four key facets, that is, transient synchronization among nodes, end-to-end delay constrained, high availability for real-time traffic flows, and network resources management. In TSN, bridges and end stations can be synchronized in due time to guarantee the minutest jitter and synchronization requirements of time-sensitive services. To make faultless redundancy and guarantee the network reliability of real-time services, the source node can forward duplicated packets along diverse disjoint routes in network connectivity (Vlk et al., 2022; Fedullo et al., 2022). For enhancing the scheduling-based TSN, no-wait scheduling TSN models are proposed to provide the deterministic transmission schedule with the fewest possible network resources so that entirely of the residual resources can be utilized to boost the throughput across best-effort services. In addition, a joint model for data packet fragmentation is developed. Because of that, a spec for the joint model issue is addressed in order that off-the-shelf solvers can be utilized to determine the optimum solutions. To enhance the network scalability, the worst-case latency of data packets is explored, and then, a heuristic model is developed to build latency-sensitive schedules (Jin et al., 2021).

8.1.2. TSN-based wireless network

Network connectivity is a key element in the adoption of TSN in IIoT. Nowadays, only cables or wires are used to implement TSN. For the moment, wireless devices are unable to use TSN technology. Research on how to improve TSN abilities to support the wireless network is in progress (Kang et al., 2021). Network organizations, developers, and engineers should be aware of these restrictions when designing the hardware they intend to deploy and the interfaces needed for connectivity. Sellers will also make gateways like SERCOS to communicate through the TSN-based Ethernet network, enabling them to take advantage of Ethernet without necessarily having to replace all of their elements completely, which would be very expensive (Varis and Leyrer, 2018). In this regard, a method to translate the flows between 5G and TSN based on feasibility studies and industrial automation scenarios to assess the performance of this translation is already developed (Satka et al., 2022).

8.1.3. Integrating TSN in SDN-IIoT

To further strengthen communication with dynamic network configuration and time-sensitive data transmission, several authors combine TSN and SDN. This synergy allows a set of standards to meet strict QoS requirements enforced by critical services and ensures low delay in the forwarding of data packets (Said et al., 2019). To exemplify, the authors in Balasubramanian et al. (2021) propose a paradigm based on online techniques that use SDN in the backdrop of TSN to develop a control rule method named TSNu that provides communication time-slot distributions for scheduled flows and reduces congestion. This work optimizes routing, admission control, and scheduling in the TSN network with gate control constraints.

Smart manufacturing improvement is imposing the upcoming IIoT networks to go well with flexibility and adaptability. Furthermore, future converged network services will need an upper level of scalability. In line with that, SDN, OPC UA, and TSN are the three technologies that are now receiving special attention to enhance adaptability, flexibility, and scalability. Like in Kobzan et al. (2020), SDN, OPC UA, and TSN are merged into an industrial network architecture. However, the synergy among these three technologies is complex and standardization is necessary. In Chahed and Kassler (2021), the authors argue that SDN would be used to address the configuration issue in the TSN system. Furthermore, the authors proposed the centralized network configuration founded on a micro-service architecture to optimize network scalability and flexibility. The utility maximization method is designed to improve the resource scheduling and flow routing, as well as to ensure the latency-sensitive of data transmission in TSN scenarios. However, to deal with the incompatibility challenge posed by heterogeneous interfaces and protocols, the packet head needs to be rewritten, so that data packets from the gateway can be transmitted. Also, the authors do not take into account the implementation or the simulation of the proposed framework.

Ref. Pang et al. (2020) provides a traffic flow scheduling scheme to ensure lossless data without additional update overhead when the network is updated. Dual approaches for diverse use cases are developed. The offline approach consumes additional time but offers greater schedulability whereas the online one executes faster with lesser schedulability. SDN and TSN are fundamentally different, both have dissimilar merits and demerits as reviewed in Silva et al. (2019). Moreover, their basic operational concepts and capabilities for improving Industry 4.0 have been revised. Particularly, the domains pertaining to overhead, flexibility, mutual isolation, management of traffic, real-time communication, and QoS control-based granularity are well considered.

8.2. NFV technology

8.2.1. Introduction to NFV

NFV is a networking technology that virtualizes diverse network device functions to make the services from these devices significantly more scalable and adaptive. NFV can considerably minimize hardware costs, enhance performance effectiveness, and greatly shorten the network service procedure. Although NFV and SDN are two different and autonomous technologies, putting them together on the network architecture, they can bring several benefits to different services and domains. SDN offers logically centralized management based on programmability and dynamic configuration, whereas NFV uses virtual machines instead of network hardware. Virtual machines utilize a hypervisor interface to run software and different algorithms such as load balancing, routing, etc. Moreover, NFV enables the virtualization of network hardware and applications running on commodity servers.

Lately, the integration of SDN and NFV has received tremendous interest from both academic and industrial domains (Alam et al., 2020; Ramakrishnan et al., 2020; Kim et al., 2021; Yang et al., 2020; Chin et al., 2023; Bradai et al., 2020). This integration plays an essential function in IIoT due to making different levels of QoS in network devices and the capability to dynamically reconfigure the network according to the change. The utilization of NFV enables the division of a single physical device into multiple virtual logical networks, named slices. In the connectivity system, different network slices have different logical network topologies, requirements, functionalities, and security instructions for the goal of achieving diverse business objectives. By means of NFV, physical devices can be utilized for share-out applications, services, and resources among segregated slices. SDN brings a simple way to determine several network slices and to assign each slice the tasks it must perform, and then SDN makes it easy to configure and manage those slices while facilitating their perfect separation through network hypervisors. As standards are formalized, logical network slicing is anticipated to play a significant positive impact in moving ahead 5G networks (Wijethilaka and Liyanage, 2021; Afolabi et al., 2018).

8.2.2. Integration of NFV in SDN-IIoT

Various types of literature review have combined SDN and NFV in IIoT. For instance, PrioSDN-Resource Manager (Leonardi et al., 2020), is a resource management scheme based on the admission control to decrease the complexity of network management pursuant to the changeable workload and traffic priorities. This scheme enforces bounds on resource consumption for network slices, which in doing so, share several topology links while remaining the separation from each other. This scheme has also leveraged a runtime bandwidth-based priority with a distribution approach to dynamically respond to load modifications (for example, due to alarms). In Struhár et al. (2019), a Dynamic Bandwidth Allocation (DART) mechanism is proposed, which is capable of dynamically allocating available bandwidth among multiple services to provide effective utilization of network resources. In DART, SDN controllers collaborate to allocate the bandwidth between network slices in order to lessen network congestion on shared links. DART contains dual main elements, that is, the centralized element and the distributed element. The centralized element is in charge of organizing the bandwidth distribution over the whole network and the distributed element addresses the synchronization of bandwidth among virtual slices.

The works stored in Girs and Ashiaei (2018), Ashjaei and Girs (2020) discuss the dynamic transmission required to offer network flexibility. They consider a hybrid network composed of wireless and wired networks managed by the SDN controller. An admission control deployed inside the SDN controller is used to dynamically coordinate the bandwidth distribution among wireless devices. SDN controller comprises the overall information regarding the bandwidth utilization and each network device requests the specific bandwidth from the SDN controller. Then, SDN controller can make a decision on how to allocate the accessible bandwidth among the connected devices according to their traffic priority. Ref. Ji et al. (2021) proposes a mechanism named DNSO, which is a Dynamic mode based on Network Slicing Orchestration for remote adaptation and efficient installation in IIoT. DNSO mechanism supports different requirements and intentions of IIoT services such as video surveillance, real-time information monitoring, and remote process. Moreover, two heuristic models based on AI and theoretical study of the network slicing technology are developed to improve the distribution of network resources and adjust according to the dynamic network environment.

A scheme based on virtual fog-Radio Access Network (RAN) using 5G wireless networking is proposed (Rahimi et al., 2021). Within this scheme, NFV-based fog computing improves the performance of cloud computing at the edge network level. SDN controller is responsible for managing RAN functions, while the NFV orchestrator takes care of the baseband unit instantiation and the radio resource manager deals with the distribution of radio resource and beamforming transmission. Specifically, the scheme optimizes the functionalities of dynamic radio, baseband resource distribution, and beamforming transmission for gratifying IIoT users.

Table 10
A summary of existing articles on TSN/NFV deployed in SDN-IIoT/SDIN.

Ref.	Algorithms and techniques	TSN	NV/ NFV	NS	IIoT/ IoT P	CD	Delay/L	PLR	EC	Thr.	Band.	CPA CN	Evaluation	Environment
Balasubramanian et al. (2021)	- Maximal edge-disjoint spanning trees - Edmonds's algorithm - Breadth-first search - Shortest route - Lyapunov function	✓	×	×	×	×	✓	×	×	×	×	C POX	Simulation	SDN-IIoT
Kobzan et al. (2020)	- TAS - RTman	✓	×	×	OPC UA	×	✓	×	×	×	×	D ODL	Implementation	SDIN
Chahed and Kassler (2021)	- CNC - Micro-service-based architecture - Kubernetes cluster	✓	×	×	×	×	×	×	×	×	×	C SDNC (NS)	×	SDIN
Pang et al. (2020)	- ILP-based scheduling - Time-Triggered (TT)	✓	×	×	×	×	✓	✓	×	×	✓	C NETC (NS)	Simulation	SDIN
Leonardi et al. (2020)	- Priority-based admission control	×	NV	✓	×	×	×	×	×	✓	×	D Floodlight	Testbed	SDN-IIoT
Struhár et al. (2019)	- Admission control	×	NV	✓	×	×	×	×	×	×	✓	D Floodlight	Testbed	SDN-IIoT
Girs and Ashiaei (2018)	- Admission control	×	NV	✓	×	Fog Cloud	×	×	×	✓	×	D Floodlight	Testbed	SDN-IIoT
Ji et al. (2021)	- EA; - NAA; - LARAC - Shortest route	×	NFV	✓	OPC UA	×	✓	×	×	×	×	C Ryu	Simulation Implementation	SDN-IIoT
Rahimi et al. (2021)	- MINLP; - SCA - Shortest route - Dual Descent (DD) - FDD; - MKP - Poisson model	×	NFV	×	×	Edge Fog Cloud	×	×	✓	×	×	D Edge Controller, RR Controller, SDNC (NS)	Simulation	SDN-IIoT
Mai et al. (2021b)	- DDPG; - TMDDPG - TRL; - MDP	×	NFV	✓	LoRaWAN	×	✓	✓	✓	×	×	D SDNC (NS)	Simulation	SDN-IIoT

Ref.: Reference; TSN: Time-Sensitive Networking; NV: Network Virtualization; NFV: Network Functions Virtualization; NS: Network Slicing; P: Protocol; CD: Cloud Domain; L: Latency; PLR: Packet Loss Rate; EC: Energy Consumption; Thr.: Throughput; Band.: Bandwidth; CPA: Control Plane Architecture; CN: Controller Name; D: Decentralized; C: Centralized; SDNC: SDN Controller; NETC: Network Controller; NC: Not Specified; RR: Radio Resource; TAS: Time Aware Shaper; CNC: Centralized Network Configuration; EA: Evolutionary Algorithm; LARAC: LAgrange Relaxation-based Aggregated Cost; NAA: Natural Aggregation Algorithm; MINLP: Mixed-Integer Nonlinear Problem; SCA: Successive Convex Approximation; FDD: Frequency Division Duplex; MKP: Multiple Knapsack Problem, DDPG: Deep Deterministic Policy Gradient; TMDDPG: Transfer learning-based Multiagent DDPG; TRL: Transfer Reinforcement Learning; MDP: Markov Decision Process; ✓: Considered; ×: Not considered.

A virtual network slicing paradigm over LoRaWAN-based SDN is proposed (Mai et al., 2021b), where the OpenFlow controller dynamically separates hardware resources into several network slices. Because of the limited number of available channels on LoRa gateways, network slices may experience a lack of resources and malfunctions. To tackle this issue, a Deep Deterministic Policy Gradient (DDPG) approach founded on a slice optimization model is developed in favor of determining the optimum slice parameters used in the configuration process. Furthermore, a Transfer learning-based Multi-agent DDPG (TMDDPG) framework is developed to speed up the learning procedure along different LoRa gateways. Having analyzed all the proposals discussed in this section, a summary of articles related to SDN-IIoT-NFV/TSN is presented in Table 10.

8.2.3. A case study based on an SDN-IIoT-NFV architecture

Network slicing enables different logical networks to be designed based on a single shared network hardware. Every slice is an independent network entity that can be adjusted and customized to meet the demands of a specific service. As every slice is logically isolated, there is no conflict between the traffic of one and that of the other. It is possible to slice and configure the segments as needed without disturbing the network system. This allows operators to define application requirements for different scenarios on a network system, and update or modify them as needed. Another benefit of network slicing is the capability to provide additional security measures to specific slices that deal with critical services. Furthermore, network slicing is resilient to cyber-attacks as breaches can be restrained in a single slice and prevented from spreading to other network sections (Zahoor et al., 2022; Javed et al., 2022). Fig. 15 shows an architecture of SDN-NFV-IIoT. This architecture contains three layers as follows: infrastructure layer, control and application layers.

(a) Infrastructure Layer: this layer contains IIoT devices such as SDN switches and edge servers that focus on data transmission and storage, respectively, as well as intelligent machines used in smart manufacturing. These IIoT devices, edge servers, and intelligent machines are deployed in different network slices. As a case in point, IIoT devices and machines used to provide non-critical services and do not require real-time communication can be performed under one slice, while IIoT devices and machine used to provide critical services and require real-time communication with network access priority operate on another. Each SDN hardware switch is used to design logically isolated virtual switches and each hardware edge is used to design logically isolated edges (Dressler et al., 2022). The hardware switch contains several hardware Network Interface Cards (NICs), and every hardware NIC makes a virtual switch by designing virtual NICs through the hypervisor software. The hypervisor software makes virtual NICs able to run different virtual edges constructed based on a single hardware edge since every virtual edge requires its virtual NIC. So, flow rules run on both hardware switches and virtual switches (Bueno et al., 2022; Paganelli et al., 2021).

(b) Control Layer: this layer contains one hardware global controller and two hardware local controllers. Each hardware local controller constitutes dual logically isolated virtual controllers and each slice is controlled with its associated virtual local controller. FlowVisor is adopted in the architecture because it is the most common hypervisor software utilized to operate and communicate with physical devices in SDN layers. It monitors virtual network segments and assigns the network resources (e.g., link capacity) to every network slice (Blenk et al., 2016).

(c) Application Layer: this layer involves a set of IIoT network applications. Every network application runs in its own network slice. Tenants run several applications concurrently on each virtual local controller in entire virtual networks (Paganelli et al., 2021; Wang et al., 2017). In closing, a Virtual Infrastructure Management (VIM) is connected to a Network Orchestrator to deal with inter-domain routing. The VIM supports the performance of virtual network applications like load balancing, access control, firewall, routing, etc.

8.3. Lesson learned: Recap and insights

This section provides the integration of emerging technologies in SDN-IIoT, especially TSN and NFV. For the side of TSN technology,

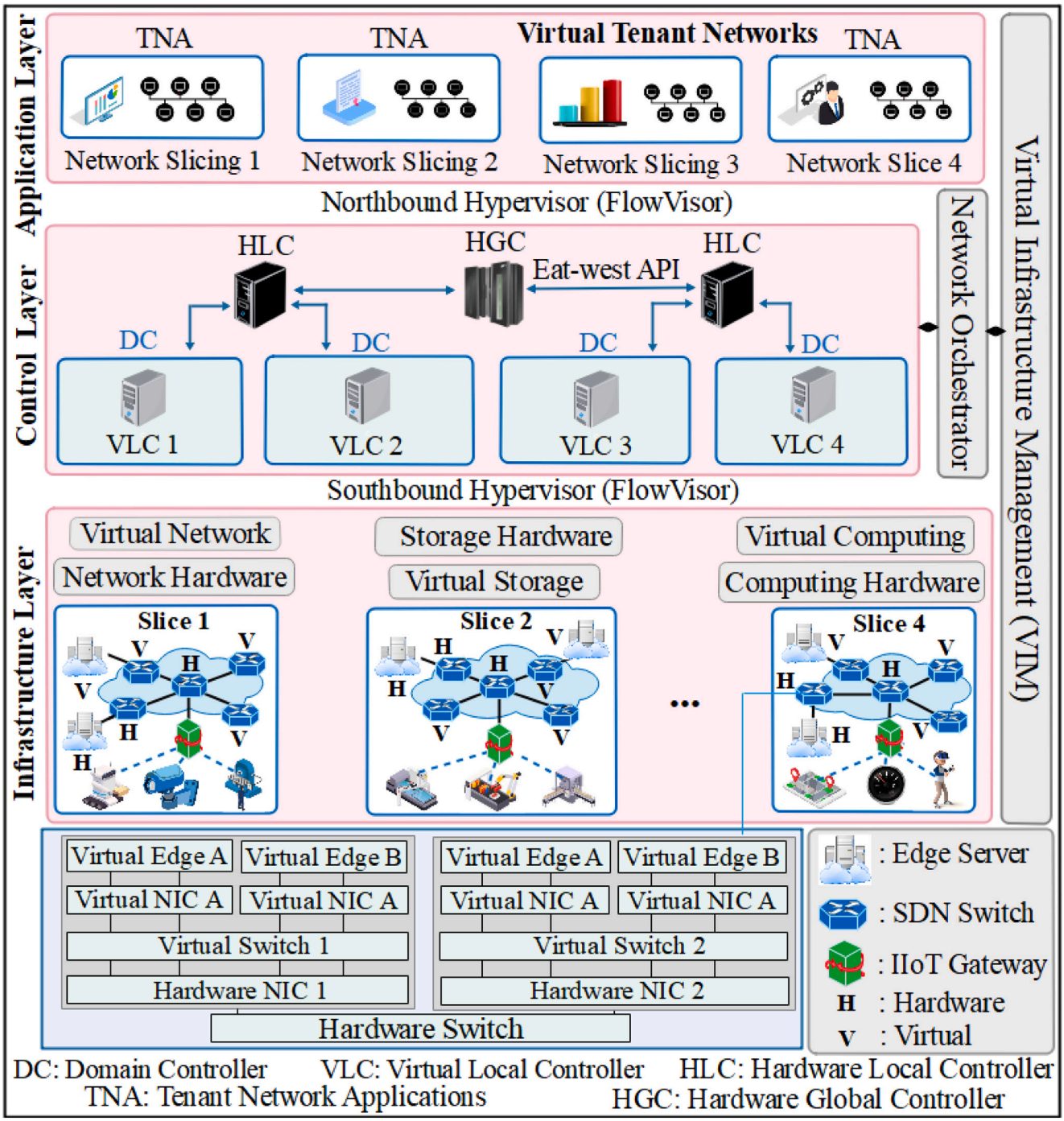

Fig. 15. An architecture of SDN-IIoT-NFV.

we describe the benefits of integrating TSN in IIoT and the project of deploying TSN in wireless networks. Furthermore, articles that focused on the inclusion of TSN in SDN-IIoT were discussed. For the side of NFV technology, we focus on network slicing because of its ability to enhance performance and simplify the development of different applications for diverse organizations. Furthermore, network slicing facilitates the deployment of 5G technology in SDN-IIoT and tackles all the issues of 'DiffServ' which is one of the best-known QoS solutions. However, there is still no industry consensus on how to properly implement network slicing. Table 10 summarizes the details of articles connecting TSN to SDN-IIoT and NFV to SDN-IIoT. As shown in this Table, many performance evaluation results are validated based on the simulation and testbed. Hence, the use of TSN in SDN-IIoT and NFV in SDN-IIoT needs further advanced hardware implementations. This is the key to their standardization efforts and real deployment in the current smart manufacturing. Additionally, Python and MATLAB simulators have been used in the Refs. Balasubramanian et al. (2021), Kobzan et al. (2020), Pang et al. (2020), Ji et al. (2021), Rahimi et al. (2021). However, these simulators are not specifically developed for SDN-TSN which may affect the results. Therefore, further research is needed for developing new simulators for SDN- IIoT-TSN and SDN-IIoT-NFV frameworks to improve the accuracy of results. Admission control has been adopted in several SDN-NFV frameworks. The reason for this is that, admission control guarantees that there are adequate resources available in a cluster in order to prevent the lack of resources.

Moreover, it avoids network congestion, providing resource optimization, and improved network security.

The deployment of SDN and synchronization of diverse virtual functions with orchestrator could potentially enhance IIoT networks. The use of NFV for offloading the complex tasks to a gateway or other IIoT nodes can be extremely helpful in different technologies, including but not limited to image analysis, blockchain, and Augmented Reality (AR)/Virtual Reality (VR). Most of the solutions mentioned in SDN-IIoT-NFV rely on third-party services to support and manage the network topology, which often leads to compatibility challenges. In that regard, open standards can provide more benefits. Researchers may also focus on implementing SDN-NFV based on advanced real-time applications to orchestrate and well manage IIoT nodes, particularly in the realm of 5G/6G mobile networks.

SDN-IIoT-NFV and SDN-IIoT-TSN domains consist of different components, virtual functions, applications, APIs, etc. If one of these domains is targeted by malicious actions from adversaries or attacks, it could lead to negative impacts on the entire system. In case these domains are well protected, then availability, integrity, accountability, access control, and confidentiality can be maintained. From this point of view, research direction on packet inspection methods and access control requires more investigation, particularly for resource-constrained IIoT nodes. Moreover, the implementation of a dedicated firewall SDN controller that includes an intelligent approach for detecting anomalies is necessary to coordinate the deployment of virtual functions in suitable network locations.

9. Applicability of ML and AI in SDN-IIoT

In this section, we review the current literature review on AI/ML applied to improve the performance of SDN-IIoT. AI and ML have learning capabilities to provide traffic optimization and network management. What is more is that AI/ML can be utilized to discover anomalies and vulnerabilities and facilitates proper decision-making in an automated manner. AI/ML models can be used as independent modules or integrated into SDN layers to bring additional intelligence to the SDN architecture, especially at the control plane layer. This layer utilizes AI/ML to carry out data analysis and error/failure detection (Han et al., 2022).

9.1. The synergy of ML/AI with SDN-IIoT

To enhance the QoS of industrial communication, network traffic prediction has become a significant topic for researchers, which is helpful for the security and management of the network. Unfavorably, the traffic flows of the SDN-IIoT ecosystem comprise multiple irregular variations, which lead to the complex for network traffic prediction. To address this challenge, the authors in Wang et al. (2022b), provide a paradigm founded on multi-task learning and Long Short-Term Memory (LSTM) with Convolutional Neural Network (CNN) to predict network traffic based on the temporal and spatial characteristics of network traffic. Hence, this paradigm achieves accuracy and real-time. Moreover, the Iterative Proportional Fitting Process (IPFP) model has been adopted to ulteriorly regulate network traffic forecasters.

In order to accomplish the efficient data transmission that ensures QoS in SDN, the QoS requirements of packet flow must first be identified effectively and accurately. The integration of several ML models and a packet flow classification algorithm named "Misclassification Aware Conjoint Classification Approach Random Forest Random Forest (MACCA-2RF)" is proposed (Sun et al., 2021). MACCA-2RF contains dual mean classifier mechanisms, a misclassification results judgment mechanism based on the C4.5 technique and a decision mechanism founded on RF technique. With the condition of picking a few amounts of data streams, MACCA-2RF can rapidly and accurately classify packet flows to achieve QoS requirements. The thresholds and link parameters are updated according to diverse QoS requirements. Moreover, route selection, local routing change, and smart routing algorithms are designed to dynamically regulate the forwarding route based on the QoS requirements of packet flows and the link status, before and after network congestion.

A traffic classifier scheme in SDN-IoT is proposed (Owusu and Nayak, 2020). Three ML models: Decision Tree Classifier (DTC), K-Nearest Neighbors (KNN), and Random Forest Classifier (RFC) are analyzed based on the ToR dataset. These models utilized statistical properties of data flows to classify the traffic according to the QoS requirements including bandwidth and latency. DTC and RFC achieve good results in terms of training time and F1 score. KNN was not further considered because of poor performance in comparison with the other two. Dual selection techniques: Shapley additive explanations (SHAP) and Sequential Feature Selection (SFS) were used on the two picked ML models to optimize their functionalities and minimize both the training time and the number of properties. The evaluation displayed that SFS with RFC attained the most effective results.

A chapter on advanced deep learning for image processing is provided in Lv et al. (2022). The system of image processing and the applicability of deep learning in SDN-IIoT are introduced with the simulation experiment. However, the role of SDN is not clearly illustrated. In Manogaran et al. (2021), the authors propose a scheme that combines flow management and service virtualization for appropriate utilization of resources in 6G-cloud-IoT-based terahertz communication. The service virtualization procedure is made by a linear decision-making algorithm for distinguishing tardiness and overloading, as well as producing virtual machines. This algorithm is also used to perform the service distribution and request mapping. Besides, user allocation is granted through deep ML to recognize partial states and acts of the service requests. In Chen et al. (2022), an Adaptive Load-Balancing scheme founded on dynamic Link-condition Prediction (ALBLP) is proposed to address the impact of communication latency between data plane switches and SDN controllers on load-balancing. ALBLP predicts the link-condition value using an optimized LSTM model and Dijkstra route cost is considered as the predicted value to determine the best route among hosts based on the proactive flow installation method. To minimize the communication latency between data plane switches and SDN controllers, the decision-making model was adopted in the prediction process.

In Gao et al. (2020b), the authors analyze the prospective implicit connections among APIs and among users by developing a similarity computation between dual APIs and between dual users. The determined similarities are to be implicit knowledge in the IIoT system using the collaborative learning mechanisms. This work also proposes three APIs recommendation approaches which are constructed by utilizing matrix factorization. One approach is developed with the regularization term regarding the user, another approach is developed with the regularization term regarding the API. After that, another approach is an ensemble approach merging mined implicit knowledge of API and user. The authors in Liu et al. (2021) propose an Imitation Learning-based Flow Scheduling (ILFS) scheme. ILFS takes as input to the network state gathered through P4-based In-band Network Telemetry (INT) and tries diverse flow forwarding routes to learning experiences of diverse network conditions. ILFS mixes the Soft-Actor-Critic (SAC) model with the Generative Adversarial Imitation Learning (GAIL) algorithm. This combination allows the control plane to independently learn and determine the optimal scheduling route for the big flow in accordance with the network environment. Attracted readers may refer to Hauser et al. (2023) for more information regarding P4.

9.2. A case study based on an SDN-IIoT-AI network architecture

Fig. 16 depicts an architecture for SDN-IIoT-AI which contains four layers, that is, application layer, control layer, virtual and device layers. Clustering method is utilized at the device layer level to systematize IIoT devices into groups or clusters. In the domain of WSN,

Table 11
A summary of existing articles related to ML/AI.

Ref.	Algorithms and techniques	NFV	5/6G	Topology	CD	Delay/L	PLR	Acc.	Thr.	Band.	CPA CN	Evaluation	Environment
Wang et al. (2022b)	- CNN; - LSTM; - IPFP - Multi-task learning - Network tomography - Network traffic prediction	×	×	Mesh	Edge	×	×	✓	×	✓	C SDNC (NS)	Simulation	SDN-IIoT
Sun et al. (2021)	- MACCA-2RF; - ECMP - QoS guaranteed route selection - Dijkstra-based Fibonacci heap - Local routing change; - QI-RM	×	×	Tree	×	✓	✓	✓	✓	×	C Floodlight	Simulation	SDN-IoT
Owusu and Nayak (2020)	- Game-theory-based SHAP; - DTC - Wrapper; - SFS - Greedy search; - KNN - Pre-trained ML classifier-based RF	×	×	Line	×	×	×	✓	×	×	C SDNC (NS)	Simulation	SDN-IoT
Manogaran et al. (2021)	- Linear decision-making - Deep learning	✓	✓	Star	Cloud	✓	×	×	×	×	C SDNC (NS)	Simulation	SDN-IoT
Gao et al. (2020b)	- LFM; - MF - Implicit knowledge discovery - Collaborative learning	×	×	Mesh	×	×	×	✓	×	×	C NS	Simulation	SDN-IIoT
Liu et al. (2021)	- SPF; - SAC - Imitation learning - GAIL neural network - Traffic scheduling	×	×	Tree	×	✓	✓	×	✓	✓	C SDN (NS)	Simulation	SDIN
Chen et al. (2022)	- ALBLP - Dijkstra's shortest path - LSTM training algorithm	×	×	GÉANT	×	×	×	×	✓	×	C Ryu	Simulation	SDN

Ref.: Reference; NFV: Network Functions Virtualization; CD: Cloud Domain; L: Latency; PLR: Packet Loss Rate; Acc.: Accuracy; Thr.: Throughput; Band.: Bandwidth; CPA: Control Plane Architecture; CN: Controller Name; D: Decentralized; C: Centralized; SDNC: SDN Controller; NS: Not Specified; ECMP: Equal Cost Multi-Path; QI-RM: QoS guaranteed Intelligent Routing Method; LFM: Latent Factor Models; MF: Matrix Factorization; ✓: Considered; ×: Not considered.

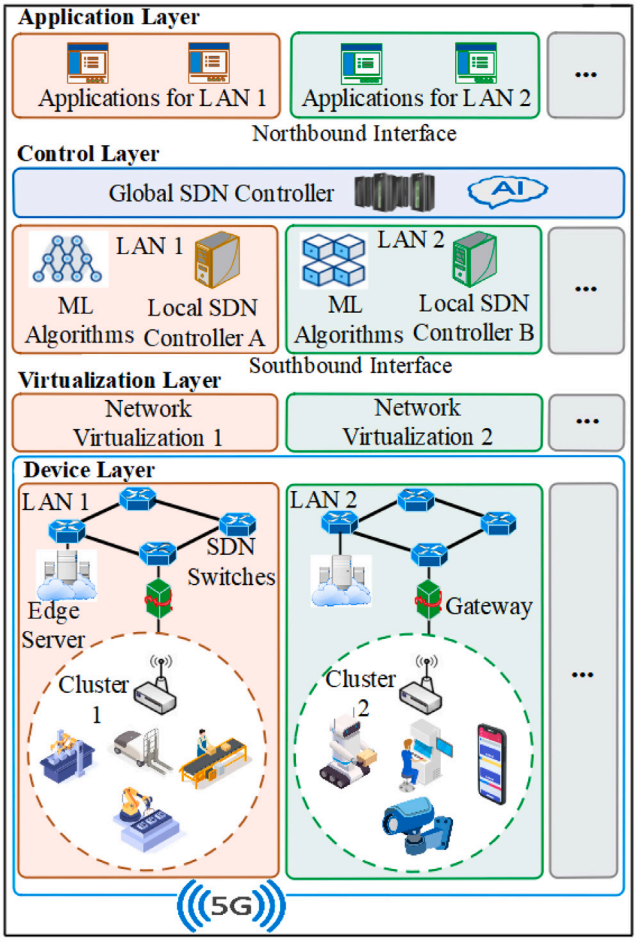

Fig. 16. A Conceptual of SDN-IIoT-AI network architecture.

clustering would make it simple to handle network topology setup and configuration by considering energy efficiency in performance. In this regard, clustering allows adjacent IIoT devices to share common information and resources, thus, reducing scarce resources like energy and bandwidth (Pokhrel et al., 2020). ML algorithms and AI technology are inserted in SDN controllers to provide decision-making abilities to the system. Hence, SDN controllers can be turned into intelligent things, and complexity of IIoT services can be effectively handled and carried out in a better way. In this context, the ML algorithms needed in LAN 1 are installed inside local controller A which manages this LAN 1. Similarly, the ML algorithms needed in the LAN 2 are installed inside local controller B. This is also the case in other LANs deployed in the system. This strategy will significantly reduce local controller storage memory and control channel overhead because each local controller will maintain algorithms that are only needed in its corresponding LAN. In the architecture, every LAN has its corresponding cluster deployed in the device layer which uses 5G networks for communication. Therefore, LANs can utilize the high-speed 5G networks to enable parallel execution of multiple tasks in SDN-IIoT-AI. A virtualization layer is attached (e.g. OpenVirtex) to guarantee fine-grained communication services. By using the virtualization layer based on network slicing, it is possible to decouple the underlay network into several virtual networks to improve the management of heterogeneous networks. As noted in the literature and previous sections, the combination of SDN, NFV, ML, and 5G technologies has paved the way for smart manufacturing and IIoT innovation. Almost all future innovations of IIoT connectivity will rely on wireless networks and high-speed communication. To achieve these innovations, 5G, SDN, ML, and NFV will play an important role.

9.3. Lesson learned: Recap and insights

This section analyzes the use of ML/AI in SDN-IIoT and the advantages of applying ML/AI in SDN-IIoT. Table 11 summarizes current research focusing on the integration of ML/AI in SDN-IIoT. As illustrated in this Table, the most popular ML algorithms are used to improve the performance of SDN-IIoT. Some of them are CNN, LSTM, deep learning, Random Forest, Imitation learning, network traffic prediction, etc. However, it is clear that many papers have considered simulations to evaluate the results of experiments. In future work,

further frameworks considering hardware implementation in the validation of results are needed to improve the use of ML/AI in SDN-IIoT and the standardization of SDN-IIoT-AI for the next generation. To this end, SDN can be utilized to develop multiple-scale traffic predictions with higher accuracy. Moreover, deep learning neurons could be effortlessly modified in heterogeneous IIoT devices by using SDN. This could offer the flexibility to deploy, change, and update deep learning mechanisms in SDN-IIoT networks.

10. Network security in SDN-IIoT

In this section, we showcase several technologies used to enhance SDN-IIoT security, including Manufacturer Usage Description (MUD) and blockchain. Similar to other IoT networks, IIoT becomes more widespread and the challenges of managing its network security also increase. Indeed, network security risks lead to significant financial losses for industrialists and erode users' trust for SDN-IIoT (Ferrag et al., 2022; Jhanjhi et al., 2021; Ali and Khan, 2022).

10.1. Protection against attacks in SDN-IIoT

The centralized control plane strategy used by SDN architecture exposes the SDN controller to the danger of a single point of failure (López-Millán et al., 2023). Firstly, this strategy makes it simple for the SDN controller to be the target area for attackers. When the attacker has completely attacked the SDN controller, it will disturb the whole network managed by the SDN controller. Secondly, due to this centralization strategy, the SDN controller becomes vulnerable to resource depletion attacks like DDoS (Alhijawi et al., 2022; Hu et al., 2021; Mwanza and Kalita, 2023). In case the DDoS attack has been successfully raided and held one or various host servers in the SDN-IIoT system, these managed host servers may transmit a lot of messages to the controller rapidly. Consequently, the controller may collapse or stop replying to regular requests, which may finally result in a breakdown of the overall SDN-IIoT system. What is worse is that hackers may gain access to IIoT devices by invading the SDN controller, therefore, impacting the normal performance of IIoT devices (Du and Wang, 2019; Zainudin et al., 2022; Etxezarreta et al., 2023; Alshahrani et al., 2023).

Multiple frameworks have been proposed to deal with the above challenges. For instance, in Du and Wang (2019), the authors propose a pseudo-honeypot game approach with theoretical enactment to cope with DDoS attacks in SDN-IIoT. This approach can offer dynamic security for SDN-IIoT if confronted with DDoS and malicious attacks. Moreover, the optimum equilibrium between attackers and legitimate users is considered. The authors in Zainudin et al. (2022), present an extreme gradient boosting framework founded on a feature selection technique with a hybrid deep learning classification algorithm. The framework classified network traffic with a small number of trustworthy subsets and recognized different attacks such as DNS attack, UDP flood attack, and SYN flood attack. This framework contains triple segments, that is, data preprocessing, classification of attack types, and feature selection. First, the framework starts with data preprocessing, features are systematized to a specific allotment. Afterwards, the feature selection procedure is pursued. The CICDDoS2019 standard dataset was used in the evaluation of the experiments.

Ref. Yan et al. (2018) presents a multi-level DDoS mitigation scheme to protect against DDoS attacks in SDN-IIoT, which consists of the cloud computing layer, fog computing and edge computing layers. The edge computing layer mostly contains IIoT gateways utilized to interconnect IIoT nodes using a variety of protocols. The fog computing layer comprises applications and different OpenFlow controllers to identify and prevent DDoS attacks. The cloud computing layer utilizes intelligent computing and big data technology to explore traffic flows, as well as discover and stop DDoS attacks. In Li et al. (2020a), the authors propose a paradigm to protect packet forwarding and alleviate Malicious Packet-alteration Attacks (MPA) in an SDN-IIoT-based backend network. The SDN controller explores reports occasionally sent by end hosts to detect malicious switches and proactively chooses reliable routes to recover the communication when the MPA disrupts the system.

A multi-attribute network secure framework for SDN-IIoT is proposed in Chaudhary et al. (2018). This framework applied a cuckoo filter-based speedy transmission method and an attribute-based encryption model to protect packet forwarding. The cuckoo filter as a stochastic data structure is utilized for checking the membership of an element through a hash function. Additionally, the authentication protocol named Kerberos is developed. This protocol facilitates authenticating peer entities by utilizing a third-party authenticator. Hence, users can verify the integrity and authenticity of their saved files on the protected cloud server via their credentials-based Kerberos. Ref. Wang and Liu (2021), conducts research to counteract transmission device attacks in SDN-IIoT. Deep Q-Learning (DQL) and Generative Adversarial Network (GAN) were adopted to improve security. The DQL-based attack tolerance mechanism can direct benign traffic flow in a positive direction around the attacked network devices to enhance the successful arrival rate and offer self-recovery capability. Moreover, GAN was applied to render a practical virtual environment and evaluate the attack tolerance mechanism.

10.2. MUD in SDN-IIoT

MUD technology allows IoT nodes to function in the way that their creators intended. When attacks attempt to attack the IoT node, MUD prevents it from being utilized by attacks because it forces the IoT node to only share information with an authorized destination host. This eliminates the barrier between the creator and the users, and rises a level of protection and trust in the communication. Owing to this, the creators of devices can thereby improve the security of their nodes (Mazhar et al., 2021).

MUD standard has received tremendous interest from the side of IIoT (Krishnan et al., 2021a), and it is regarded by the National Institute of Standards and Technology (NIST) as a favorable technology for the security of IoT nodes against denial-of-service (DoS) attacks (Dodson et al., 2021). SDN paradigms that are in charge of translating MUD instructions into a format that is understandable and executable by OpenFlow switches are presented (García et al., 2019; Hamza et al., 2018; Ranganathan et al., 2019). Specifically, the authors in García et al. (2019) develop a security scheme based on MUD technology to protect IIoT networks. The SDN controller is in charge of converting MUD instructions into flow rules that can be executed by OpenFlow switches. These instructions are also converted into a mediator named Medium-level Security Policy Language (MSPL). A lightweight algorithm extends the MUD benchmark structure in order that only authorized nodes are allowed to access a specific area and protective behavioral profiles are imposed to limit the attack surface. The Protocol for supporting Authentication for Network Access (PANA) as the benchmark protocol to transmit the Extensible Authentication Protocol (EAP) is adopted. Moreover, pre-shared key authentication was applied to support the system on resource-constrained nodes.

In Hamza et al. (2019), the authors develop an ML mechanism for finding out anomalies of MUD-compliant behavior through fine-grained (per-flow) and coarse-grained (per-node) SDN telemetry for every IoT node. The authors evaluate the network behavior of IoT nodes via gathering benign and capacity of attacks. Additionally, this mechanism is able to detect the attack capability on various user IoT nodes with high accuracy and offers insights into the network system. Finally, the authors have developed the dataset and is available for interested researchers and organizations. The authors in Krishnan et al. (2021b), provide an SDN scheme that contains a security monitoring application and multiple Intrusion Detection Systems (IDSs) that collaboratively monitor and secure the MUD-assisted IoT and edge network. This scheme targets to fight against attacks in the edge infrastructure by

controlling the MUD-compliant IoT nodes. In order to resolve conflicts in operator solutions and rules, the work extended the NDT with MUD profiles and created algorithms for discovering behavioral anomalies between the IoT environment and network traffic. The tasks of each node such as SDN switch, IoT gateway, and SDN controller are tracked by using traffic analysis (per-flow and per-node) on several time scales.

10.3. Blockchain in SDN-IIoT

Blockchain technology has been broadly used for IIoT/IoT due to its characteristics of decentralization and immutability. Blockchain has attracted attention from academia and industry, since it provides trustful data storage with verifiability (Latif et al., 2022; Huo et al., 2022b; Jiang et al., 2020; Algarni et al., 2022; Rahman et al., 2020; Meng et al., 2021). Blockchain can handle the security issues being faced by IIoT/IoT nodes because it only allows trusted parties to communicate with one another (Khan et al., 2023; Hakak et al., 2020). The high performance and decentralization merits of blockchain allow it to monitor the information exchange and cache data of decentralized devices, with the hope that there is almost no chance of tampering with this data. Several current works have focused on the use of blockchain in SDN-IIoT/SDN-IoT. For instance, the authors in Gao et al. (2020a), present a secure data-sharing approach for SDN-based Prevalent Edge Computing (PEC) utilizing blockchain and the Identity-Based Proxy Re-Encryption (IBPRE) algorithm to enhance the security of IIoT communication. The encrypted data nodes were outsourced to a third-party cloud server and the IBPRE algorithm was used for safely transmitting the cryptographic file keys among the owners of data and others. Users can cooperate with the blockchain and upload crypto keys from the blockchain, as well as updating and searching files using smart contracts.

10.3.1. Distributed SDN architecture-based blockchain

The applicability of blockchain-based consensus protocol in the IIoT area controlled by a distributed control plane is presented in Qiu et al. (2018), where blockchain performed as a trusted third party to gather and synchronize network broad views among multiple SDN controllers. To enhance the throughput of the blockchain application, the work jointly takes into consideration the trust properties of blockchain devices and SDN controllers, with the computational ability of the blockchain platform. Due to the difficulty of using traditional approaches to address the joint issue, a Dueling DQL (DDQL) solution was proposed to deal with this issue. The paper in Luo et al. (2020), also provides a distributed blockchain scheme using a Deep Recurrent Q-Network (DRQN) algorithm and a Partially Observable Markov Decision Procedure (POMDP) algorithm, as well as considering the optimum trade-off for saving energy and maximizing throughput using a reward function. SDN controllers are incorporated with the blockchain platform to synchronize the local perspective of several SDN controllers and ultimately obtain the global perspective of the entire system via the procedure of reaching a consensus. In Medhane et al. (2020), the authors develop a blockchain paradigm based on distributed security to protect the SDN-IoT ecosystem. In this paradigm, IoT monitoring-based programmability is used to detect the identification of attacks. Moreover, the paradigm accomplishes the data confidentiality of legitimate users in case users are moving or walking. However, the paradigm uses additional energy as it deals with the challenge of node mobility and collaborative attack detection.

10.3.2. Blockchain with IDS

The authors in Derhab et al. (2019), applied an IDS and a KNN-Random Subspace Learning (KNN-RSL) model to detect and defend misrouting of codes and malicious/forged codes. Moreover, a security paradigm-based blockchain was proposed to discover any insertion of deceitful flow rules inside the virtual OpenFlow switches. In the end, this work has illustrated that the KNN-RSL classifier gave preferable outcomes compared to the conventional ML classifiers. The article archived in Madhawa et al. (2018) proposes an IDS that integrates the specification policies and anomaly detection to increase the detection rate with small false positives. With this IDS, it is not necessary for the designer/developer and user to have any prior knowledge of the network structure or its specific components. The data-flow detection mechanism deployed in the SDN switch will automatically build the network structure after verification.

10.3.3. Blockchain with Zero-Knowledge-Proof (ZKP)

To address the issues like DDoS attack and a single point of failure, the work in Singh et al. (2020), proposes a framework-based blockchain that utilizes the ZKP theory for registration of novel SDN switches on the blockchain architecture. Voting founded on the consensus technique is utilized to allocate the affirmative method among SDN switches. The permissioned blockchain architecture only takes into account the domains corresponding to the SDN to improve the block size. Besides, the Deep Boltzmann Machine (DBM) algorithm deployed in SDN controllers was applied to discover anomalies in arriving data flows.

Two different scenarios of malware injection at the switches and controllers have been illustrated (Aujla et al., 2020). For the first scenario, the malware-assault switch can insert its own data and deteriorate the original data. For the second scenario, the malware-assault switch can put malicious commands on several adjacent switches in order to transmit the DDoS attack on the side of controllers. To overcome these attack cases, blockchain-based mitigation mechanisms have been developed. The permissioned blockchain algorithm is adopted to protect the flow tables and traffic flows. Besides, the identity issuer checks SDN switches using the authentication information given by the ZKP theory. Lastly, today's unresolved problems related to the deployment of blockchain in the smart city ecosystem managed by SDN are mentioned.

A zero-trust security framework founded on blockchain-based node authentication is proposed to protect IoT nodes in 5G networks (Li et al., 2022a). Furthermore, we can utilize the blockchain to diminish rumors from spreading like firecrackers. Engineers in this area can support a lot of companies and organizations to prevent bad reputations caused by such misinformation (Yu et al., 2021). To this end, a survey on the deployment of blockchain in the SDN architecture is proposed (Alharbi, 2020), and Table 12 displays a comprehensive summary of network security in SDN-IIoT.

10.4. Lesson learned: Recap and insights

This section presented the network security in SDN-IIoT and outlined the advantages of protecting SDN-IIoT. We provide the current schemes used to fight against attacks in SDN-IIoT. Moreover, different technologies used to improve SDN-IIoT security are discussed, especially MUD and blockchain. The MUD automatizes node category identification which decreases operational costs and enhances security through the standard-based mechanism. However, network nodes should be limited to a predefined anticipated behavior. On the other hand, we discussed the blockchain-based distributed SDN architecture, blockchain with IDS, and blockchain with ZKP. Blockchain performs on a decentralized network, without the need for a central authority, making it very hard for attackers to jeopardize the SDN-IIoT system. Nevertheless, blockchain can slow down when there are many network users and does not scale well due to its consensus technique, and some blockchain-based frameworks consume more energy. Table 12 provides a comprehensive summary of articles related to network security. We notice that some well-known algorithms have been used to improve network security in SDN-IIoT, such as Game theory, deep neural network, CNN, LSTM, Random Forest, Poisson distribution, Markov chain model, KNN, clustering, and so on.

Lack of standardization and the high demand for product delivery have caused significant security risks in SDN-IIoT, which can seriously

Table 12
A summary of existing papers on network security.

Ref.	Considered Attacks	Defense Mechanism	NFV	BC	MUD	CD	Acc.	DT/DR	EC	Thr.	DE/TC	CPA CN	Evaluation Environment
Du and Wang (2019)	- Anti-honeypot - DDoS, - Malicious	Protection Detection	×	×	×	×	×	✓	✓	×	✓	C SDNC (NS)	Testbed SDN-IIoT
Zainudin et al. (2022)	- DDoS (DNS, UDP, and SYN)	Detection Classification Mitigation	×	×	×	Cloud	✓	×	×	×	✓	D ONOS	Simulation SDN-IIoT
Yan et al. (2018)	- DDoS	Detection Mitigation	×	×	×	Cloud Fog Edge	×	×	×	×	✓	D Ryu	Simulation SDN-IIoT
Li et al. (2020a)	- Malicious packet-modification	Detection Mitigation	×	×	×	×	×	✓	×	×	×	C Ryu	Simulation SDN-IIoT
Chaudhary et al. (2018)	- Data masquerading - MITM, - DDoS, etc.	Prevention Authentication	✓	×	×	Cloud	×	×	×	✓	×	D SDNC (NS)	Testbed SDN-IIoT
Gao et al. (2020a)	- DDoS	Mitigation Authentication	×	✓	×	Edge Cloud	×	×	×	✓	✓	C SDNC (NS)	Simulation SDN-IIoT
Qiu et al. (2018)	- Malicious	Mitigation Authentication	×	✓	×	Edge	×	×	×	✓	×	D SDNC (NS)	Simulation SDN-IIoT
Luo et al. (2020)	- Malicious	Mitigation	×	✓	×	Edge Cloud	×	×	✓	✓	×	D SDNC (NS)	Simulation SDN-IIoT
Medhane et al. (2020)	- Various attacks (NS)	Detection Mitigation	×	✓	×	Edge Cloud	✓	×	×	✓	✓	D SDNC (NS)	Simulation SDN-IoT
Derhab et al. (2019)	- Forged commands - Misrouting of commands	Detection	×	✓	×	Cloud	✓	✓	×	×	×	C ONOS	Simulation SDN-IIoT
Wang and Liu (2021)	- Flow table emptying - Flow rule flooding - Packet-in flooding - Malformed message injection, etc.	Mitigation	×	×	×	×	×	×	×	×	×	D ONOS, POX ODL, Ryu	Simulation SDN-IIoT
Singh et al. (2020)	- DDoS (or malicious activity)	Detection prevention	×	✓	×	×	✓	×	×	✓	✓	C POX	Simulation SDN-IIoT
Krishnan et al. (2021a)	- Behavioral anomalies - Malicious attacks - Spoofing, - MITM	Detection Prevention	✓	✓	✓	Edge Cloud	×	×	×	✓	✓	C ODL	Testbed SDN-IIoT
García et al. (2019)	- DoS	Detection Mitigation	×	×	✓	×	×	×	×	×	✓	C ONOS	Testbed SDN-IIoT
Hamza et al. (2019)	- Volumetric attacks (Ping of Death, Fraggle, ARP spoof, etc.)	Detection Mitigation	×	×	✓	Cloud	✓	✓	×	×	×	D Ryu, Faucet, Gauge	Testbed Prototype SDN-IoT
Krishnan et al. (2021b)	- Mirai botnet - Malware, - DDoS, etc.	Detection Mitigation	×	×	✓	Edge Cloud	✓	✓	×	✓	✓	C SDNC (NS)	Simulation SDN-IoT

Ref.: Reference; NFV: Network Functions Virtualization; BC: Blockchain; MUD: Manufacturer Usage Description; CD: Cloud Domain; Acc.: Accuracy; DT: Detection Time; DR: Detection Rate; EC: Energy Consumption; Thr.: Throughput; DE: Delay; TC: Time Cost; CPA: Control Plane Architecture; CN: Controller Name; D: Decentralized; C: Centralized; SDNC: SDN Controller; NC: Not Specified; MITM: Man in the Middle; ✓: Considered; ×: Not considered.

destroy the network infrastructure. The absence of real-time simulation software for SDN-IIoT experiments makes it difficult for the research community to develop efficient security solutions. Furthermore, it is crucial to develop a comprehensive security framework that combines various solutions so as to reduce the security risks in SDN-IIoT. For instance, multiple research studies have proposed security solutions to mitigate field devices attacks, data plane attacks, APIs attacks, and SDN controller attacks separately. Nonetheless, a comprehensive security framework that integrates the features of all solutions would bring significant advantages to the SDN-IIoT environment.

Protecting heterogeneous networks, IIoT nodes, and different service demands is a big issue in the SDN-IIoT environment. Additionally, the SDN-IIoT environment incorporates different platforms, protocols, technologies, servers, and network topologies. All these pose complexities to manage, manipulate, and secure applications that run on various platforms and locations. Moreover, the majority of IIoT devices are constantly connected to the Internet, making them vulnerable to being exploited by attacks. It is also difficult to update security platforms across all IIoT devices because of their different operating systems. The SDN-IIoT application layer could be leveraged to develop new security schemes based on pattern analysis to protect against attacks targeting IIoT devices (Abou El Houda et al., 2021).

11. SDN controllers

SDN controllers (e.g., ODL, Floodlight, Ryu, ONOS, POX, etc.) deployed in the control plane layer manage data plane devices through southbound APIs and receive suitable guidelines from the application layer via northbound APIs. Controllers configure flow entries in data plane devices to execute necessary tasks, like forwarding packets, dropping packets, and so on. There are different types of SDN controllers (distributed, centralized, and hybrid) which are designed to accomplish multiple operations based on several programming languages (Java, C, C++, Python, etc.) (Mamushiane et al., 2018). Ryu is simple and uncomplicated to program. Beginners can effortlessly install and utilize this controller for their studies and projects. ODL is complicated and challenging to develop new concepts; however, it performs better in comparison with the others. For specialists, it is a fantastic choice to use ODL as it is compatible with more protocols and southbound interfaces (PCEP, OVSDB, OpenFlow, NETCONF, etc.) for managing and configuring data plane switches. Modularity, scalability, consistency, Graphical User Interface (GUI), reliability, and security are key points when creating a proficient and reliable SDN controller (Ahmad and Mir, 2021; Hussain et al., 2022). The existing SDN controllers missing standards for anomaly detection, and high-level data analysis techniques. It has been observed that creating a novel brand SDN controller may not be the right decision. But, the current SDN control features need to be improved and strengthened to overcome the aforementioned problems.

11.1. SDN controllers fit for IIoT

In this subsection, the five SDN controllers (Ryu, POX, Floodlight, ONOS, and ODL) more repeated in this survey are explained, which could be appropriate for IIoT services. These five SDN controllers were chosen due to their excellent performance and they are the most widely used in IIoT as portrayed in the previous sections.

There are different techniques used to select the SDN controller. Consequently, picking the optimal controller is a Multi-Criteria Decision Making problem (MCDM). To overcome this problem, diverse

Table 13
A comparison of the most used SDN controllers in IIoT.

SDN controllers	Type	Language	Modularity	Interfaces	Reliability	Scalability	Consistency	Security	Strength(s)
ODL	D	Java	High	OpenFlow 1.1 – 1.5, NETCONF, REST API, PCEP, OVSDB, OSGi, etc.	High	High	High	High	- High performance.
Ryu	C	Python	High	OpenFlow 1.0 – 1.5, REST API, NETCONF	High	Medium	Low	low	- Effortless to program. - Compatible with OpenStack.
Floodlight	C	Java	Fair	OpenFlow 1.0 – 1.5, REST API, Quantum, etc.	Low	Low	High	Medium	- Applicable in different OS.
ONOS	D	Java	High	OpenFlow 1.0/1.3, Neutron, REST API, NETCONF	High	High	High	High	- Compatible with Apache 2.0 licenses
POX	C	Python	Low	OpenFlow 1.0, Ad hoc API, OVSDB	Low	Low	High	Low	- Less memory space. - Applicable in different OS.

D: Distributed; C: Centralized; OS: Operating System.

MCDA strategies are utilized such as ELECTRE III, Goal Programming (GP), Analytic Hierarchy Process (AHP), Evamix, Regime, and so on Khondoker et al. (2014). We recommend researchers to utilize AHP in order to select the optimal SDN controller. This is due to the fact AHP is developed for supporting the procedure of decision-making and is considerably utilized in multiple domains, like engineering, education, administration, and so forth (Ahmad and Mir, 2021). The selection technique which utilizes AHP has several benefits such as the following: pairwise prioritization of conditions as an input, stability verifying, and advantages of prioritizing-based relative instead of prioritizing-based linear. It is crucial to take into account the specificities and requirements of IIoT services/applications before using the AHP method. To this end, a comparison of the five picked SDN controllers that are most used in IIoT is provided in Table 13.

11.2. Lesson learned: Recap and insights

This section provides explanations related to the most used SDN controller in the field of IIoT as displayed in the previous sections. We consider the five most used SDN controllers in experiments, two of them are the distributed SDN control plane (ODL and ONOS), whereas three others are the centralized SDN control plane (Floodlight, Ryu, and POX). Briefly, each SDN controller mentioned above is as follows: (a) ODL is an open-source controller that has almost all the capabilities of a Network Operating System (NOS) and guarantees high-level performance. It has been designed to meet the requirements of different organizations and fields including institutions, data centers, service provider networks, industrial networks, etc. (b) ONOS is an open-source controller for constructing SDN and NFV solutions. ONOS is specifically developed to manage and control large networks with many network nodes and a vast number of flows. It also supports different southbound APIs, making it compatible with multiple network nodes and technologies. (c) Floodlight is an open-source controller that can support a hybrid connectivity system where OpenFlow switches are connected to non-OpenFlow switches in the same network topology. However, its resilience and security are weak. To address this security challenge, Big Switch Networks enterprise has developed a new version named SE-Floodlight. (d) Ryu is an open-source controller developed to enhance network agility and simplify network management, as well as to overcome traffic issues. (e) POX is an open-source controller which uses less memory to run but has low throughput compared to other controllers. Ryu and POX are easy to program for engineers and developers who want to design new applications. However, they are unable to support the distributed SDN control plane architecture. Table 13 synthesizes these five SDN controllers according to their type, language, modularity, interfaces, reliability, scalability, consistency, security, and strength. As depicted in this table, all the analyzed SDN controllers capable of supporting the distributed SDN control plane use the Java programming language. However, there are other SDN controllers compatible with the distributed architecture which are not developed in the Java programming language. Some of them are Kandoo (developed in the Python, C, and C++ programming language), Loom (developed in the Erlang programming language), ZeroSDN (developed in the C++ programming language), and so on.

12. Open issues and future research orientations in SDN-IIoT

After a completely done review of the aforementioned current proposals regarding SDN-IIoT, the next research challenges have been specified. This comprises the issues that have been inadequately addressed and the others that are still available to open new research areas.

12.1. Synergy of SDN with IIoT

Previous studies have been carried out to enhance IIoT through management-based SDN. The deployment of SDN in IIoT has been considered to solve many problems but not all. Therefore, the next concerns would be taken into account before integrating SDN in IIoT. Firstly, SDN has its own feebleness due to the centralized management and limited flow table size of the SDN switch. Secondly, to successfully manage IIoT networks, SDN must be improved. In this context, it should be noted that IIoT imposes particular requirements on the placement of distributed SDN control planes stemming from the processing of a huge quantity of data and a significant number of heterogeneous IIoT devices. Thirdly, the effective identification mechanisms of IIoT nodes and their communication interfaces are among the crucial points in the application of SDN in IIoT. Nevertheless, SDN does not support all IIoT nodes, only it works with special hardware. Lastly, the centralization architecture utilized by SDN can address management challenges and improve network scalability. However, this centralization generates new difficulties in terms of delay, reliability, throughput, single point of failure, and so on.

12.2. Resilience and fault-tolerance applications in SDN-IIoT

Multiple works investigate resilience and fault tolerance applications as illustrated in Section 5. Nevertheless, platforms like Docker (Docker, 2023) and Kubernetes (or K8S) (Kubernetes, 2023) should be more explored in future work to avoid a single point of failure. These platforms would provide an extremely scalable solution to the controller. They have the capacity to spawn a virtualized image of a controller in a very short time and prevent the breakdown of the entire connectivity. The Kubernetes platform provides orchestration of virtualized pods when a pod collapses, as the system will respawn a mirror image automatically. Another platform like OpenStack also renders the capacity to build an overlay network which could comprise

the original controller and some duplicates of the original controller. This would facilitate the enhancement of virtualization on the SDN controller side and also for the overlay network.

Many works have focused on resilience when a link failure or several links/switches failure in SDN-IIoT. However, it is an open challenge on how to recover fast from the failure of an IIoT network device (e.g., sensor, actuator, gateway, sink node, etc.) without affecting the QoS requirements of IIoT, especially delay-sensitive and loss-sensitive. Additionally, In the SDN architecture, Bidirectional Forwarding Detection (BFD) is the standard protocol utilized to speedily recognize a link breakdown in the connectivity. The BFD protocol runs a control technique and echo messages to verify the performance of active links in a very short period (e.g., 5 ms) (Vestin et al., 2015). On the other hand, developing a standard protocol that can detect the failure of IIoT devices and send the notification to the SDN controller to fix the problem promptly is an open problem.

Flow tables at the data plane switch have limited TCAM memory, which may overflow when multiple backup paths are proactively installed. To address this issue, future work should aim to increase the size of flow tables. Furthermore, multiple IIoT services impose SDN controllers to keep more parameters, including updates, cache state awareness, etc. This may overload the SDN controller and result in network disruptions. Thus, deploying a backup controller can be a fast solution when an active controller failure. However, considering the variety of controllers in IIoT, having the backup controller of every controller can be costly. Therefore, it is necessary to investigate low-cost redundancy solutions. There is also a need for scalable and non-complex architectures to overcome this challenge.

12.3. IIoT workstation safety

The IIoT environment is naturally exposed to inevitable external problems such as fires and other natural disasters. These problems constitute a danger to IIoT infrastructures as they can harm connectivity or result in total stoppage of network services and activities. Therefore, disasters and emergencies in the IIoT environment should be given top priority. In case a catastrophe occurs, generating a warning and declaration of alarms to emergency response service providers like fire brigade, police officers, disaster management department, ambulance, and other law enforcement agencies are needed to protect manufacturing industry employees, apparatuses, nodes, documents, etc. Nowadays, most of these emergency services are available and perform their tasks accurately. Nevertheless, the lack or misfunction of communication materials in the catastrophe zone becomes a significant challenge. In case there is a catastrophic incident, cooperation and communication among different IIoT devices and IIoT applications are complex problems. A MultiRoute Resilience method based on SDN technology was proposed to render reliability and greater rerouting in smart city networks in the presence of natural disasters (Aljohani and Alenazi, 2021). However, further research efforts are needed to develop disaster-resilient frameworks and independent applications that would guarantee fast recovery and robust communication in the presence of calamity.

12.4. SDN-IIoT APIs

SDN provides communication services that are not dependent on device vendors because of standardization efforts offered by ONF. The SDN can manage heterogeneous networks where multiple sensors and actuators from diverse vendors can operate on the same network. Therefore, Openflow has the ability to enable compatibility between different data plane devices. On the other hand, when there are multi-vendor solutions at data, control, and application layers, standardization of northbound API becomes absolutely crucial. This standardization will considerably improve network management through the use of customized applications. Until now, no attempt has been made to standardize the northbound API, and many solutions only utilize REST API.

The communication between the controller plane and the data plane occurs mostly through OpenFlow. In spite of that, with the presence of different controllers and IIoT nodes, the appropriateness of OpenFlow may need reevaluation. Moreover, programmable data plane switches and other nodes (for instance, sinks, sensors, actuators, and gateways) may require the optimization of southbound APIs (Vestin et al., 2018). The southbound API, which has the capability to work effectively with all types of SDN controllers and many IIoT nodes, will be another fascinating research direction. In an IIoT network, access points and mobile devices may also require configuration and policy enforcement. This will require improvements to OpenFlow API or the development of novel southbound APIs. Furthermore, with the advancement of network slicing and NFV and their deployment in SDN-IIoT, FlowVisor must be extended to support different architectures and functions.

12.5. Effective big data analytic platforms for SDN-IIoT

For attaining the goal of IIoT and realizing the maximum advantages from the massive amount of data produced by IIoT nodes, there is a considerable demand for effective and reliable big data analytics platforms. Different database management applications are incapable to provide the intended results because these applications are incapable to effectively analyze and handle a large volume of data. Moreover, handling IIoT data by considering delay-sensitive, loss-sensitive, and high security is extremely important since these data are utilized for critical industrial services. Hence, to fulfill the different requirements of IIoT services, there is a need for effective and robust big data analytics platforms to manipulate the data produced by IIoT nodes. In this regard, more research on the use of SDN with edge computing can be considered to overcome the complexity of managing big data and IIoT services. There is also a need for lightweight algorithms to process and classify the data (e.g., Abdellatif et al. (2019)). These algorithms can provide seamless connection, interoperability, data filtering, computation offloading, and security of heterogeneous networks. Therefore, the research community may focus on possibilities of using these algorithms to improve the data analysis in SDN-IIoT.

12.6. Network security in SDN-IIoT

Trustworthiness and confidentiality of IIoT communication are very important in networks. Heterogeneous sub-systems with wireless networks increase the danger. As the connection among nodes will further increase in the upcoming years, the connectivity will become more appealing to hackers. To guarantee network security, the SDN-IIoT system requires to limit network applications to subsets of nodes and detecting unauthorized users in real-time. Moreover, distributed SDN controllers can be used to prevent a single point of failure. In this case, east/westbound APIs will be exposed to attackers. Therefore, more security frameworks should be designed to protect east/westbound APIs against flooding attacks.

The majority of network devices that will use 5G/6G in the future are not only laptops and smartphones, but IIoT/IoT devices (e.g., sensors, actuators, smart robotics, etc.) will also use it. Consequently, securing all these network devices would be heavy work in the next few years. IIoT/IoT devices will be mainly vulnerable to botnet attacks such as Mirai. To mitigate these attacks, new particularized methods must be created. Additionally, the Federated Learning algorithm (Ma et al., 2022; Nguyen et al., 2021; Alotaibi and Barnawi, 2023) should be further considered to improve privacy and security in SDN-IIoT. This algorithm is suitable for privacy-protecting with lightweight models for complex computations.

One of the role models used in the security of SDN-IIoT is the utilization of a virtual firewall. The virtual firewall can be installed in each level of the SDN-IIoT architecture. For instance, at the node level,

gateway or access point level, switch level, and finally at the control plane level. With this strategy, the chances of detecting the danger would be considerably improved, which would also decentralize the detection process.

Besides, blockchain technology can successfully address the security issues faced by IIoT nodes because blockchain permits only trusted participants to communicate with each other. SDN enables continuous network monitoring, whereas blockchain offers decentralized security to prevent a single point of failure. Therefore, the combination of blockchain and SDN can be leveraged to develop a secure layer between the IIoT infrastructure and the edge layer to guarantee privacy and security. Research community may also focus on blockchain to further protect high-speed commutation in SDN-IIoT-based 5G network. In this survey, multiple security challenges and corresponding solutions which predominantly depend on the controller to impose policies have been taken into account. These solutions are mainly developed as external modules to collaborate with the controllers. Moving forward, researchers may emphasize on the opportunities to integrate these modules within the SDN controllers, leading to improved scalability. Moreover, the evaluation of real-time against diverse attack vectors is needed to enhance the accuracy of solutions. It is significant to mention that some solutions (e.g., Li et al. (2019)) utilize regression methods and ML, which are effective in mitigating different types of attacks.

12.7. Deployment of TSN/WTSN in SDN-IIoT

For future SDN-IIoT, it is paramount to appreciate the benefits and relevance of TSN technology and its abilities which enable time-sensitive. Specifically, TSN guarantees different QoS requirements of SDN-IIoT and fault tolerance with no additional applications or devices. TSN is compatible with different devices from different producers and it has the capacity to support current and legacy switching platforms. Due to these, engineers for industrial automation (e.g., PROFINET), middleware platforms (e.g., OPC UA), and automotive networking applications (e.g., BroadR-Reach) have already analyzed the potential of TSN. Nonetheless, the advantages of TSN appear with some drawbacks, such as update cycles for network switches before specified time and great efforts during configuration, which may be solved by the auto-configuration platform (Dürkop et al., 2015) or with the assistance of SDN. Therefore, the inclusion of SDN with TSN would provide an efficient connection between smart factory devices and enable them to meet the essential requirements of IIoT communication. This inclusion should be further investigated in future research.

Intel Labs has made substantial progress in implementing a new WTSN platform compatible with Wi-Fi (Cavalcanti et al., 2019). Primarily, synchronization and time distribution need to be extended to support Wi-Fi. Data from sensors and actuators must be precisely synchronized in due time in order to provide effective and secure control decisions. The synchronization ability could allow nodes and applications along the wireless (e.g., 5G and Wi-Fi 6) and wired fields to synchronize with a grandmaster clock. As a case in point, a WTSN framework for evaluating the time-sensitive through a synergistic robotics system has been developed (Sudhakaran et al., 2022). In this framework, the WTSN schedule was applied to fulfill the time budget-based application and to verify it using a synergistic robotic workcell in the experiment. The results obtained show that the time-sensitive data is less than 5 ms. So far, none of the research has considered the deployment of WTSN in SDN-IIoT. Consequently, further research is needed on the combination of WTSN and SDN to leverage the strengths of both technologies, especially for the coming IIoT networks.

12.8. Collaboration between wired and wireless protocols in SDN-IIoT

With the evolution of communication technologies, many SDN-IIoT systems will contain multiple types of connections, including wired connections and wireless connections. Therefore, an efficient interface for communication between wired and wireless connections becomes absolutely important in SDN-IIoT. In Wang et al. (2019b), the authors overcome the conversion between the wired TSN protocol and the wireless WIA-PA protocol in an industrial environment. However, more investigation is needed on how the wired TSN protocol and other industrial wired protocols can effectively communicate with different industrial wireless protocols such as 6TiSCH, WirelessHART, 6LoWPAN, LoRaWAN, and so on.

12.9. Simulation software for SDN-IIoT

SDN simulators/emulators should be extended to evaluate different SDN-IIoT architectures, protocols, and algorithms. Basically, the Mininet emulator (Lantz et al., 2010) does not support node mobility and wireless modeling. The ns-3 emulator has limited support for the SDN controller and does not totally simulate the handover procedure. Moreover, the EstiNet emulator (Wang, 2014) does not allow modification or expansion of the source-code level. To address the aforementioned challenges, OpenNet was created (Chan et al., 2014), which is an open-source emulator built on ns-3 and Mininet. OpenNet connects ns-3 to Mininet to benefit from the mobility and wireless modeling capacity of ns-3, as well as compatibility with the SDN controller provided by Mininet. OpenNet also extends ns-3 by attaching the channel scanning method. Furthermore, the Mininet-WiFi emulator (Fontes and Rothenberg, 2016) extended the functionality of Mininet by adding virtual Wi-Fi stations, mobility management, and access points without affecting the normal performance of the original SDN platform. To continue, MiniNext (Schlinker, 2014) was implemented as an extension layer for Mininet to integrate network management protocols (e.g., DHCP) and traditional routing engines (e.g., BIRD, Quagga, etc.). The primary problem with MiniNext is that it is only compatible with Mininet v.2.1.0. Distributed Mininet (Lantz and O'Connor, 2015) and Mininet Cluster Edition (CE) (Antonenko and Smelyanskiy, 2013) platforms allow running many Mininet instances on a single network system, and running Mininet over a distributed network system. However, both platforms only support OpenFlow switches.

To simulate SDN-IoT networks, a Mininet-IoT emulator has been implemented to provide the configuration of the 6LoWPAN protocol in an SDN environment (Setiawan et al., 2021). However, on the Mininet-IoT, sensors only perform as nodes. In addition, the integration of other different industrial wireless protocols (e.g., 6TiSCH, WirelessHART, and WIA-PA) in Mininet-IoT remains an open challenge for future research. In Sylla et al. (2022), a new simulator for 5G communication, called Emu5GNet is developed for simulation of essential components of the 5G network architecture, including 5G New Radio (NR), Wi-Fi connection, 5G core network, edge computing, NFV, and SDN. Apart from the emulators mentioned above with their incomplete points, there is a need for more emulators platforms that can simulate and evaluate diverse protocols, frameworks, and architectures based on heterogeneous networks in SDN-IIoT. Additionally, special simulators able to support the combination of SDN with the well-known core elements of IIoT networks like TSN/WTSN and 6G should be developed in the future.

12.10. SDN-IIoT datasets

In 2020, Bennett University published DDOS attack SDN datasets created using a Mininet simulator. The project started by designing ten network topologies in the Mininet simulator with SDN switches controlled by a centralized Ryu controller (Ahuja et al., 2020). However, top-quality training datasets are needed to improve the accuracy of results in SDN-IIoT experiments. Up to now, standard open-source datasets of SDN-IIoT compatible with different simulators and controllers are not yet available. Standard datasets will motivate researchers to further explore the SDN-IIoT network architecture and different experiments through diverse mechanisms, including ML models. As a consequence, the lack of standard datasets in the SDN-IIoT field may limit the implementation of different frameworks.

12.11. Interoperability in SDN-IIoT

The SDN-IIoT system is composed of multiple protocols, heterogeneous wired/wireless networks, multivendor devices, and different cellular networks such as 5G, 4G, etc. Interoperability among these heterogeneous networks and devices is a serious issue. Moreover, data privacy, stream sharing, synchronization, and resource sharing make interoperability more confusing. Therefore, it is essential to develop a calibration architecture for adaptable nodes/devices that will improve interoperability, make synchronization simple, and support standardization. As a case in point, to achieve interoperability and collaboration between the industrial Internet and the IWSN standard (e.g., WIA-PA), a framework that adopts the OPC UA protocol as a middleware to incorporate WIA-PA in the industrial Internet was proposed (Pu et al., 2022). The CoAP protocol is more appropriate in IIoT than HTTP. Consequently, a lightweight approach used to accomplish the transmission of OPC UA messages through the CoAP-based protocol has been developed (Wang et al., 2020). This approach improves the interoperability of resource constrained nodes. Hence, the combination of OPC UA and CoAP can facilitate interoperability in SDN-IIoT. By considering other different protocols suitable for SDN-IIoT (e.g., 6TiSCH, 6LoWPAN, WirelessHART, etc.), further research is needed for designing frameworks that enhance the interoperability of SDN-IIoT. In addition, one potential solution is to consider Cross Technology Communication (CTC), which provides the efficient sharing of information among multiple platforms (Li and He, 2017). Nevertheless, the existing CTC techniques mainly focus on packet forwarding between heterogeneous nodes deployed in the physical layer, yet they fail to offer an entire solution to the SDN-IIoT system. In the future, this technology needs further exploration and extension to efficiently transmit a large amount of data and achieve energy efficiency in the entire SDN-IIoT system. To this end, it is crucial to tackle the issue of interoperability between the newly proposed SDN-IIoT platforms and the existing ones. Journal publishers and famous research organizations should determine the main problems of SDN-IIoT as special issues to minimize the incomplete solutions implemented in the literature review. These journals and organizations should also contribute towards the standardization of SDN-IIoT.

12.12. Applicability of 5G/6G in SDN-IIoT

Emerging 5G/6G networks with their supporting technologies like NFV and Ultra-Dense Networks (UDN) are able to provide new interconnection and generate novel opportunities to enhance the functionality of SDN-IIoT in terms of deterministic communication and energy efficiency (Beshley et al., 2022; Saleh and Fathy, 2023; Pivoto et al., 2023). However, when deploying 5G/6G networks in SDN-IIoT, the number of services provided by IIoT nodes can be abundant, which implies that the resource consumed can also increase significantly. Consequently, the management of these services and IIoT nodes, and synergies between them become a major challenge in SDN-IIoT-based 5G/6G networks. As a result, further development of resource management-based mechanisms and connectivity frameworks to adjust interactions among IIoT nodes and services are needed in the future. Network slicing and synchronization protocols should be given more consideration to develop solutions specified for SDN-IIoT. Moreover, 5G/6G technology is a core element that enables the fast communication network in SDN-IIoT; however, detecting attackers and protecting the fast communication network in the presence of eavesdropping is the main issue that needs to be addressed in the upcoming years.

Effective implementation of Software-Defined Wireless Network in IIoT (SDWN-IIoT) is still an issue due to lack of standardization. In this context, the development of standards for SDWN-IIoT will induce vast opportunities for utilizing 5G/6G in SDWN-IIoT. Energy harvesting and spectrum sharing also pose a significant issue in SDWN-IIoT. Therefore, the deployment of 5G/6G in SDWN-IIoT will bring new solutions for spectrum sharing, which can significantly enhance the coverage where high-speed IIoT nodes and high-speed data plane switches can communicate seamlessly. Moreover, there is an urgent need for the SDWN-IIoT service infrastructure that can fully leverage the capabilities offered by high-speed 5G/6G networks.

12.13. Network slicing in SDN-IIoT

Network slicing technology has been approved as a potential method to sustain different services for SDN-IIoT. Since it ensures adequate performance in order to meet several requirements corresponding to the system visualization (Wu et al., 2022). However, the issue of determining where slices should be located and the precision number of slice instances that are required to deploy based on the number of distributed SDN controllers is so far not well solved in SDN-IIoT. Moreover, we know that network slicing will be crucial for 5G and SDN-IIoT innovation, but an absence of industry consensus prevails on how to implement network slicing is an issue that needs to be flushed out as well. The latency issue also needs to be addressed for guaranteeing that network slicing does not affect 5G Ultra-Reliable Low Latency Communications (5G URLLC) as 5G comes to materialization.

12.14. Mobility management in SDN-IIoT

The mobility of IIoT nodes has a negative impact on the operation of the overall system. If the nodes are mobile, rotation cycles will not work perfectly due to the topology change every node requires to wake up often and update the topology in accordance with the most recent wireless network state in order to maintain efficient dynamic connectivity. Furthermore, the capacities of links, network parameters, forwarding routes, and placement of distributed SDN controllers are all affected by movement initiated by mobility. The mobility of IIoT nodes can lead to data loss and increases latency, which, in turn, increases power utilization. Therefore, it is significant to further develop frameworks in terms of energy efficiency, delay-sensitive, loss-sensitive, and well-dynamic distributed control plane placement strategy that adapts to the node mobility. Some SDN-IIoT are made up of heterogeneous mobile nodes. A key difficulty is to configure them in line with the rules enforced by the application layer. This needs important research before developing the unified mechanism. To this end, solutions from mobile Internet protocol (Mobile IP), self-tuning networks, and the Internet of Drones (IoD) (Derhab et al., 2023) can also be further considered in handling mobility issues in SDN-IIoT for overcoming the given challenges.

12.15. Applicability of wireless networks in SDN-IIoT

Due to the increasing diversity of wireless networks in smart manufacturing, there is a need to improve their capabilities to guarantee the same degree of reliability as provided in wired networks. Furthermore, the lack of deterministic networking for many types of wireless networks and the issues that seem to handle the spectrum need to be considered in the future. Dissimilar to other industrial wired networks, some industrial wireless networks have the challenge of QoS requirements in terms of data loss and high latencies, especially for mission-critical applications. Moreover, using SDN for fast reconfiguration of industrial wireless networks according to the change in connectivity to achieve seamless communication is an open challenge to be tackled.

12.16. SDN controller viewpoint

Research on the placement of controllers or other components deployed in the control layer is limited, mainly because many solutions use only a single controller. However, it becomes important to carefully consider the placement of different controllers in SDN-IIoT networks. In a single SDN-IIoT system, there will be many IIoT devices,

possibly ranging from hundreds to even thousands. To handle these large numbers, the scalability of controllers is crucial. This involves not only having scalable architectures, but also taking into account the storage, programming languages, and processing capacity of the controllers. The task of synchronizing SDN controllers and their policies is also an interesting research direction. Moreover, it is necessary to define the IIoT requirements and then use only the controllers that have the features and capabilities to meet those requirements. The utilization of 5G/6G wireless networks and the integration of NFV in SDN-IIoT may lead to the virtualization of controllers. Thus, virtualization of various controllers will improve the performance of SDN-IIoT systems; however, coordination among them is a complicated task. Likewise, the placement of virtualized SDN controllers in SDN-IIoT would be an interesting research topic.

13. Conclusion

The SDN technology is well-considered in the IIoT environment due to programmable networks that enable flexible network management by separating the control plane and the data plane. Hence, the term "SDN-IIoT" is the strategy of using SDN in IIoT to improve its performance. This survey bestows a review of the current research that utilizes SDN solutions to cope with IIoT challenges. Moreover, we display the difference between the SDN architecture and the traditional architecture, as well as discuss the merits provided by the SDN architecture for the IIoT environment. In this context, OpenFlow protocol, IIoT environment, and computing/networking in IIoT were explained in detail. We propose an intelligence SDN-IIoT architecture based on hierarchical distributed SDN controllers for further solutions on the smart factory side. Based on the SDN architecture, we demonstrate advanced and standard flow installation techniques, as well as show the differences between these techniques. We analyze multiple works related to the SDN-IIoT domain by considering different fields such as fault tolerance management, optimization of traffic routing, resource management, interoperability-based OPC UA, SDN-IIoT advanced testbed, SDN-IIoT-based DT, energy efficiency, real-time, and network security. In addition, we discuss some technologies used to enhance the functionality of SDN-IIoT such as NFV, network slicing, TSN, edge/fog/cloud computing, MUD, blockchain, etc. The positive impacts of incorporating AI/ML algorithms in SDN-IIoT were also considered. Furthermore, the basic criterion for picking the controller, one of the key components of the solution, has been presented. In this regard, the characteristics of the five most used SDN controllers in IIoT have been described. Despite numerous works proposing SDN solutions, this new technology is not adequate to cover all IIoT issues; since there are still multiple challenges that need to be solved in the future in order to enhance the implementation of SDN-IIoT. Accordingly, current problems and future research orientations in the field of SDN-IIoT were outlined. This survey has analyzed many references, which is more beneficial for researchers and students who want to become familiar with SDN and IIoT technologies.

For future work of this survey, a candidate architecture for SDN-IIoT will be extended by adding other technologies such as Virtual Machine (VL) and container, as well as adopting the federated learning structure in the design of the hierarchical distributed control plane. Last but not least, the following areas will be deeply analyzed in the future survey: the use of 5G/6G technology in SDN-IIoT, SDN-IIoT-based network slicing solutions, and the applicability of WTSN in SDN-IIoT.

CRediT authorship contribution statement

Nteziriza Nkerabahizi Josbert: Conceptualization, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Min Wei:** Methodology, Validation, Writing – original draft, Writing – review & editing. **Ping Wang:** Funding acquisition, Investigation, Project administration, Supervision. **Ahsan Rafiq:** Formal analysis, Methodology, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by the National Key Research and Development Program of China (2021YFB3301000) and the Chongqing Natural Science Foundation (CSTB2023NSCQ-LZX0123).

References

Aazam, Mohammad, Zeadally, Sherali, Harras, Khaled A., 2018. Deploying fog computing in industrial internet of things and industry 4.0. IEEE Trans. Ind. Inform. 14 (10), 4674–4682.

Abdellatif, Alaa Awad, Emam, Ahmed, Chiasserini, Carla-Fabiana, Mohamed, Amr, Jaoua, Ali, Ward, Rabab, 2019. Edge-based compression and classification for smart healthcare systems: Concept, implementation and evaluation. Expert Syst. Appl. 117, 1–14.

Abid, M.A., Afaqui, N., Khan, M.A., Akhtar, M.W., Malik, A.W., Munir, A., Ahmad, J., Shabir, B., 2022. Evolution towards smart and software-defined Internet of Things. AI 3 (1), 100–123.

Abou El Houda, Zakaria, Hafid, Abdelhakim Senhaji, Khoukhi, Lyes, 2021. A novel machine learning framework for advanced attack detection using sdn. In: 2021 IEEE Global Communications Conference. GLOBECOM, IEEE, pp. 1–6.

Afolabi, Ibrahim, Taleb, Tarik, Samdanis, Konstantinos, Ksentini, Adlen, Flinck, Hannu, 2018. Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. IEEE Commun. Surv. Tutor. 20 (3), 2429–2453.

Ahmad, Suhail, Mir, Ajaz Hussain, 2021. Scalability, consistency, reliability and security in SDN controllers: a survey of diverse SDN controllers. J. Netw. Syst. Manage. 29 (9), 1–59.

Ahmed, Khandakar, Blech, Jan Olaf, Gregory, Mark A., Schmidt, Heinrich, 2015. Software defined networking for communication and control of cyber-physical systems. In: 2015 IEEE 21st International Conference on Parallel and Distributed Systems. ICPADS, IEEE, pp. 803–808.

Ahmed, Khandakar, Blech, Jan O., Gregory, Mark A., Schmidt, Heinz W., 2018. Software defined networks in industrial automation. J. Sensor Actuator Netw. 7 (3), 33.

Ahuja, Sanjay P., Deval, Niharika, 2021. From cloud computing to fog computing: Platforms for the Internet of Things (IoT). Res. Anthol. Archit. Framew. Integr. Strateg. Distrib. Cloud Comput. 999–1010.

Ahuja, Nisha, Singal, Gaurav, Mukhopadhyay, Debajyoti, 2020. DDOS attack SDN dataset. [Online]. Available: https://data.mendeley.com/datasets/jxpfjc64kr/1. (Accessed 8 December 2023).

Åkerberg, Johan, Gidlund, Mikael, Björkman, Mats, 2011. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In: 2011 9th IEEE International Conference on Industrial Informatics. IEEE, pp. 410–415.

Al-Rubaye, Saba, Kadhum, Ekhlas, Ni, Qiang, Anpalagan, Alagan, 2017. Industrial Internet of Things driven by SDN platform for smart grid resiliency. IEEE Internet Things J. 6 (1), 267–277.

Alam, Iqbal, Sharif, Kashif, Li, Fan, Latif, Zohaib, Karim, Md Monjurul, Biswas, Sujit, Nour, Boubakr, Wang, Yu, 2020. A survey of network virtualization techniques for internet of things using SDN and NFV. ACM Comput. Surv. 53 (2), 1–40.

Algarni, Sultan, Eassa, Fathy, Almarhabi, Khalid, Algarni, Abdullah, Albeshri, Aiiad, 2022. BCNBI: A blockchain-based security framework for northbound interface in software-defined networking. Electronics 11 (7), 996.

Alharbi, Talal, 2020. Deployment of blockchain technology in software defined networks: A survey. IEEE Access 8, 9146–9156.

Alhijawi, Bushra, Almajali, Sufyan, Elgala, Hany, Salameh, Haythem Bany, Ayyash, Moussa, 2022. A survey on DoS/DDoS mitigation techniques in SDNs: Classification, comparison, solutions, testing tools and datasets. Comput. Electr. Eng. 99, 107706.

Alhilali, Ahmed Hazim, Montazerolghaem, Ahmadreza, 2023. Artificial intelligence based load balancing in SDN: A comprehensive survey. Internet Things 22, 100814.

Ali, Ikbal, Hussain, S.M. Suhail, 2017. Control and management of distribution system with integrated DERs via IEC 61850 based communication. Eng. Sci. Technol., Int. J. 20 (3), 956–964.

Ali, Jehad, Jhaveri, Rutvij H., Alswailim, Mohannad, Roh, Byeong-hee, 2023. ESCALB: An effective slave controller allocation-based load balancing scheme for multi-domain SDN-enabled-IoT networks. J. King Saud Univ.-Comput. Inf. Sci. 35 (6), 101566.

Ali, Yasir, Khan, Habib Ullah, 2022. GTM approach towards engineering a features-oriented evaluation framework for secure authentication in IIoT environment. Comput. Ind. Eng. 168, 108119.

Aljohani, Sarah L., Alenazi, Mohammed J.F., 2021. MPResiSDN: Multipath resilient routing scheme for SDN-enabled smart cities networks. Appl. Sci. 11 (4), 1900.

Almasan, Paul, Suárez-Varela, José, Rusek, Krzysztof, Barlet-Ros, Pere, Cabellos-Aparicio, Albert, 2022. Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. Comput. Commun. 196, 184–194.

Alotaibi, Asma, Barnawi, Ahmed, 2023. IDSoft: A federated and softwarized intrusion detection framework for massive Internet of Things in 6G network. J. King Saud Univ.-Comput. Inf. Sci. 35 (6), 101575.

Alshahrani, Hani, Khan, Attiya, Rizwan, Muhammad, Reshan, Mana Saleh Al, Sulaiman, Adel, Shaikh, Asadullah, 2023. Intrusion detection framework for Industrial Internet of Things using software defined network. Sustainability 15 (11), 9001.

Anon, 2023. IEEE 802.1 Working Group, Time-sensitive networking task group. [Online]. Available: http://www.ieee802.org/1/pages/tsn.html. (Accessed 25 December 2023).

Anon, 2024. Pantou: OpenFlow 1.3 for OpenWRT. [Online]. Available: https://github.com/CPqD/ofsoftswitch13/wiki/OpenFlow-1.3-forOpenWRT. (Accessed 13 April 2024).

Antonenko, Vitaly, Smelyanskiy, Ruslan, 2013. Global network modelling based on mininet approach. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. pp. 145–146.

Arthurs, Peter, Gillam, Lee, Krause, Paul, Wang, Ning, Halder, Kaushik, Mouzakitis, Alexandros, 2021. A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles. IEEE Trans. Intell. Transp. Syst. 23 (7), 6206–6221.

Ashjaei, Mohammad, Girs, Svetlana, 2020. Dynamic resource distribution using SDN in wireless networks. In: 2020 IEEE International Conference on Industrial Technology. ICIT, IEEE, pp. 967–972.

Assefa, Beakal Gizachew, Özkasap, Öznur, 2019. A survey of energy efficiency in SDN: Software-based methods and optimization models. J. Netw. Comput. Appl. 137, 127–143.

Atharvan, Gannu, Koolikkara Madom Krishnamoorthy, Sreelakshmi, Dua, Amit, Gupta, Shashank, 2022. A way forward towards a technology-driven development of industry 4.0 using big data analytics in 5G-enabled IIoT. Int. J. Commun. Syst. 35 (1), e5014.

Aujla, Gagangeet Singh, Singh, Maninderpal, Bose, Arnab, Kumar, Neeraj, Han, Guangjie, Buyya, Rajkumar, 2020. Blocksdn: Blockchain-as-a-service for software defined networking in smart city applications. IEEE Netw. 34 (2), 83–91.

Aujla, Gagangeet Singh, Singh, Amritpal, Kumar, Neeraj, 2019. Adaptflow: Adaptive flow forwarding scheme for software-defined industrial networks. IEEE Internet Things J. 7 (7), 5843–5851.

Babbar, Himanshi, Rani, Shalli, Singh, Aman, Abd-Elnaby, Mohammed, Choi, Bong Jun, 2021. Cloud based smart city services for industrial Internet of Things in software-defined networking. Sustainability 13 (16), 8910.

Babiceanu, Radu F., Seker, Remzi, 2019. Cyber resilience protection for Industrial Internet of Things: A software-defined networking approach. Comput. Ind. 104, 47–58.

Babiker Mohamed, Monzir, Matthew Alofe, Olasunkanmi, Ajmal Azad, Muhammad, Singh Lallie, Harjinder, Fatema, Kaniz, Sharif, Tahir, 2022. A comprehensive survey on secure software-defined network for the internet of things. Trans. Emerg. Telecommun. Technol. 33 (1), e4391.

Baddeley, Michael, Nejabati, Reza, Oikonomou, George, Gormus, Sedat, Sooriyabandara, Mahesh, Simeonidou, Dimitra, 2017. Isolating SDN control traffic with layer-2 slicing in 6TiSCH industrial IoT networks. In: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks. NFV-SDN, IEEE, pp. 247–251.

Bakhshi Kiadehi, Katayoun, Rahmani, Amir Masoud, Sabbagh Molahosseini, Amir, 2021. A fault-tolerant architecture for internet-of-things based on software-defined networks. Telecommun. Syst. 77 (1), 155–169.

Balasubramanian, Venkatraman, Aloqaily, Moayad, Reisslein, Martin, 2021. An SDN architecture for time sensitive industrial IoT. Comput. Netw. 186, 107739.

Balasubramanian, Venkatraman, Aloqaily, Moayad, Reisslein, Martin, 2023. Fed-TSN: Joint failure probability based federated learning for fault-tolerant time-sensitive networks. IEEE Trans. Netw. Serv. Manag. 20 (2), 1470–1486.

Bansal, Sharu, Kumar, Dilip, 2020. IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication. Int. J. Wirel. Inf. Netw. 27, 340–364.

Barakabitze, Alcardo Alex, Ahmad, Arslan, Mijumbi, Rashid, Hines, Andrew, 2020. 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. Comput. Netw. 167, 106984.

Barakabitze, Alcardo Alex, Walshe, Ray, 2022. SDN and NFV for QoE-driven multimedia services delivery: The road towards 6G and beyond networks. Comput. Netw. 214, 109133.

Barricelli, Barbara Rita, Casiraghi, Elena, Fogli, Daniela, 2019. A survey on digital twin: Definitions, characteristics, applications, and design implications. IEEE Access 7, 167653–167671.

Basir, Rabeea, Qaisar, Saad, Ali, Mudassar, Aldwairi, Monther, Ashraf, Muhammad Ikram, Mahmood, Aamir, Gidlund, Mikael, 2019. Fog computing enabling industrial Internet of Things: State-of-the-art and research challenges. Sensors 19 (21), 4807.

Becker, Matthias, Lu, Zhonghai, Chen, De-Jiu, 2019. An adaptive resource provisioning scheme for industrial SDN networks. In: 2019 IEEE 17th International Conference on Industrial Informatics, Vol. 1. INDIN, IEEE, pp. 877–880.

Bedhief, Intidhar, Foschini, Luca, Bellavista, Paolo, Kassar, Meriem, Aguili, Taoufik, 2019. Toward self-adaptive software defined fog networking architecture for IIoT and industry 4.0. In: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks. CAMAD, IEEE, pp. 1–5.

Bekri, Wiem, Jmal, Rihab, Chaari Fourati, Lamia, 2020. Internet of things management based on software defined networking: a survey. Int. J. Wirel. Inf. Netw. 27, 385–410.

Bello, Lucia Lo, Lombardo, Alfio, Milardo, Sebastiano, Patti, Gaetano, Reno, Marco, 2020. Experimental assessments and analysis of an SDN framework to integrate mobility management in industrial wireless sensor networks. IEEE Trans. Ind. Inform. 16 (8), 5586–5595.

Bennis, Mehdi, Debbah, Mérouane, Poor, H. Vincent, 2018. Ultrareliable and low-latency wireless communication: Tail, risk, and scale. Proc. IEEE 106 (10), 1834–1853.

Beshley, Mykola, Klymash, Mikhailo, Scherm, Ilona, Beshley, Halyna, Shkoropad, Yuriy, 2022. Emerging network technologies for digital transformation: 5G/6G, IoT, SDN/IBN, cloud computing, and blockchain. In: IEEE Lnternational Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering. Springer, pp. 1–20.

Bi, Yuanguo, Han, Guangjie, Lin, Chuan, Peng, Yan, Pu, Huayan, Jia, Yazhou, 2019. Intelligent quality of service aware traffic forwarding for software-defined networking/open shortest path first hybrid industrial internet. IEEE Trans. Ind. Inform. 16 (2), 1395–1405.

Bianchi, Giuseppe, Bonola, Marco, Capone, Antonio, Cascone, Carmelo, 2014. Openstate: Programming platform-independent stateful openflow applications inside the switch. ACM SIGCOMM Comput. Commun. Rev. 44 (2), 44–51.

bin Salleh, Rosli, Koubaa, Anis, Khan, Zahid, Khan, Muhammad Khurram, Ali, Ihsan, et al., 2023. Data plane failure and its recovery techniques in SDN: A systematic literature review. J. King Saud Univ.-Comput. Inf. Sci. 35 (3), 176–201.

Blenk, Andreas, Basta, Arsany, Reisslein, Martin, Kellerer, Wolfgang, 2016. Survey on network virtualization hypervisors for software defined networking. IEEE Commun. Surv. Tutor. 18 (1), 655–685.

Bolla, R., Bruschi, R., Davoli, F., Di Gregorio, L., Donadio, P., Fialho, L., Collier, M., Lombardo, A., Recupero, D.R., Szemethy, T., 2013. The green abstraction layer: A standard power-management interface for next-generation network devices. IEEE Internet Comput. 17 (2), 82–90.

Boobalan, Parimala, Ramu, Swarna Priya, Pham, Quoc-Viet, Dev, Kapal, Pandya, Sharnil, Maddikunta, Praveen Kumar Reddy, Gadekallu, Thippa Reddy, Huynh-The, Thien, 2022. Fusion of federated learning and industrial Internet of Things: A survey. Comput. Netw. 212, 109048.

Bradai, Abbas, Rehmani, Mubashir Husain, Haque, Israat, Nogueira, Michele, Bukhari, Syed Hashim Raza, 2020. Software-defined networking (SDN) and network function virtualization (NFV) for a hyperconnected world: Challenges, applications, and major advancements. J. Netw. Syst. Manage. 28 (3), 433–435.

Bueno, Guilherme, Saquetti, Mateus, Rodrigues, Pablo, Lamb, Ivan, Gaspary, Luciano, Luizelli, Marcelo C., Zhani, Mohamed Faten, Azambuja, José Rodrigo, Cordeiro, Weverton, 2022. Managing virtual programmable switches: Principles, requirements, and design directions. IEEE Commun. Mag. 60 (2), 53–59.

Caiza, Gustavo, Chiliquinga, Santiago, Manzano, Santiago, Garcia, Marcelo V., 2020. Software-Defined Network (SDN) based Internet of Things within the context of low-cost automation. In: 2020 IEEE 18th International Conference on Industrial Informatics. INDIN, 1, IEEE, pp. 587–591.

Cao, Kun, Hu, Shiyan, Shi, Yang, Colombo, Armando Walter, Karnouskos, Stamatis, Li, Xin, 2021. A survey on edge and edge-cloud computing assisted cyber-physical systems. IEEE Trans. Ind. Inform. 17 (11), 7806–7819.

Cäsar, Matthias, Pawelke, Tobias, Steffan, Jan, Terhorst, Gabriel, 2022. A survey on bluetooth low energy security and privacy. Comput. Netw. 205, 1–18.

Cavalcanti, Dave, Perez-Ramirez, Javier, Rashid, Mohammad Mamunur, Fang, Juan, Galeev, Mikhail, Stanton, Kevin B., 2019. Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems. Proc. IEEE 107 (6), 1132–1152.

Chahed, Hamza, Kassler, Andreas J., 2021. Software-defined time sensitive networks configuration and management. In: 2021 IEEE Conference on Network Function Virtualization and Software Defined Networks. NFV-SDN, IEEE, pp. 124–128.

Chan, Min-Cheng, Chen, Chien, Huang, Jun-Xian, Kuo, Ted, Yen, Li-Hsing, Tseng, Chien-Chao, 2014. OpenNet: A simulator for software-defined wireless local area network. In: 2014 IEEE Wireless Communications and Networking Conference. WCNC, IEEE, pp. 3332–3336.

Chattopadhyay, Subhrendu, Chatterjee, Soumyajit, Nandi, Sukumar, Chakraborty, Sandip, 2020. Aloe: fault-tolerant network management and orchestration framework for IoT applications. IEEE Trans. Netw. Serv. Manag. 17 (4), 2396–2409.

Chaudhary, Rajat, Aujla, Gagangeet Singh, Garg, Sahil, Kumar, Neeraj, Rodrigues, Joel J.P.C., 2018. SDN-enabled multi-attribute-based secure communication for smart grid in IIoT environment. IEEE Trans. Ind. Inform. 14 (6), 2629–2640.

Chaudhary, Rajat, Aujla, Gagangeet Singh, Kumar, Neeraj, Chouhan, Pushpinder Kaur, 2022. A comprehensive survey on software-defined networking for smart communities. Int. J. Commun. Syst. e5296.

Chen, Junyan, Wang, Yong, Huang, Xuefeng, Xie, Xiaolan, Zhang, Hongmei, Lu, Xiaoye, 2022. ALBLP: adaptive load-balancing architecture based on link-state prediction in software-defined networking. Wirel. Commun. Mob. Comput. 2022, 1–16.

Chin, Wen-Long, Ko, Hsin-An, Chen, Ning-Wen, Chen, Pin-Wei, Jiang, Tao, 2023. Securing NFV/SDN IoT using vnfs over a compute-intensive hardware resource in NFVI. IEEE Netw. 1–8.

Chourasia, Shubhangi, Tyagi, Ankit, Pandey, S.M., Walia, R.S., Murtaza, Qasim, 2022. Sustainability of Industry 6.0 in global perspective: benefits and challenges. Mapan 37 (2), 443–452.

Chu, Teng-Wei, Shen, Chung-An, Wu, Chun-Wei, 2018. The hardware and software co-design of a configurable QoS for video streaming based on OpenFlow protocol and NetFPGA platform. Multimedia Tools Appl. 77, 9071–9091.

Czachórski, Tadeusz, Gelenbe, Erol, Kuaban, Godlove Suila, Marek, Dariusz, 2021. Time-dependent performance of a multi-hop software defined network. Appl. Sci. 11 (6), 1–21.

Das, Rohit Kumar, Ahmed, Nurzaman, Pohrmen, Fabiola Hazel, Maji, Arnab Kumar, Saha, Goutam, 2020. 6Le-sdn: an edge-based software-defined network for internet of things. IEEE Internet Things J. 7 (8), 7725–7733.

Das, Rohit Kumar, Jha, Monica, Harizan, Subash, 2022. Performance appraisal of 6LoWPAN and OpenFlow in SDN enabled edge-based IoT network. In: Advanced Computational Paradigms and Hybrid Intelligent Computing: Proceedings of ICACCP 2021. Springer, pp. 21–29.

Derhab, Abdelouahid, Cheikhrouhou, Omar, Allouch, Azza, Koubaa, Anis, Qureshi, Basit, Ferrag, Mohamed Amine, Maglaras, Leandros, Khan, Farrukh Aslam, 2023. Internet of drones security: Taxonomies, open issues, and future directions. Veh. Commun. 39, 100552.

Derhab, Abdelouahid, Guerroumi, Mohamed, Gumaei, Abdu, Maglaras, Leandros, Ferrag, Mohamed Amine, Mukherjee, Mithun, Khan, Farrukh Aslam, 2019. Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security. Sensors 19 (14), 3119.

Desai, Prasad Ramesh, Mini, S., Tosh, Deepak K., 2022. Edge-based optimal routing in SDN-enabled Industrial Internet of Things. IEEE Internet Things J. 9 (19), 18898–18907.

Devan, P. Arun Mozhi, Hussin, Fawnizu Azmadi, Ibrahim, Rosdiazli, Bingi, Kishore, Khanday, Farooq Ahmad, 2021. A survey on the application of WirelessHART for industrial process monitoring and control. Sensors 21 (15), 4951.

Docker, [Online]. Available: https://www.docker.com/. (Accessed 30 December 2023).

Dodson, Donna, Montgomery, Douglas, Polk, W., Ranganathan, Mudumbai, Souppaya, Murugiah, Johnson, Steve, Kadam, Ashwini, Pratt, Craig, Thakore, Darshak, Walker, Mark, et al., 2021. Securing Small Business and Home Internet of Things (IoT) Devices: Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD). Technical Report, NIST, SPECIAL PUBLICATION, pp. 1–947.

Dressler, Falko, Chiasserini, Carla Fabiana, Fitzek, Frank H.P., Karl, Holger, Cigno, Renato Lo, Capone, Antonio, Casetti, Claudio, Malandrino, Francesco, Mancuso, Vincenzo, Klingler, Florian, et al., 2022. V-edge: Virtual edge computing as an enabler for novel microservices and cooperative computing. IEEE Netw. 36 (3), 24–31.

Du, Miao, Wang, Kun, 2019. An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial Internet of Things. IEEE Trans. Ind. Inform. 16 (1), 648–657.

Dürkop, Lars, Jasperneite, Jürgen, Fay, Alexander, 2015. An analysis of real-time ethernets with regard to their automatic configuration. In: 2015 IEEE World Conference on Factory Communication Systems. WFCS, IEEE, pp. 1–8.

El-Hefnawy, Nancy Abbas, Raouf, Osama Abdel, Askr, Heba, 2022. Dynamic routing optimization algorithm for software defined networking. Comput. Mater. Continua 70 (1), 1349–1362.

Elamanov, Sherzod, Son, Hyeonseo, Flynn, Bob, Yoo, Seong Ki, Dilshad, Naqqash, Song, JaeSeung, 2022. Interworking between Modbus and internet of things platform for industrial services. Digit. Commun. Netw..

Etxezarreta, Xabier, Garitano, Iñaki, Iturbe, Mikel, Zurutuza, Urko, 2023. Software-defined networking approaches for intrusion response in industrial control systems: A survey. Int. J. Crit. Infrastruct. Prot. 42, 100615.

Fedullo, Tommaso, Morato, Alberto, Tramarin, Federico, Rovati, Luigi, Vitturi, Stefano, 2022. A comprehensive review on time sensitive networks with a special focus on its applicability to industrial smart and distributed measurement systems. Sensors 22 (4), 1638.

Ferrag, Mohamed Amine, Friha, Othmane, Hamouda, Djallel, Maglaras, Leandros, Janicke, Helge, 2022. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access 10, 40281–40306.

Ficzere, Dániel, Patel, Dhruvin, Sachs, Joachim, Ansari, Junaid, Soós, Gábor, Varga, Pál, 2022. 5G public network integration for a real-life PROFINET application. In: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. IEEE, pp. 1–5.

Firouzi, Ramin, Rahmani, Rahim, 2022. A distributed SDN controller for distributed IoT. IEEE Access 10, 42873–42882.

Fonseca, Paulo César, Mota, Edjard Souza, 2017. A survey on fault management in software-defined networks. IEEE Commun. Surv. Tutor. 19 (4), 2284–2321.

Fontes, Fernando, Rocha, Bruno, Mota, Alexandre, Pedreiras, Paulo, Silva, Valter, 2020. Extending mqtt-sn with real-time communication services. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation, Vol. 1. IEEE, pp. 1–4.

Fontes, Ramon dos Reis, Rothenberg, Christian Esteve, 2016. Mininet-wifi: A platform for hybrid physical-virtual software-defined wireless networking research. In: Proceedings of the 2016 ACM SIGCOMM Conference. pp. 607–608.

Foukalas, Fotis, Tziouvaras, Athanasios, 2021. Edge artificial intelligence for industrial Internet of Things applications: an industrial edge intelligence solution. IEEE Ind. Electron. Mag. 15 (2), 28–36.

Gao, Ying, Chen, Yijian, Hu, Xiping, Lin, Hongliang, Liu, Yangliang, Nie, Laisen, 2020a. Blockchain based IIoT data sharing framework for SDN-enabled pervasive edge computing. IEEE Trans. Ind. Inform. 17 (7), 5041–5049.

Gao, Honghao, Qin, Xi, Barroso, Ramon J. Duran, Hussain, Walayat, Xu, Yueshen, Yin, Yuyu, 2020b. Collaborative learning-based industrial IoT API recommendation for software-defined devices: the implicit knowledge discovery perspective. IEEE Trans. Emerg. Top. Comput. Intell. 6 (1), 66–76.

García, Sara Nieves Matheu, Molina Zarca, Alejandro, Hernández-Ramos, José Luis, Bernabé, Jorge Bernal, Gómez, Antonio Skarmeta, 2019. Enforcing behavioral profiles through software-defined networks in the industrial Internet of Things. Appl. Sci. 9 (21), 4576.

Girs, Svetlana, Ashiaei, Mohammad, 2018. Designing a bandwidth management scheme for heterogeneous virtualized networks. In: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation, Vol. 1. ETFA, IEEE, pp. 1079–1082.

Golightly, Lewis, Modesti, Paolo, Garcia, Rémi, Chang, Victor, 2023. Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN. Cyber Secur. Appl. 1, 1–20.

Google Cloud Computing, Hosting Services & APIs |, [Online]. Available: https://cloud.google.com/. (Accessed 13 April 2024).

Goudarzi, Mohammad, Palaniswami, Marimuthu, Buyya, Rajkumar, 2022. Scheduling IoT applications in edge and fog computing environments: a taxonomy and future directions. ACM Comput. Surv. 55 (7), 1–41.

GSMA Spec, 2014. The global telecom tower esco market overview of the global market for energy to telecom towers in off-grid and bad-grid areas. 1–52, [Online]. Available: https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2015/01/140617-GSMA-report-draft-vF-KR-v7.pdf. (Accessed 29 December 2023).

Guck, Jochen W., Van Bemten, Amaury, Reisslein, Martin, Kellerer, Wolfgang, 2017. Unicast QoS routing algorithms for SDN: A comprehensive survey and performance evaluation. IEEE Commun. Surv. Tutor. 20 (1), 388–415.

Gupta, Bulbul, Mittal, Pooja, Mufti, Tabish, 2021. A review on amazon web service (aws), microsoft azure & google cloud platform (gcp) services. In: Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development. ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India.

Haghnegahdar, Lida, Joshi, Sameehan S., Dahotre, Narendra B., 2022. From IoT-based cloud manufacturing approach to intelligent additive manufacturing: Industrial Internet of Things—An overview. Int. J. Adv. Manuf. Technol. 1–18.

Haji, Saad H., Zeebaree, Subhi R.M., Saeed, Rezgar Hasan, Ameen, Siddeeq Y., Shukur, Hanan M., Omar, Naaman, Sadeeq, Mohammed A.M., Ageed, Zainab Salih, Ibrahim, Ibrahim Mahmood, Yasin, Hajar Maseeh, 2021. Comparison of software defined networking with traditional networking. Asian J. Res. Comput. Sci. 9 (2), 1–18.

Hakak, Saqib, Khan, Wazir Zada, Gilkar, Gulshan Amin, Imran, Muhammad, Guizani, Nadra, 2020. Securing smart cities through blockchain technology: Architecture, requirements, and challenges. IEEE Netw. 34 (1), 8–14.

Haleplidis, Evangelos, Pentikousis, Kostas, Denazis, Spyros, Salim, J. Hadi, Meyer, David, Koufopavlou, Odysseas, 2015a. Software-defined networking (SDN): Layers and architecture terminology. pp. 1–35, Internet Eng. Task Force (IETF), No. RFC7426, Technical report.

Haleplidis, Evangelos, Salim, Jamal Hadi, Halpern, Joel M., Hares, Susan, Pentikousis, Kostas, Ogawa, Kentaro, Wang, Weiming, Denazis, Spyros, Koufopavlou, Odysseas, 2015b. Network programmability with ForCES. IEEE Commun. Surv. Tutor. 17 (3), 1423–1440.

Hamza, Ayyoob, Gharakheili, Hassan Habibi, Benson, Theophilus A., Sivaraman, Vijay, 2019. Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity. In: Proceedings of the 2019 ACM Symposium on SDN Research. pp. 36–48.

Hamza, Ayyoob, Gharakheili, Hassan Habibi, Sivaraman, Vijay, 2018. Combining MUD policies with SDN for IoT intrusion detection. In: Proceedings of the 2018 Workshop on IoT Security and Privacy. pp. 1–7.

Han, Guangjie, Abu-Mahfouz, Adnan M, Rodrigues, Joel J.P.C., Wang, Xianbin, 2022. Guest editorial: AI-enabled software-defined industrial networks: Architectures, algorithms, and applications. IEEE Trans. Ind. Inform. 18 (6), 4210–4214.

Hauser, Frederik, Häberle, Marco, Merling, Daniel, Lindner, Steffen, Gurevich, Vladimir, Zeiger, Florian, Frank, Reinhard, Menth, Michael, 2023. A survey on data plane programming with p4: Fundamentals, advances, and applied research. J. Netw. Comput. Appl. 212, 103561.

Hou, Ting-Chao, Liu, Lin-Hung, Lan, Yan-Kai, Chen, Yi-Ting, Chu, Yuan-Sun, 2022. An improved network time protocol for industrial internet of things. Sensors 22 (13), 1–15.

Hu, Bing, Bi, Yuanguo, Zhi, Mingjian, Zhang, Kuan, Yan, Feihong, Zhang, Qian, Liu, Zheng, 2021. A deep one-class intrusion detection scheme in software-defined industrial networks. IEEE Trans. Ind. Inform. 18 (6), 4286–4296.

Huo, Ru, Yu, Fei Richard, Huang, Tao, Xie, Renchao, Liu, Jiang, Leung, Victor C.M., Liu, Yunjie, 2016. Software defined networking, caching, and computing for green wireless networks. IEEE Commun. Mag. 54 (11), 185–193.

Huo, Ru, Zeng, Shiqin, Wang, Zhihao, Shang, Jiajia, Chen, Wei, Huang, Tao, Wang, Shuo, Yu, F. Richard, Liu, Yunjie, 2022a. A comprehensive survey on blockchain in Industrial Internet of Things: Motivations, research progresses, and future challenges. IEEE Commun. Surv. Tutor. 24 (1), 88–122.

Huo, Ru, Zeng, Shiqin, Wang, Zhihao, Shang, Jiajia, Chen, Wei, Huang, Tao, Wang, Shuo, Yu, F. Richard, Liu, Yunjie, 2022b. A comprehensive survey on blockchain in industrial internet of things: Motivations, research progresses, and future challenges. IEEE Commun. Surv. Tutor. 24 (1), 88–122.

Hussain, Mudassar, Shah, Nadir, Amin, Rashid, Alshamrani, Sultan S., Alotaibi, Aziz, Raza, Syed Mohsan, 2022. Software-defined networking: Categories, analysis, and future directions. Sensors 22 (15), 1–47.

Imran, Ghaffar, Zeba, Alshahrani, Abdullah, Fayaz, Muhammad, Alghamdi, Ahmed Mohammed, Gwak, Jeonghwan, 2021. A topical review on machine learning, software defined networking, Internet of Things applications: Research limitations and challenges. Electronics 10 (8), 1–28.

Isong, Bassey, Molose, Reorapetse Ramoliti Samuel, Abu-Mahfouz, Adnan M., Dladlu, Nosipho, 2020. Comprehensive review of SDN controller placement strategies. IEEE Access 8, 170070–170092.

Ja'afreh, Mohammed Al, Adhami, Hikmat, Alchalabi, Alaa Eddin, Hoda, Mohamed, El Saddik, Abdulmotaleb, 2022. Toward integrating software defined networks with the Internet of Things: a review. Cluster Comput. 1–18.

Jang, Hung-Chin, Lin, Jian-Ting, 2019. Bandwidth management framework for smart homes using SDN: Isp perspective. Int. J. Internet Protoc. Technol. 12 (2), 110–120.

Javed, Farhana, Antevski, Kiril, Mangues-Bafalluy, Josep, Giupponi, Lorenza, Bernardos, Carlos J., 2022. Distributed ledger technologies for network slicing: A survey. IEEE Access 10, 19412–19442.

Jawed, Md Saquib, Sajid, Mohammad, 2022. A comprehensive survey on cloud computing: architecture, tools, technologies, and open issues. Int. J. Cloud Appl. Comput. (IJCAC) 12 (1), 1–33.

Jayalaxmi, P.L.S., Saha, Rahul, Kumar, Gulshan, Kim, Tai-Hoon, 2022. Machine and deep learning amalgamation for feature extraction in Industrial Internet-of-Things. Comput. Electr. Eng. 97, 1–14.

Jecan, Eusebiu, Pop, Catalin, Ratiu, Ovidiu, Puschita, Emanuel, 2022. Predictive energy-aware routing solution for industrial IoT evaluated on a WSN hardware platform. Sensors 22 (6), 2107.

Jhanjhi, N.Z., Humayun, Mamoona, Almuayqil, Saleh N., 2021. Cyber security and privacy issues in Industrial Internet of Things. Comput. Syst. Sci. Eng. 37 (3), 361–380.

Jhaveri, Rutvij H., Ramani, Sagar V., Srivastava, Gautam, Gadekallu, Thippa Reddy, Aggarwal, Vaneet, 2021. Fault-resilience for bandwidth management in industrial software-defined networks. IEEE Trans. Netw. Sci. Eng. 8 (4), 3129–3139.

Ji, Luyue, He, Shibo, Wu, Wenjie, Gu, Chaojie, Bi, Jichao, Shi, Zhiguo, 2021. Dynamic network slicing orchestration for remote adaptation and configuration in industrial IoT. IEEE Trans. Ind. Inform. 18 (6), 4297–4307.

Jiang, Shan, Cao, Jiannong, Wu, Hanqing, Yang, Yanni, 2020. Fairness-based packing of industrial IoT data in permissioned blockchains. IEEE Trans. Ind. Inform. 17 (11), 7639–7649.

Jiang, Jehn-Ruey, Huang, Hsin-Wen, Liao, Ji-Hau, Chen, Szu-Yuan, 2014. Extending Dijkstra's shortest path algorithm for software defined networking. In: The 16th Asia-Pacific Network Operations and Management Symposium. IEEE, pp. 1–4.

Jiang, Jinfang, Lin, Chuan, Han, Guangjie, Abu-Mahfouz, Adnan M, Shah, Syed Bilal Hussain, Martínez-García, Miguel, 2022. How AI-enabled SDN technologies improve the security and functionality of industrial IoT network: Architectures, enabling technologies, and opportunities. Digit. Commun. Netw. 1–18.

Jin, Xi, Xia, Changqing, Guan, Nan, Zeng, Peng, 2021. Joint algorithm of message fragmentation and no-wait scheduling for time-sensitive networks. IEEE/CAA J. Autom. Sin. 8 (2), 478–490.

Josbert, Nteziriza Nkerabahizi, Joyce, Harubwira Nyampinga, Wang, Jun, Bosco, Musabe Jean, 2021a. End-to-end QoS routing scheme in Industrial Internet of Things managed by software-defined networking platform. In: 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology. CEI, IEEE, pp. 542–549.

Josbert, Nteziriza Nkerabahizi, Ping, Wang, Wei, Min, Li, Yong, 2021b. Industrial networks driven by SDN technology for dynamic fast resilience. Information 12 (10), 1–30.

Josbert, Nteziriza Nkerabahizi, Ping, Wang, Wei, Min, Muthanna, Mohammed Saleh Ali, Rafiq, Ahsan, 2021c. A framework for managing dynamic routing in industrial networks driven by software-defined networking technology. IEEE Access 9, 74343–74359.

Josbert, Nteziriza Nkerabahizi, Ping, Wang, Wei, Min, Rafiq, Ahsan, 2021d. Solution for industrial networks: Resilience-based sdn technology. In: 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering. ICBAIE, IEEE, pp. 392–400.

Kafetzis, Dimitrios, Vassilaras, Spyridon, Vardoulias, Georgios, Koutsopoulos, Iordanis, 2022. Software-defined networking meets software-defined radio in mobile ad hoc networks: state of the art and future directions. IEEE Access 10, 9989–10014.

Kalita, Alakesh, Khatua, Manas, 2022. 6Tisch–ipv6 enabled open stack iot network formation: A review. ACM Trans. Internet Things 3 (3), 1–36.

Kaljic, Enio, Maric, Almir, Njemcevic, Pamela, Hadzialic, Mesud, 2019. A survey on data plane flexibility and programmability in software-defined networking. IEEE Access 7, 47804–47840.

Kang, Yoohwa, Lee, Sunwoo, Gwak, Songi, Kim, Taekyeong, An, Donghyeok, 2021. Time-sensitive networking technologies for industrial automation in wireless communication systems. Energies 14 (15), 4497.

Kaur, Kuljeet, Garg, Sahil, Aujla, Gagangeet Singh, Kumar, Neeraj, Rodrigues, Joel J.P.C., Guizani, Mohsen, 2018. Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay. IEEE Commun. Mag. 56 (2), 44–51.

Khan, Habib Ullah, Ali, Farhad, Nazir, Shah, 2024. Systematic analysis of software development in cloud computing perceptions. J. Softw.: Evol. Process 36 (2), e2485.

Khan, Abdullah Ayub, Bourouis, Sami, Kamruzzaman, M.M., Hadjouni, Myriam, Shaikh, Zaffar Ahmed, Laghari, Asif Ali, Elmannai, Hela, Dhahbi, Sami, 2023. Data security in healthcare Industrial Internet of Things with blockchain. IEEE Sens. J. 23 (20), 25144–25151.

Khan, Wazir Zada, Rehman, M.H., Zangoti, Hussein Mohammed, Afzal, Muhammad Khalil, Armi, Nasrullah, Salah, Khaled, 2020. Industrial Internet of Things: Recent advances, enabling technologies and open challenges. Comput. Electr. Eng. 81, 106522.

Kherbache, Mehdi, Maimour, Moufida, Rondeau, Eric, 2021. When digital twin meets network softwarization in the industrial IoT: real-time requirements case study. Sensors 21 (24), 1–17.

Khondoker, Rahamatullah, Zaalouk, Adel, Marx, Ronald, Bayarou, Kpatcha, 2014. Feature-based comparison and selection of software defined networking (SDN) controllers. In: 2014 World Congress on Computer Applications and Information Systems. WCCAIS, IEEE, pp. 1–7.

Khorsandroo, Sajad, Sánchez, Adrián Gallego, Tosun, Ali Saman, Arco, José M., Doriguzzi-Corin, Roberto, 2021. Hybrid SDN evolution: A comprehensive survey of the state-of-the-art. Comput. Netw. 192, 1–40.

Kiadehi, Katayoun Bakhshi, Rahmani, Amir Masoud, Molahosseini, Amir Sabbagh, 2021. Increasing fault tolerance of data plane on the internet of things using the software-defined networks. PeerJ Comput. Sci. 7, e543.

Kim, Yong-hwan, Gil, Joon-Min, Kim, Dongkyun, 2021. A location-aware network virtualization and reconfiguration for 5G core network based on SDN and NFV. Int. J. Commun. Syst. 34 (2), e4160.

Kipongo, J., Esenogho, E., Swart, T.G., 2022. Efficient topology discovery protocol using IT-SDN for software-defined wireless sensor network. Bull. Electr. Eng. Inform. 11, 256–269.

Klimis, Vasileios, 2021. Abstractions and Optimisations for Model-Checking Software-Defined Networks (Ph.D. thesis). University of Sussex, Falmer, Bright, UK.

Kobzan, Thomas, Blöcher, Immanuel, Hendel, Maximilian, Althoff, Simon, Gerhard, Alissa, Schriegel, Sebastian, Jasperneite, Jürgen, 2020. Configuration solution for TSN-based industrial networks utilizing SDN and OPC UA. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation. ETFA, 1, IEEE, pp. 1629–1636.

Krishnan, Prabhakar, Jain, Kurunandan, Achuthan, Krishnashree, Buyya, Rajkumar, 2021a. Software-defined security-by-contract for blockchain-enabled MUD-aware industrial IoT edge networks. IEEE Trans. Ind. Inform. 18 (10), 7068–7076.

Krishnan, Prabhakar, Jain, Kurunandan, Buyya, Rajkumar, Vijayakumar, Pandi, Nayyar, Anand, Bilal, Muhammad, Song, Houbing, 2021b. MUD-based behavioral profiling security framework for software-defined IoT networks. IEEE Internet Things J. 9 (9), 6611–6622.

Kubernetes, [Online]. Available: https://kubernetes.io/. (Accessed 30 December 2023).

Kunz, Thomas, Muthukumar, Karpakamurthy, 2017. Comparing OpenFlow and NETCONF when interconnecting data centers. In: 2017 IEEE 25th International Conference on Network Protocols. ICNP, IEEE, pp. 1–6.

Kutuzov, D., Osovsky, A., Stukach, O., Starov, D., 2021. Modeling of IIoT traffic processing by intra-chip NoC routers of 5G/6G networks. In: 2021 International Siberian Conference on Control and Communications. SIBCON, IEEE, pp. 1–5.

Lantz, Bob, Heller, Brandon, McKeown, Nick, 2010. A network in a laptop: rapid prototyping for software-defined networks. In: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. pp. 1–6.

Lantz, Bob, O'Connor, Brian, 2015. A mininet-based virtual testbed for distributed SDN development. ACM SIGCOMM Comput. Commun. Rev. 45 (4), 365–366.

Latif, Sohaib A., Wen, Fang B. Xian, Iwendi, Celestine, Li-Li, F. Wang, Mohsin, Syed Muhammad, Han, Zhaoyang, Band, Shahab S., 2022. AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems. Comput. Commun. 181, 274–283.

Lázaro, Jesús, Cabrejas, Jimena, Zuloaga, Aitzol, Muguira, Leire, Jiménez, Jaime, 2022. Time sensitive networking protocol implementation for linux end equipment. Technologies 10 (3), 1–11.

Lee, Seung-Yong, Sung, Minyoung, 2022. OPC-UA agent for legacy programmable logic controllers. Appl. Sci. 12 (17), 8859.

Leonardi, Luca, Lo Bello, Lucia, Aglianò, Simone, 2020. Priority-based bandwidth management in virtualized software-defined networks. Electronics 9 (6), 1009.

Li, Chenning, Cao, Zhichao, 2022. Lora networking techniques for large-scale and long-term iot: A down-to-top survey. ACM Comput. Surv. 55 (3), 1–36.

Li, Zhijun, He, Tian, 2017. Webee: Physical-layer cross-technology communication via emulation. In: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking. pp. 2–14.
Li, Tao, Hofmann, Christoph, Franz, Elke, 2020a. Secure and reliable data transmission in sdn-based backend networks of industrial iot. In: 2020 IEEE 45th Conference on Local Computer Networks. LCN, IEEE, pp. 365–368.
Li, Shengru, Hu, Daoyun, Fang, Wenjian, Ma, Shoujiang, Chen, Cen, Huang, Huibai, Zhu, Zuqing, 2017. Protocol oblivious forwarding (POF): Software-defined networking with enhanced programmability. IEEE Netw. 31 (2), 58–66.
Li, Shan, Iqbal, Muddesar, Saxena, Neetesh, 2022a. Future industry internet of things with zero-trust security. Inf. Syst. Front. 1–14.
Li, Xiaomin, Li, Di, Wan, Jiafu, Liu, Chengliang, Imran, Muhammad, 2018. Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing. IEEE Internet Things J. 5 (3), 1351–1360.
Li, Jinying, Maiti, Ananda, Fei, Jiangang, 2023. Features and scope of regulatory technologies: Challenges and opportunities with Industrial Internet of Things. Future Internet 15 (8), 256.
Li, Wenjuan, Meng, Weizhi, Kwok, Lam For, 2016a. A survey on OpenFlow-based software defined networks: Security challenges and countermeasures. J. Netw. Comput. Appl. 68, 126–139.
Li, Dong, Peng, Z., Ming, Y., Xueting, Y., Haibin, Y., 2016b. A framework of controller with flow table cache and performance analysis in software defined industrial networks. Rev. Técn. Fac. Ing. Univ. Zulia 39 (3), 208–215.
Li, Yuhong, Su, Xiang, Ding, Aaron Yi, Lindgren, Anders, Liu, Xiaoli, Prehofer, Christian, Riekki, Jukka, Rahmani, Rahim, Tarkoma, Sasu, Hui, Pan, 2020b. Enhancing the Internet of Things with knowledge-driven software-defined networking technology: Future perspectives. Sensors 20 (12), 1–20.
Li, Hongda, Wei, Feng, Hu, Hongxin, 2019. Enabling dynamic network access control with anomaly-based IDS and SDN. In: Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization. pp. 13–16.
Li, Xiaoyue, Zhou, Chaoqun, Liang, Zilong, Yu, Qiang, Chen, Xiankai, He, Zhiyuan, et al., 2022b. UCB-based route and power selection optimization for SDN-enabled industrial IoT in smart grid. Wirel. Commun. Mob. Comput. 2022.
Li, Dong, Zhou, Ming-Tuo, Zeng, Peng, Yang, Ming, Zhang, Yan, Yu, Haibin, 2016c. Green and reliable software-defined industrial networks. IEEE Commun. Mag. 54 (10), 30–37.
Liatifis, Athanasios, Sarigiannidis, Panagiotis, Argyriou, Vasileios, Lagkas, Thomas, 2023. Advancing sdn from openflow to p4: A survey. ACM Comput. Surv. 55 (9), 1–37.
Liu, Chao, Su, Ziwei, Xu, Xun, Lu, Yuqian, 2022. Service-oriented industrial internet of things gateway for cloud manufacturing. Robot. Comput.-Integr. Manuf. 73, 1–14.
Liu, Yazhi, Wu, Qianqian, Niu, Jianwei, Li, Xiong, Song, Zheng, 2021. Imitation learning based heavy-hitter scheduling scheme in software-defined industrial networks. IEEE Trans. Ind. Inform. 18 (6), 4254–4264.
Long, Qingyue, Chen, Yanliang, Zhang, Haijun, Lei, Xianfu, 2019. Software defined 5G and 6G networks: A survey. Mobile Netw. Appl. 27, 1–21.
López-Millán, Gabriel, Marín-López, Rafael, Pereñíguez-García, Fernando, Canovas, Oscar, Espín, José Antonio Parra, 2023. Analysis and practical validation of a standard SDN-based framework for ipsec management. Comput. Stand. Interfaces 83, 103665.
Luo, Jia, Yu, F. Richard, Chen, Qianbin, Tang, Lun, 2020. Blockchain-enabled software-defined industrial internet of things with deep recurrent q-network. In: ICC 2020-2020 IEEE International Conference on Communications. ICC, IEEE, pp. 1–6.
Lv, Zhihan, Han, Yang, Singh, Amit Kumar, Manogaran, Gunasekaran, Lv, Haibin, 2020. Trustworthiness in industrial IoT systems based on artificial intelligence. IEEE Trans. Ind. Inform. 17 (2), 1496–1504.
Lv, Zhihan, Qiao, Liang, Wu, Jingyi, Lv, Haibin, 2022. Advanced deep learning for image processing in industrial internet of things under software-defined network. In: Software Defined Internet of Everything, A chapter. Springer, pp. 271–294.
Lv, Zhihan, Xiu, Wenqun, 2019. Interaction of edge-cloud computing based on SDN and NFV for next generation IoT. IEEE Internet Things J. 7 (7), 5706–5712.
Ma, Xiaohang, Liao, Lingxia, Li, Zhi, Lai, Roy Xiaorong, Zhang, Miao, 2022. Applying federated learning in software-defined networks: A survey. Symmetry 14 (2), 195.
Maddikunta, Praveen Kumar Reddy, Pham, Quoc-Viet, Prabadevi, B., Deepa, Natarajan, Dev, Kapal, Gadekallu, Thippa Reddy, Ruby, Rukhsana, Liyanage, Madhusanka, 2022. Industry 5.0: A survey on enabling technologies and potential applications. J. Ind. Inf. Integr. 26, 1–19.
Madhawa, Surendar, Balakrishnan, P., Arumugam, Umamakeswari, 2018. Data driven intrusion detection system for software defined networking enabled industrial Internet of Things. J. Intell. Fuzzy Systems 34 (3), 1289–1300.
Mahmoodi, Toktam, Kulkarni, Vivek, Kellerer, Wolfgang, Mangan, Peter, Zeiger, Florian, Spirou, Spiros, Askoxylakis, Ioannis, Vilajosana, Xavier, Einsiedler, Hans Joachim, Quittek, Jürgen, 2016. VirtuWind: virtual and programmable industrial network prototype deployed in operational wind park. Trans. Emerg. Telecommun. Technol. 27 (9), 1281–1288.
Mahmoudi, Marjan, Avokh, Avid, Barekatain, Behrang, 2022. SDN-DVFS: an enhanced QoS-aware load-balancing method in software defined networks. Cluster Comput. 25 (2), 1237–1262.
Mai, Tianle, Yao, Haipeng, Zhang, Ni, He, Wenji, Guo, Dong, Guizani, Mohsen, 2021a. Transfer reinforcement learning aided distributed network slicing optimization in industrial IoT. IEEE Trans. Ind. Inform. 18 (6), 4308–4316.
Mai, Tianle, Yao, Haipeng, Zhang, Ni, He, Wenji, Guo, Dong, Guizani, Mohsen, 2021b. Transfer reinforcement learning aided distributed network slicing optimization in industrial IoT. IEEE Trans. Ind. Inform. 18 (6), 4308–4316.
Mamushiane, Lusani, Lysko, Albert, Dlamini, Sabelo, 2018. A comparative evaluation of the performance of popular SDN controllers. In: 2018 Wireless Days. WD, IEEE, pp. 54–59.
Manguri, Kamaran H., Omer, Saman M., 2022. SDN for IoT environment: a survey and research challenges. In: ITM Web of Conferences, Vol. 42. EDP Sciences, pp. 1–6.
Manogaran, Gunasekaran, Baabdullah, Tahani, Rawat, Danda B., Shakeel, P Mohamed, 2021. AI-assisted service virtualization and flow management framework for 6G-enabled cloud-software-defined network-based IoT. IEEE Internet Things J. 9 (16), 14644–14654.
Mao, Wenliang, Zhao, Zhiwei, Chang, Zheng, Min, Geyong, Gao, Weifeng, 2021. Energy-efficient Industrial Internet of Things: Overview and open issues. IEEE Trans. Ind. Inform. 17 (11), 7225–7237.
Masood, Mohsin, Fouad, Mohamed Mostafa, Seyedzadeh, Saleh, Glesk, Ivan, 2019. Energy efficient software defined networking algorithm for wireless sensor networks. Transp. Res. Procedia 40, 1481–1488.
Mazhar, Muhammad Shoaib, Saleem, Yasir, Almogren, Ahmad, Arshad, Jehangir, Jaffery, Mujtaba Hussain, Rehman, Ateeq Ur, Shafiq, Muhammad, Hamam, Habib, 2022. Forensic analysis on internet of things (IoT) device using machine-to-machine (M2M) framework. Electronics 11 (7), 1–23.
Mazhar, Noman, Salleh, Rosli, Zeeshan, Muhammad, Hameed, M. Muzaffar, 2021. Role of device identification and manufacturer usage description in iot security: A survey. IEEE Access 9, 41757–41786.
Medhane, Darshan Vishwasrao, Sangaiah, Arun Kumar, Hossain, M. Shamim, Muhammad, Ghulam, Wang, Jin, 2020. Blockchain-enabled distributed security framework for next-generation IoT: An edge cloud and software-defined network-integrated approach. IEEE Internet Things J. 7 (7), 6143–6149.
Meng, Weizhi, Li, Wenjuan, Zhou, Jianying, 2021. Enhancing the security of blockchain-based software defined networking through trust-based traffic fusion and filtration. Inf. Fusion 70, 60–71.
METRICS Project, [Online]. Available: http://www.av.it.pt/metrics/. (Accessed 8 December 2023).
Microsoft Azure Cloud Computing Platform & Services, [Online]. Available: https://azure.microsoft.com/en-us/. (Accessed 13 April 2024).
Mondal, Ayan, Misra, Sudip, Maity, Ilora, 2019. AMOPE: Performance analysis of OpenFlow systems in software-defined networks. IEEE Syst. J. 14 (1), 124–131.
Moreno Escobar, Jesus Jaime, Morales Matamoros, Oswaldo, Lina Reyes, Ixchel, Tejeida-Padilla, Ricardo, Chanona Hernandez, Liliana, Posadas Duran, Juan Pablo Francisco, 2020. Energy-efficient industrial internet of things software-defined network by means of the peano fractal. Sensors 20 (10), 2855.
Mostafavi, Seyedakbar, Hakami, Vesal, Sanaei, Maryam, 2021. Quality of service provisioning in network function virtualization: a survey. Computing 103 (5), 917–991.
Moutinho, Luis, Pedreiras, Paulo, Almeida, Luis, 2019. A real-time software defined networking framework for next-generation industrial networks. IEEE Access 7, 164468–164479.
Municio, Esteban, Latre, Steven, Marquez-Barja, Johann M., 2020. Extending network programmability to the things overlay using distributed industrial IoT protocols. IEEE Trans. Ind. Inform. 17 (1), 251–259.
Mwanza, Ntumpha P., Kalita, Jugal, 2023. Detecting ddos attacks in software defined networks using deep learning techniques: A survey. Int. J. Netw. Secur. 25 (2), 360–376.
Naeem, Faisal, Tariq, Muhammad, Poor, H. Vincent, 2020a. SDN-enabled energy-efficient routing optimization framework for industrial Internet of Things. IEEE Trans. Ind. Inform. 17 (8), 5660–5667.
Naeem, Faisal, Tariq, Muhammad, Poor, H. Vincent, 2020b. SDN-enabled energy-efficient routing optimization framework for industrial Internet of Things. IEEE Trans. Ind. Inform. 17 (8), 5660–5667.
Nayak, Naresh Ganesh, Dürr, Frank, Rothermel, Kurt, 2016. Time-sensitive software-defined network (TSSDN) for real-time applications. In: Proceedings of the 24th International Conference on Real-Time Networks and Systems. pp. 193–202.
Nguyen, Quang-Duy, Dhouib, Saadia, Chanet, Jean-Pierre, Bellot, Patrick, 2022. Towards a web-of-things approach for opc ua field device discovery in the industrial iot. In: 2022 IEEE 18th International Conference on Factory Communication Systems. WFCS, IEEE, pp. 1–4.
Nguyen, Dinh C., Ding, Ming, Pathirana, Pubudu N., Seneviratne, Aruna, Li, Jun, Poor, H. Vincent, 2021. Federated learning for Internet of Things: A comprehensive survey. IEEE Commun. Surv. Tutor. 23 (3), 1622–1658.
Nisar, Kashif, Jimson, Emilia Rosa, Hijazi, Mohd Hanafi Ahmad, Welch, Ian, Hassan, Rosilah, Aman, Azana Hafizah Mohd, Sodhro, Ali Hassan, Pirbhulal, Sandeep, Khan, Sohrab, 2020. A survey on the architecture, application, and security of software defined networking: Challenges and open issues. Internet Things 12, 100289.
Njah, Yosra, Cheriet, Mohamed, 2021. Parallel route optimization and service assurance in energy-efficient software-defined industrial IoT networks. IEEE Access 9, 24682–24696.

Noor-A-Rahim, Md, Firyaguna, Fadhil, John, Jobish, Khyam, M. Omar, Pesch, Dirk, Armstrong, Eddie, Claussen, Holger, Poor, H. Vincent, 2022. Toward industry 5.0: Intelligent reflecting surface in smart manufacturing. IEEE Commun. Mag. 60 (10), 72–78.

OASIS Standard, 2019. MQTT Version 5.0. pp. 1–137, Retrieved 22 June.

Ojo, Mike, Adami, Davide, Giordano, Stefano, 2016. A SDN-IoT architecture with NFV implementation. In: 2016 IEEE Globecom Workshops. GC Wkshps, IEEE, pp. 1–6.

Oktian, Yustus Eko, Lee, SangGon, Lee, HoonJae, Lam, JunHuy, 2017. Distributed SDN controller system: A survey on design choice. Comput. Netw. 121, 100–111.

Okwuibe, Jude, Haavisto, Juuso, Kovacevic, Ivana, Harjula, Erkki, Ahmad, Ijaz, Islam, Johirul, Ylianttila, Mika, 2021. Sdn-enabled resource orchestration for industrial iot in collaborative edge-cloud networks. IEEE Access 9, 115839–115854.

Okwuide, Jude, Haavisto, Juuso, Harjula, Erkki, Ahmad, Ijaz, Ylianttila, Mika, 2020. SDN enhanced resource orchestration for industrial IoT in containerized edge applications. IEEE Access 2169–3536.

Oliveira, Tadeu F., Xavier-de Souza, Samuel, Silveira, Luiz F., 2021. Improving energy efficiency on SDN control-plane using multi-core controllers. Energies 14 (11), 3161.

Open Networking Foundation, [Online]. Available: https://www.opennetworking.org/. (Accessed 8 December 2023).

Open vSwitch, [Online]. Available: https://www.openvswitch.org/. (Accessed 13 April 2024).

Orozco-Santos, Federico, Sempere-Payá, Víctor, Silvestre-Blanes, Javier, Albero-Albero, Teresa, 2021. Multicast scheduling in sdn wise to support mobile nodes in industrial wireless sensor networks. IEEE Access 9, 141651–141666.

Ouhab, Abdallah, Abreu, Thiago, Slimani, Hachem, Mellouk, Abdelhamid, 2020. Energy-efficient clustering and routing algorithm for large-scale SDN-based IoT monitoring. In: ICC 2020-2020 IEEE International Conference on Communications. ICC, IEEE, pp. 1–6.

Owusu, Ampratwum Isaac, Nayak, Amiya, 2020. An intelligent traffic classification in sdn-iot: A machine learning approach. In: 2020 IEEE International Black Sea Conference on Communications and Networking. BlackSeaCom, IEEE, pp. 1–6.

Padrah, Zoltan, Pastrav, Andra, Palade, Tudor, Ratiu, Ovidiu, Puschita, Emanuel, 2021. Development and validation of an ISA100. 11a simulation model for accurate industrial WSN planning and deployment. Sensors 21 (11), 1–29.

Paganelli, Federica, Cappanera, Paola, Cuffaro, Giovanni, 2021. Tenant-defined service function chaining in a multi-site network slice. Future Gener. Comput. Syst. 121, 1–18.

Paliwal, Manish, Shrimankar, Deepti, 2019. Effective resource management in SDN enabled data center network based on traffic demand. IEEE Access 7, 69698–69706.

Pang, Zaiyu, Huang, Xiao, Li, Zonghui, Zhang, Sukun, Xu, Yanfen, Wan, Hai, Zhao, Xibin, 2020. Flow scheduling for conflict-free network updates in time-sensitive software-defined networks. IEEE Trans. Ind. Inform. 17 (3), 1668–1678.

Park, Jun-Hong, Kim, Hyeong-Su, Kim, Won-Tae, 2018. Dm-mqtt: An efficient mqtt based on sdn multicast for massive iot communications. Sensors 18 (9), 3071.

Pfaff, Ben, Davie, Bruce, 2013. The open vswitch database management protocol. pp. 1–35, Internet Eng. Task Force (IETF), No. RFC 7047, Technical report.

Pfaff, Ben, Lantz, Bob, Heller, Brandon, Barker, C., Beckmann, C., Cohn, D., Talayco, D., Erickson, D., McDysan, D., Ward, D., et al., 2012. Openflow Switch Specification, version 1.3.1. Open Netw. Found., Menlo Park, CA, USA, pp. 1–128.

Pivoto, Diego G.S., Rezende, Tibério T., Facina, Michelle S.P., Moreira, Rodrigo, de Oliveira Silva, Flávio, Cardoso, Kleber V., Correa, Sand L., Araujo, Antonia V.D., Silva, Rogério S., Neto, Heitor Scalco, et al., 2023. A detailed relevance analysis of enabling technologies for 6G architectures. IEEE Access 11, 89644–89684.

Pokhrel, Shiva Raj, Verma, Sandeep, Garg, Sahil, Sharma, Ajay K, Choi, Jinho, 2020. An efficient clustering framework for massive sensor networking in industrial internet of things. IEEE Trans. Ind. Inform. 17 (7), 4917–4924.

Priya, A. Vishnu, Radhika, N., 2019. Performance comparison of SDN OpenFlow controllers. Int. J. Comput. Aided Eng. Technol. 11 (4–5), 467–479.

Pu, Chenggen, Ding, Xiwu, Wang, Ping, Xie, Shunji, Chen, Junhua, 2022. Semantic interconnection scheme for industrial wireless sensor networks and industrial internet with OPC UA pub/sub. Sensors 22 (20), 7762.

Qiu, Tie, Chi, Jiancheng, Zhou, Xiaobo, Ning, Zhaolong, Atiquzzaman, Mohammed, Wu, Dapeng Oliver, 2020. Edge computing in industrial internet of things: Architecture, advances and challenges. IEEE Commun. Surv. Tutor. 22 (4), 2462–2488.

Qiu, Chao, Yu, F. Richard, Yao, Haipeng, Jiang, Chunxiao, Xu, Fangmin, Zhao, Chenglin, 2018. Blockchain-based software-defined industrial Internet of Things: A dueling deep q-learning approach. IEEE Internet Things J. 6 (3), 4627–4639.

Rafique, Wajid, Qi, Lianyong, Yaqoob, Ibrar, Imran, Muhammad, Rasool, Raihan Ur, Dou, Wanchun, 2020. Complementing IoT services through software defined networking and edge computing: A comprehensive survey. IEEE Commun. Surv. Tutor. 22 (3), 1761–1804.

Rahimi, Payam, Chrysostomou, Chrysostomos, Pervaiz, Haris, Vassiliou, Vasos, Ni, Qiang, 2021. Joint radio resource allocation and beamforming optimization for Industrial Internet of Things in software-defined networking-based virtual fog-radio access network 5G-and-beyond wireless environments. IEEE Trans. Ind. Inform. 18 (6), 4198–4209.

Rahman, Anichur, Sara, Umme, Kundu, Dipanjali, Islam, Saiful, Islam, Md Jahidul, Hasan, Mahedi, Rahman, Ziaur, Nasir, M., 2020. DistB-SDoindustry: Enhancing security in industry 4.0 services based on distributed blockchain through sdn-iot enabled architecture. Int. J. Adv. Comput. Sci. Appl. 11 (9), 674–681.

Rahouti, Mohamed, Xiong, Kaiqi, Xin, Yufeng, 2020. Secure software-defined networking communication systems for smart cities: Current status, challenges, and trends. IEEE Access 9, 12083–12113.

Ramakrishnan, Jayabrabu, Shabbir, Muhammad Salman, Kassim, Normalini Md, Nguyen, Phong Thanh, Mavaluru, Dinesh, 2020. A comprehensive and systematic review of the network virtualization techniques in the IoT. Int. J. Commun. Syst. 33 (7), e4331.

Ranganathan, Mudumbai, Montgomery, Douglas, El Mimouni, Omar Ilias, 2019. Soft MUD: Implementing manufacturer usage descriptions on OpenFlow SDN switches. In: Proceedings of the International Conference on Networks. ICN, pp. 1–6.

Rehman, A.U., Aguiar, Rui L., Barraca, Joao Paulo, 2019. Fault-tolerance in the scope of software-defined networking (sdn). IEEE Access 7, 124474–124490.

Ren, Yuzheng, Xie, Renchao, Yu, F. Richard, Huang, Tao, Liu, Yunjie, 2020. Potential identity resolution systems for the industrial Internet of Things: A survey. IEEE Commun. Surv. Tutor. 23 (1), 391–430.

Ribeiro, Paulo A., Duoba, Liudas, Prior, Rui, Crisostomo, Sergio, Almeida, Luis, 2019. Real-time wireless data plane for real-time-enabled SDN. In: 2019 15th IEEE International Workshop on Factory Communication Systems. WFCS, IEEE, pp. 1–4.

Rinaldi, Stefano, Bonafini, Federico, Ferrari, Paolo, Flammini, Alessandra, Sisinni, Emiliano, Di Cara, Dario, Panzavecchia, Nicola, Tinè, Giovanni, Cataliotti, Antonio, Cosentino, Valentina, et al., 2018. Characterization of IP-based communication for smart grid using software-defined networking. IEEE Trans. Instrum. Meas. 67 (10), 2410–2419.

Rodriguez-Natal, Alberto, Portoles-Comeras, Marc, Ermagan, Vina, Lewis, Darrel, Farinacci, Dino, Maino, Fabio, Cabellos-Aparicio, Albert, 2015. LISP: a southbound SDN protocol? IEEE Commun. Mag. 53 (7), 201–207.

Romero-Gázquez, José L., Bueno-Delgado, M., 2018. Software architecture solution based on SDN for an industrial IoT scenario. Wirel. Commun. Mob. Comput. 2018, 1–14.

Rosli, Ahmad Nazhan, Mohamad, Roslina, Yusof, Yuslinda Wati Mohamad, Shahbudin, Shahrani, Rahman, Farah Yasmin Abdul, 2020. Implementation of mqtt and lorawan system for real-time environmental monitoring application. In: 2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics. ISCAIE, IEEE, pp. 287–291.

Saha, Niloy, Bera, Samaresh, Misra, Sudip, 2018. Sway: Traffic-aware QoS routing in software-defined IoT. IEEE Trans. Emerg. Top. Comput. 9 (1), 390–401.

Sahoo, Kshira Sagar, Tiwary, Mayank, Luhach, Ashish Kr, Nayyar, Anand, Choo, Kim-Kwang Raymond, Bilal, Muhammad, 2021. Demand–supply-based economic model for resource provisioning in industrial IoT traffic. IEEE Internet Things J. 9 (13), 10529–10538.

Said, Siwar Ben Hadj, Truong, Quang Huy, Boc, Michael, 2019. SDN-based configuration solution for IEEE 802.1 time sensitive networking (TSN). ACM SIGBED Rev. 16 (1), 27–32.

Salama, Mahmoud, Elkaseer, Ahmed, Saied, Mohamed, Ali, Hazem, Scholz, Steffen, 2019. Industrial Internet of Things solution for real-time monitoring of the additive manufacturing process. In: Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology–ISAT 2018: Part I. Springer, pp. 355–365.

Saleh, Sherine Nagy, Fathy, Cherine, 2023. A novel deep-learning model for remote driver monitoring in SDN-based internet of autonomous vehicles using 5G technologies. Appl. Sci. 13 (2), 875.

Salih, Kazhan Othman Mohammed, Rashid, Tarik A., Radovanovic, Dalibor, Bacanin, Nebojsa, 2022. A comprehensive survey on the Internet of Things with the industrial marketplace. Sensors 22 (3), 730.

Sarkar, Joy Lal, Cowlessur, Sanjeev K., Ramasamy, V., Pati, Bibudhendu, Selvi, T. Mahesh, Panigrahi, Chhabi Rani, Majumder, Bibek, Verma, Rajesh Kumar, Qureshi, Nawab Muhammad Fasech, 2023. FogCom: SDN-enabled fog node selection for early detection of communicable diseases. J. King Saud Univ.-Comput. Inf. Sci. 35 (6), 101432.

Satka, Zenepe, Pantzar, David, Magnusson, Alexander, Ashjaei, Mohammad, Fotouhi, Hossein, Sjödin, Mikael, Daneshtalab, Masoud, Mubeen, Saad, 2022. Developing a translation technique for converged TSN-5G communication. In: 2022 IEEE 18th International Conference on Factory Communication Systems. WFCS, IEEE, pp. 1–8.

Savaliya, Abhishek, Jhaveri, Rutvij H., Xin, Qin, Alqithami, Saad, Ramani, Sagar, Ahanger, Tariq Ahamed, 2021. Securing industrial communication with software-defined networking. Math. Biosci. Eng. 18 (6), 8298–8314.

Schlinker, B., 2014. MiniNEXT: MiniNet extension. [Online]. Available: https://github.com/USC-NSL/miniNExT. (Accessed 8 December 2023).

Schulz, Philipp, Matthe, Maximilian, Klessig, Henrik, Simsek, Meryem, Fettweis, Gerhard, Ansari, Junaid, Ashraf, Shehzad Ali, Almeroth, Bjoern, Voigt, Jens, Riedel, Ines, et al., 2017. Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture. IEEE Commun. Mag. 55 (2), 70–78.

Serror, Martin, Hack, Sacha, Henze, Martin, Schuba, Marko, Wehrle, Klaus, 2020. Challenges and opportunities in securing the industrial internet of things. IEEE Trans. Ind. Inform. 17 (5), 2985–2996.

Setiawan, Dharma Yusuf, Hertiana, Sofia Naning, Negara, Ridha Muldina, 2021. 6LoWPAN performance analysis of IoT software-defined-network-based using mininet-io. In: 2020 IEEE International Conference on Internet of Things and Intelligence System. IoTaIS, IEEE, pp. 60–65.

Shah, Syed Danial Ali, Gregory, Mark A., Li, Shuo, 2021. Cloud-native network slicing using software defined networking based multi-access edge computing: A survey. IEEE Access 9, 10903–10924.

Shahri, Ehsan, Pedreiras, Paulo, Almeida, Luis, 2022. Extending mqtt with real-time communication services based on sdn. Sensors 22 (9), 3162.

Sharma, Pradip Kumar, Chen, Mu-Yen, Park, Jong Hyuk, 2017. A software defined fog node based distributed blockchain cloud architecture for IoT. IEEE Access 6, 115–124.

Shu, Lei, Mukherjee, Mithun, Pecht, Michael, Crespi, Noël, Han, Son N., 2017. Challenges and research issues of data management in IoT for large-scale petrochemical plants. IEEE Syst. J. 12 (3), 2509–2523.

Silva, Luis, Gonçalves, Pedro, Marau, Ricardo, Pedreiras, Paulo, 2017a. Extending OpenFlow with industrial grade communication services. In: 2017 IEEE 13th International Workshop on Factory Communication Systems. WFCS, IEEE, pp. 1–4.

Silva, Luis, Gonçalves, Pedro, Marau, Ricardo, Pedreiras, Paulo, Almeida, Luis, 2017b. Extending OpenFlow with flexible time-triggered real-time communication services. In: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation. ETFA, IEEE, pp. 1–8.

Silva, Luis, Pedreiras, Paulo, Fonseca, Pedro, Almeida, Luis, 2019. On the adequacy of SDN and TSN for Industry 4.0. In: 2019 IEEE 22nd International Symposium on Real-Time Distributed Computing. ISORC, IEEE, pp. 43–51.

Singh, Maninderpal, Aujla, Gagangeet Singh, Singh, Amritpal, Kumar, Neeraj, Garg, Sahil, 2020. Deep-learning-based blockchain framework for secure software-defined industrial networks. IEEE Trans. Ind. Inform. 17 (1), 606–616.

Singh Rajawat, Anand, Bedi, Pradeep, Goyal, S.B., Shukla, Piyush Kumar, Zaguia, Atef, Jain, Aakriti, Monirujjaman Khan, Mohammad, 2021. Reformist framework for improving human security for mobile robots in industry 4.0. Mob. Inf. Syst. 2021, 1–10.

Smith, M., Dvorkin, M., Laribi, Y., Pandey, V., Garg, P., Weidenbacher, N., 2014. OpFlex control protocol. pp. 1–24, Internet Eng. Task Force (IETF), Internet draft. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-smith-opflex-00. (Accessed 30 December 2023).

Song, Chenyang, Wu, Zhipeng, Gray, John, Meng, Zhaozong, 2023. An RFID-powered multi-sensing fusion industrial IoT system for food quality assessment and sensing. IEEE Trans. Ind. Inform. 20 (1), 337–348.

Srirama, Satish Narayana, 2024. A decade of research in fog computing: Relevance, challenges, and future directions. Softw. - Pract. Exp. 54 (1), 3–23.

Struhár, Václav, Ashjaei, Mohammad, Behnam, Moris, Craciunas, Silviu S, Papadopoulos, Alessandro V, 2019. Dart: Dynamic bandwidth distribution framework for virtualized software defined networks. In: IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society, Vol. 1. IEEE, pp. 2934–2939.

Sudhakaran, Susruth, Montgomery, Karl, Kashef, Mohamed, Cavalcanti, Dave, Candell, Richard, 2022. Wireless time sensitive networking impact on an industrial collaborative robotic workcell. IEEE Trans. Ind. Inform. 18 (10), 7351–7360.

Sun, Weifeng, Wang, Zun, Zhang, Guanghao, 2021. A qos-guaranteed intelligent routing mechanism in software-defined networks. Comput. Netw. 185, 107709.

Sylla, Tidiane, Mendiboure, Leo, Berbineau, Marion, Singh, Radheshyam, Soler, José, Berger, Michael Stübert, 2022. Emu5GNet: An open-source emulator for 5G software-defined networks. In: 2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications. WiMob, IEEE, pp. 474–477.

Tabaa, Mohamed, Monteiro, Fabrice, Bensag, Hassna, Dandache, Abbas, 2020. Green Industrial Internet of Things from a smart industry perspectives. Energy Rep. 6, 430–446.

Tadros, Catherine Nayer, Rizk, Mohamed R.M., Mokhtar, Bassem Mahmoud, 2020. Software defined network-based management for enhanced 5G network services. IEEE Access 8, 53997–54008.

Taleb, Tarik, Samdanis, Konstantinos, Mada, Badr, Flinck, Hannu, Dutta, Sunny, Sabella, Dario, 2017. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Commun. Surv. Tutor. 19 (3), 1657–1681.

Tan, Yingping, Peng, Ge, Wang, Hao, Wei, Min, 2022. Research on secure communication of industrial wireless WIA-pa networks. In: 2nd International Conference on Internet of Things and Smart City, Vol. 12249. 12249, SPIE, pp. 86–91.

Tange, Koen, De Donno, Michele, Fafoutis, Xenofon, Dragoni, Nicola, 2020. A systematic survey of industrial Internet of Things security: Requirements and fog computing opportunities. IEEE Commun. Surv. Tutor. 22 (4), 2489–2520.

Theodorou, Tryfon, Mamatas, Lefteris, 2020. A versatile out-of-band software-defined networking solution for the Internet of Things. IEEE Access 8, 103710–103733.

Thubert, P., 2019. An architecture for IPv6 over the TSCH mode of IEEE 802.15.4. pp. 1–74, Internet Eng. Task Force (IETF), draft-ietf-6tisch-architecture-23, RFC 9030.

Thubert, Pascal, Palattella, Maria Rita, Engel, Thomas, 2015. 6TiSCH centralized scheduling: When SDN meet IoT. In: 2015 IEEE Conference on Standards for Communications and Networking. CSCN, IEEE, pp. 42–47.

Tuysuz, Mehmet Fatih, Ankarali, Zekiye Kubra, Gözüpek, Didem, 2017. A survey on energy efficiency in software defined networks. Comput. Netw. 113, 188–204.

Urrea, Claudio, Benítez, David, 2021. Software-defined networking solutions, architecture and controllers for the industrial Internet of Things: A review. Sensors 21 (19), 65–85.

Vadi, Seyfettin, Bayindir, Ramazan, Toplar, Yigit, Colak, Ilhami, 2022. Induction motor control system with a Programmable Logic Controller (PLC) and profibus communication for industrial plants—An experimental setup. ISA Trans. 122, 459–471.

Varis, Pekka, Leyrer, Thomas, 2018. Time-sensitive networking for industrial automation. Texas Instruments, Texas, SPRY316. [Online]. Available: https://www.ti.com/lit/wp/spry316b/spry316b.pdf. (Accessed 8 December 2023).

Vestin, Jonathan, Kassler, Andreas, Akerberg, Johan, 2015. Resilient software defined networking for industrial control networks. In: 2015 10th International Conference on Information, Communications and Signal Processing. ICICS, IEEE, pp. 1–5.

Vestin, Jonathan, Kassler, Andreas, Åkerberg, Johan, 2018. FastReact: In-network control and caching for industrial control networks using programmable data planes. In: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation, Vol. 1. ETFA, IEEE, pp. 219–226.

Vilajosana, Xavier, Watteyne, Thomas, Chang, Tengfei, Vučinić, Mališa, Duquennoy, Simon, Thubert, Pascal, 2019. Ietf 6tisch: A tutorial. IEEE Commun. Surv. Tutor. 22 (1), 595–615.

Vlk, Marek, Brejchová, Kateřina, Hanzálek, Zdeněk, Tang, Siyu, 2022. Large-scale periodic scheduling in time-sensitive networks. Comput. Oper. Res. 137, 105512.

Wan, Jiafu, Tang, Shenglong, Shu, Zhaogang, Li, Di, Wang, Shiyong, Imran, Muhammad, Vasilakos, Athanasios V., 2016. Software-defined industrial internet of things in the context of industry 4.0. IEEE Sens. J. 16 (20), 7373–7380.

Wang, Shie-Yuan, 2014. Comparison of SDN OpenFlow network simulator and emulators: EstiNet vs. Mininet. In: 2014 IEEE Symposium on Computers and Communications. ISCC, IEEE, pp. 1–6.

Wang, Rongkai, Gu, Chaojie, He, Shibo, Shi, Zhiguo, Meng, Wenchao, 2022a. An interoperable and flat Industrial Internet of Things architecture for low latency data collection in manufacturing systems. J. Syst. Archit. 129, 1–13.

Wang, Juan, Li, Di, 2018. Adaptive computing optimization in software-defined network-based industrial internet of things with fog computing. Sensors 18 (8), 1–14.

Wang, Jiadai, Liu, Jiajia, 2021. Deep learning for securing software-defined industrial Internet of Things: attacks and countermeasures. IEEE Internet Things J. 9 (13), 11179–11189.

Wang, Shupeng, Nie, Laisen, Li, Guojun, Wu, Yixuan, Ning, Zhaolong, 2022b. A multitask learning-based network traffic prediction approach for SDN-enabled industrial Internet of Things. IEEE Trans. Ind. Inform. 18 (11), 7475–7483.

Wang, Yi, Pu, Chenggen, Wang, Ping, Wu, Junrui, 2020. A CoAP-based OPC UA transmission scheme for resource-constrained devices. In: 2020 Chinese Automation Congress. CAC, IEEE, pp. 6089–6093.

Wang, Haopei, Srivastava, Abhinav, Xu, Lei, Hong, Sungmin, Gu, Guofei, 2017. Bring your own controller: Enabling tenant-defined SDN apps in IaaS clouds. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications. IEEE, pp. 1–9.

Wang, Ping, Wang, Heng, Zhang, Chang, 2019a. SDN-based WIA-PA field network/IPV6 backhaul network joint scheduling method. US Patent 10, 306, 706.

Wang, Heng, Zeng, Leipei, Li, Min, Yang, Chuang, 2019b. A protocol conversion scheme between WIA-PA networks and time-sensitive networks. In: 2019 Chinese Automation Congress. CAC, IEEE, pp. 213–218.

Wei, Min, Li, Cheng, Li, Caiqin, 2020. An IPv6 internet accessing architecture and approach for industrial wireless network. In: 2020 14th International Conference on Ubiquitous Information Management and Communication. IMCOM, IEEE, pp. 1–6.

Wei, Min, Xiang, Xueqin, Li, Cheng, 2021. A traffic scheduling mechanism for industrial wireless network accessing IPv6 internet. In: 2021 International Conference on Information Networking. ICOIN, IEEE, pp. 764–769.

Wijethilaka, Shalitha, Liyanage, Madhusanka, 2021. Survey on network slicing for internet of things realization in 5G networks. IEEE Commun. Surv. Tutor. 23 (2), 957–994.

Wójcicki, Krzysztof, Biegańska, Marta, Paliwoda, Beata, Górna, Justyna, 2022. Internet of Things in industry: Research profiling, application, challenges and opportunities—A review. Energies 15 (5), 1806.

Wu, Yulei, 2020. Cloud-edge orchestration for the Internet of Things: Architecture and AI-powered data processing. IEEE Internet Things J. 8 (16), 12792–12805.

Wu, Yulei, Dai, Hong-Ning, Wang, Haozhe, Xiong, Zehui, Guo, Song, 2022. A survey of intelligent network slicing management for industrial IoT: Integrated approaches for smart transportation, smart energy, and smart factory. IEEE Commun. Surv. Tutor. 24 (2), 1175–1211.

Wu, Xiping, Soltani, Mohammad Dehghani, Zhou, Lai, Safari, Majid, Haas, Harald, 2021. Hybrid LiFi and WiFi networks: A survey. IEEE Commun. Surv. Tutor. 23 (2), 1398–1420.

Xia, Wenchao, Zhang, Jun, Quek, Tony Q.S., Jin, Shi, Zhu, Hongbo, 2020. Mobile edge cloud-based industrial Internet of Things: Improving edge intelligence with hierarchical SDN controllers. IEEE Veh. Technol. Mag. 15 (1), 36–45.

Xu, Nan, Fan, Xingyu, Hu, Rui, 2022. Adoption of green industrial internet of things to improve organizational performance: The role of institutional isomorphism and green innovation practices. Front. Psychol. 13, 1–9.

Yan, Qiao, Huang, Wenyao, Luo, Xupeng, Gong, Qingxiang, Yu, F. Richard, 2018. A multi-level DDoS mitigation framework for the industrial Internet of Things. IEEE Commun. Mag. 56 (2), 30–36.
Yan, Jinli, Jia, Chunbo, Tang, Lu, Li, Tao, Lv, Gaofeng, Quan, Wei, Yang, Hui, 2020. Network programming interface in general-purpose multi-core processor: A survey. In: 2020 15th International Conference on Computer Science & Education. ICCSE, IEEE, pp. 675–680.
Yan, Binghao, Liu, Qinrang, Shen, JianLiang, Liang, Dong, Zhao, Bo, Ouyang, Ling, 2021. A survey of low-latency transmission strategies in software defined networking. Comp. Sci. Rev. 40, 100386.
Yan, Wenhao, Wang, Jing, Lu, Shan, Zhou, Meng, Peng, Xin, 2023. A review of real-time fault diagnosis methods for industrial smart manufacturing. Processes 11 (2), 369.
Yang, Chao-Tung, Liu, Jung-Chun, Chen, Wei-Sheng, Leu, Fang-Yie, Chu, William Cheng-Chung, 2017. Implementation of a virtual switch monitoring system using OpenFlow on cloud. Int. J. Ad Hoc Ubiquitous Comput. 24 (3), 162–172.
Yang, Gyeongsik, Yu, Bong-yeol, Jin, Heesang, Yoo, Chuck, 2020. Libera for programmable network virtualization. IEEE Commun. Mag. 58 (4), 38–44.
Yin, H., Xie, H., Tsou, T., Lopez, D., Aranda, P., Sidi, R., 2012. Sdni: A message exchange protocol for software defined networks (sdns) across multiple domain. pp. 1–16, Internet Research Task Force, Internet-Draft, Tech. Report. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-yin-sdn-sdni-00. (Accessed 30 December 2023).
Younan, Mina, Houssein, Essam H., Elhoseny, Mohamed, Ali, Abdelmgeid A., 2020. Challenges and recommended technologies for the industrial Internet of Things: A comprehensive review. Measurement 151, 107198.
Younus, Muhammad Usman, Khan, Muhammad Khurram, Bhatti, Abdul Rauf, 2021. Improving the software-defined wireless sensor networks routing performance using reinforcement learning. IEEE Internet Things J. 9 (5), 3495–3508.
Yu, Wei, Liang, Fan, He, Xiaofei, Hatcher, William Grant, Lu, Chao, Lin, Jie, Yang, Xinyu, 2017. A survey on the edge computing for the internet of things. IEEE Access 6, 6900–6919.
Yu, Keping, Tan, Liang, Aloqaily, Moayad, Yang, Hekun, Jararweh, Yaser, 2021. Blockchain-enhanced data sharing with traceable and direct revocation in IIoT. IEEE Trans. Ind. Inform. 17 (11), 7669–7678.
Yungaicela-Naula, Noe M., Vargas-Rosales, Cesar, Pérez-Díaz, Jesús Arturo, Zareei, Mahdi, 2022. Towards security automation in software defined networks. Comput. Commun. 183, 64–82.
Zahoor, Sumbal, Ahmad, Ishtiaq, Othman, Mohamed Tahar Ben, Mamoon, Ali, Rehman, Ateeq Ur, Shafiq, Muhammad, Hamam, Habib, 2022. Comprehensive analysis of network slicing for the developing commercial needs and networking challenges. Sensors 22 (17), 6623.
Zainudin, Ahmad, Ahakonye, Love Allen Chijioke, Akter, Rubina, Kim, Dong-Seong, Lee, Jae-Min, 2022. An efficient hybrid-dnn for ddos detection and classification in software-defined iiot networks. IEEE Internet Things J. 10 (10), 8491–8504.
Zanella, Andrea, Bui, Nicola, Castellani, Angelo, Vangelista, Lorenzo, Zorzi, Michele, 2014. Internet of things for smart cities. IEEE Internet Things J. 1 (1), 22–32.
Zeng, Tao, Wang, Shibing, Liu, Shuying, 2020. Research on intelligent linkage server switch in case of power loss in computer room. In: 2020 IEEE 11th International Conference on Software Engineering and Service Science. ICSESS, IEEE, pp. 493–496.
Zhang, Ziyao, Ma, Liang, Leung, Kin K., Le, Franck, 2021. More is not always better: An analytical study of controller synchronizations in distributed SDN. IEEE/ACM Trans. Netw. 29 (4), 1580–1590.
Zhang, Jing, Yahya, Rebaz Othman, 2023. DRL-based routing algorithm with guaranteed loss, latency and bandwidth in SDN networks: Application of online video conferencing. J. King Saud Univ.-Comput. Inf. Sci. 35 (10), 101805.
Zhao, Bohan, Zhao, Jin, Wang, Xin, Wolf, Tilman, 2019. Ruletailor: Optimizing flow table updates in openflow switches with rule transformations. IEEE Trans. Netw. Serv. Manag. 16 (4), 1581–1594.
Zhou, Qinbin, Zhao, Taotao, Chen, Xiaomin, Zhong, Yuesheng, Luo, Heng, 2022. A fault-tolerant transmission scheme in SDN-based industrial IoT (IIoT) over fiber-wireless networks. Entropy 24 (2), 1–15.
Zurawski, Richard, 2014. Industrial Communication Technology Handbook. Boca Raton, FL, USA CRC Press, pp. 1–1756.

Nteziriza Nkerabahizi Josbert received the M.S. degree in Computer Science and Technology from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2016, and his Ph.D. degree in Computer Science and Technology from Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 2021. He is currently pursuing Postdoctoral research with the School of Automation, CQUPT, Chongqing, China. His research interests include SDN, IIoT, fault management, and optimization.

Min Wei received the B.S. in Automation from Zhejiang University, Hangzhou, China, in 2005 and the M.S. degree in Control Science and Engineering from CQUPT, Chongqing, China, in 2008. He received the Ph.D. degree in Computer Science and Information Communication Engineering from Konkuk University, Seoul, Korea, in 2014. He is currently a Vice-Dean and Professor at the School of Automation, CQUPT, Chongqing, China. His main research interests include SDN, IIoT, IWSN, and Network Security.

Wang Ping received the B.S. degree in coal mining, the M.S. degree in engineering machinery, and the Ph.D. degree in bridge and tunnel engineering from Chongqing University, Chongqing, China, in 1983, 1988, and 1994, respectively. He is currently the Dean of the School of Automation, CQUPT, Chongqing. He is also the Director of the Key Laboratory of IIoT and Networked Control Ministry of Education. His main research interests include SDN, IoT, IIoT, IWSN, and intelligent detection technology and instrumentation.

Ahsan Rafiq received the M.S. degree in Computer Science from the National College of Business Administration and Economics, Lahore, Pakistan, in 2016. Ph.D. degree in Computer Science and Technology from CQUPT, Chongqing, China, in 2021. He is currently pursuing the Postdoctoral research with the School of Automation, CQUPT, Chongqing, China. His research interests include SDN, 6TiSCH networks, and edge and fog computing in IIoT.