

Article

Not peer-reviewed version

Enhancing Software Defined Networking Performance through Artificial Intelligence: A Comprehensive Analysis

[Qutaiba I. Ali](#) *

Posted Date: 16 April 2025

doi: 10.20944/preprints202504.1256.v1

Keywords: software-defined networking (SDN); artificial intelligence (AI); machine learning (ML), OpenFlow Protocol; Ryu Controller



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Enhancing Software Defined Networking Performance through Artificial Intelligence: A Comprehensive Analysis

Qutaiba I. Ali

University of Mosul/Iraq; qutaibaali@uomosul.edu.iq

Abstract: Software-Defined Networking (SDN) introduces a paradigm shift in network management by decoupling the control and data planes, thereby enabling centralized, programmable network control. However, the dynamic and complex nature of modern traffic demands adaptive and intelligent decision-making beyond traditional rule-based systems. This paper explores the integration of Artificial Intelligence (AI) techniques—particularly supervised learning algorithms—into the SDN control architecture to improve performance, efficiency, and automation. The study provides an overview of SDN architecture and the OpenFlow protocol, followed by an empirical evaluation using real traffic scenarios. Multiple AI models including Support Vector Machine (SVM), Naïve Bayes (NB), and Nearest Centroid were tested on a software-defined testbed. Performance metrics such as classification accuracy, throughput, latency, packet loss, and controller decision time were analyzed. Results demonstrate that AI integration leads to significant improvements across all metrics, validating the potential of AI-SDN synergy in creating intelligent and self-optimizing networks.

Keywords: software-defined networking (SDN); artificial intelligence (AI); machine learning (ML); OpenFlow protocol; Ryu controller

1. Introduction

In the last years, many researches are presented many solutions of complex tasks of modern network application like traffic engineering, network optimization and orchestration [1]. One of the most famous solutions is proposed To solve these problems is Software-defined networking (SDN) which maintains a global view of network

states and provides a flow-level control of the underlying layers [2]. This idea caused a dramatically change in the way how networks are designed and managed .The SDN paradigm separates the control plane from the data plane, which results in achieving much faster and dynamic approach in compared with a conventional network [3].

Recently, the artificial intelligence algorithms have started to play a significant role in most of modern systems such as intelligent transportation, which gives us the chance to improve the performance of the current computer networks. The integration between the concept in SDN paradigm and AI techniques can lead to more adaptive behavior of network elements [4]. In this work, however, we provide basic relationships between the significant role of AI in SDN paradigm as:

1. **To provide a detailed architectural understanding** of Software-Defined Networking and its foundational protocols, especially OpenFlow.
2. **To explore the role of Artificial Intelligence in enhancing SDN functionality**, including its ability to enable adaptive routing and decision-making.
3. **To evaluate the performance impact** of integrating AI models (e.g., SVM, Naïve Bayes, Nearest Centroid) into SDN controllers.

4. To compare traditional SDN with AI-augmented SDN systems across critical network metrics such as accuracy, delay, throughput, and loss rate.

2. Architectural Design of SDN

Open Networking Foundation (ONF) describes a high level architecture of SDN, which functionally and vertically split into three layers.

1-Infrastructure layer: This layer consists of forwarding devices like the physical switch, router, etc. Software switches which can be accessible via open interfaces, also part of this layer. This layer is considered as forwarding layer since it allows packet switching and forwarding.

2-Control Layer: The control layer is also referred as control plane that comprises a set of software-enabled SDN controllers. This layer allows the network administrator to apply custom policies to the physical layer devices.

3-Application layer: Application layer deals with end user business applications that utilize the SDN services. Business application such as energy efficient networking, security monitoring, network virtualization etc. as shown in Figure 1 [5].

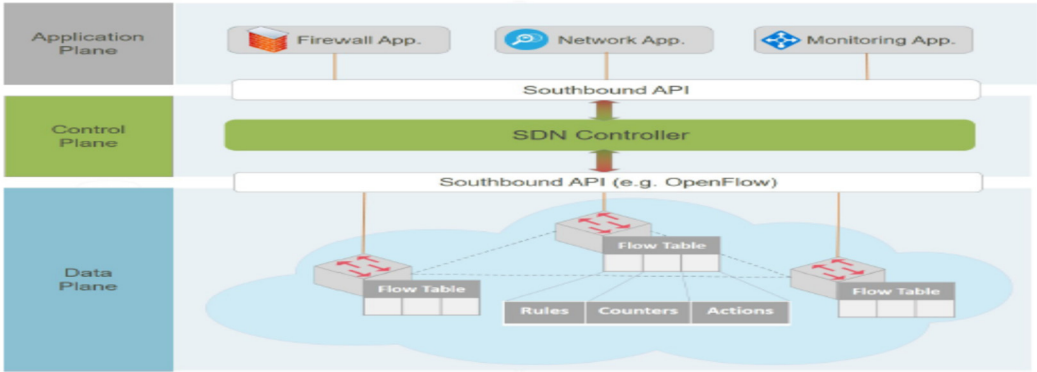


Fig. 1: A basic SDN architecture

Figure 1. The basic SDN architecture.

2.1. Openflow Protocol

OpenFlow (OF) is being continuously evolved and standardized by open networking foundation (ONF). OF supplys an abstraction layer that enables the SDN controller to securely communicate with OF-enabled forwarding elements [1]. OpenFlow has has turn out to be the de-facto standard for south bound APIs used in SDNs . A typical OF-enabled switch handles new coming packets based on its flow table. Figure 2 shows the areas of matching rules part in OF version 1.0.0.The incoming packets are described as 3 fields which are [5]:

- 1-Match fields: to match against packets depending on Figure 3 [6].
- 2-Counters: to update for matching packet.
- 3-Actions: to apply to matching packets.

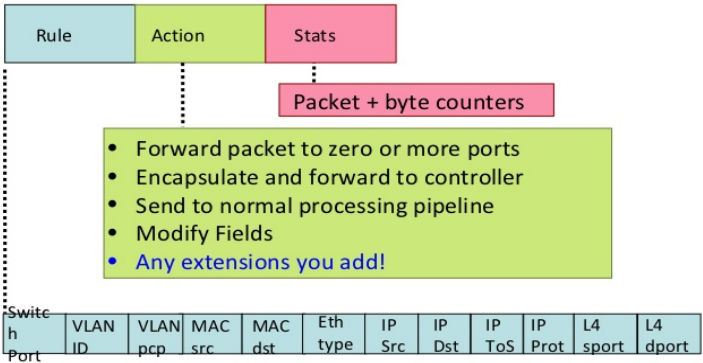


Figure 2. OpenFlow entries.

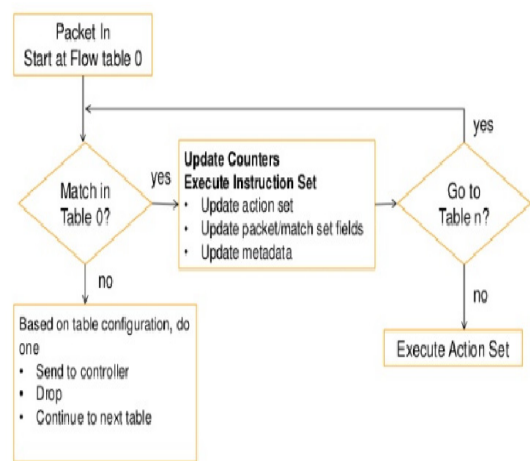


Figure 3. Packet flow in SDN switch.

In order to understand the SDN architecture, it is important to recall its basic operation. Figure 4 [4] presents the working procedures of the OpenFlow-based SDN network. Each OpenFlow switch has a flow table and uses the OpenFlow protocol to communicate with the SDN controller [3] as shown in Figure 5.

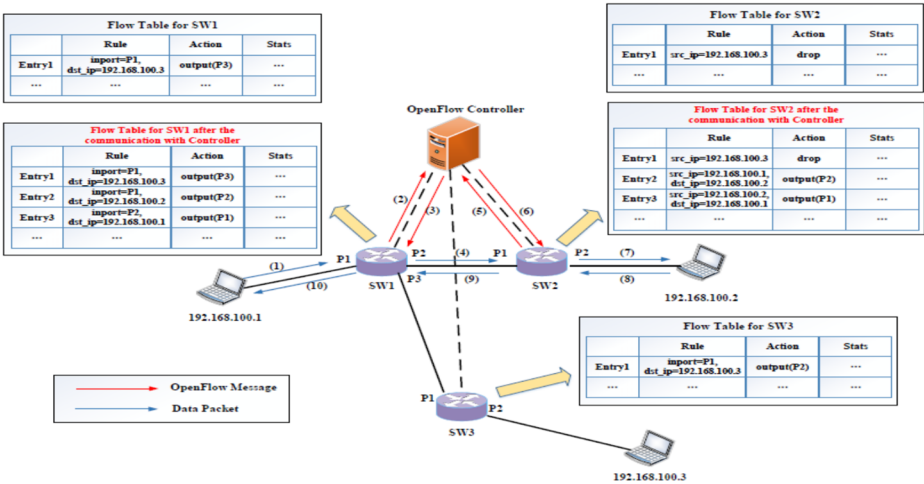


Figure 4. SDN network.

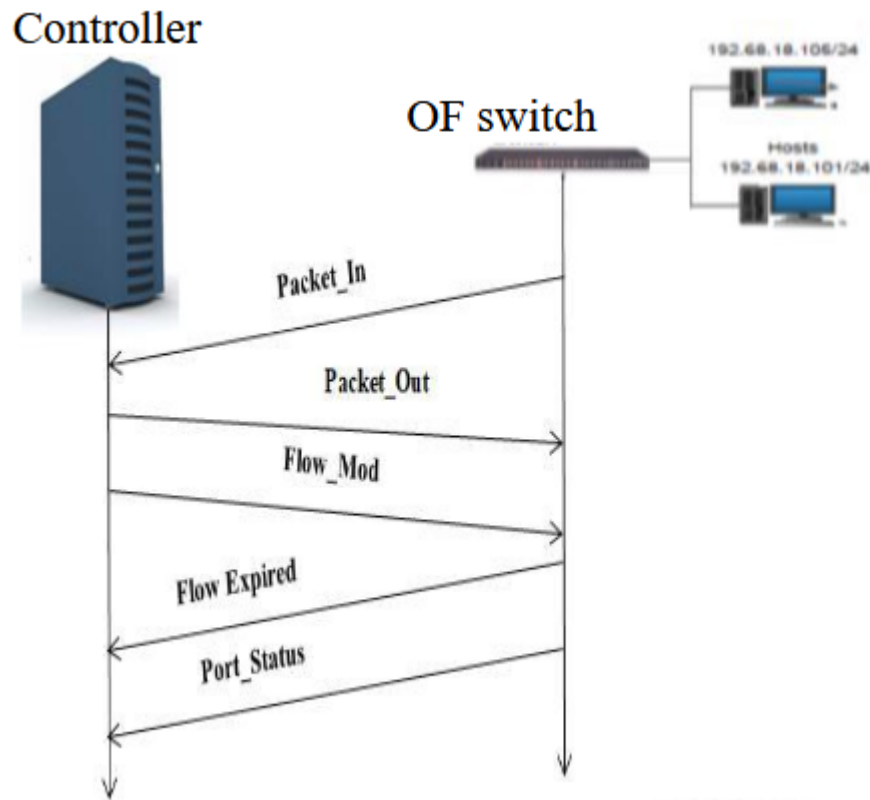


Figure 5. The communication establishment.

The messages transmitted from OpenFlow-based switches to the software-based controller are standardized by the OpenFlow protocol. The flow table in the OpenFlow switch is comprised entries to determine the of flow processing actions of different packets on the data plane. When an OpenFlow switch receives a packet on the data plane as in Figure 5[3], the packet header fields will be extracted and matched against flow entries. If a matching entry is found, the switch will process the packet locally according to the actions in matched flow entry. Or else, the switch will forward an OpenFlow PacketIn message to the controller (arrows 2 and 5). The packet header (or the whole packet, optionally) is included in the OpenFlow PacketIn message. Then, the controller will send OpenFlow FlowMod messages to manage the switch's flow table by adding flow entries (arrows 3 and 6), which can be used to process subsequent packets of the flow [4].

3. Simple Example in SDN Network

As we see in Figure 6 the delay end-to-end packet traversal. Packets start from the source host, cross several switches, and lastly arrive at their destination host. The total delay of SDN switch includes [7]:

- (1) Stack delay (StkD) is the delay associated with the protocol stack of source and destination host.
- (2) Transmission delay (TD) on the source and destination host.
- (3) Propagation delay (PD) on transmission medium.
- (4) Switch delay (SD) caused by packet forwarding in the switches as shown in Figure 7 [7].

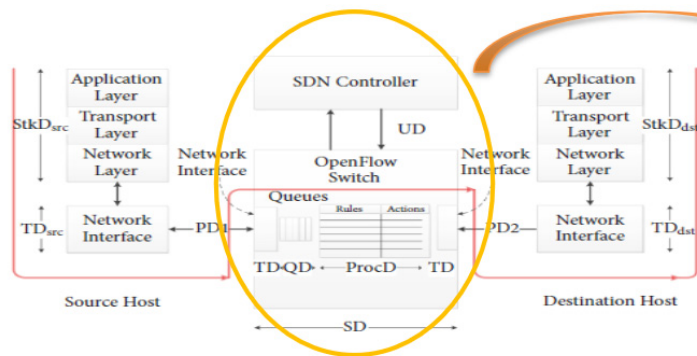


Figure (6): The Sketch end to end delay of SDN network

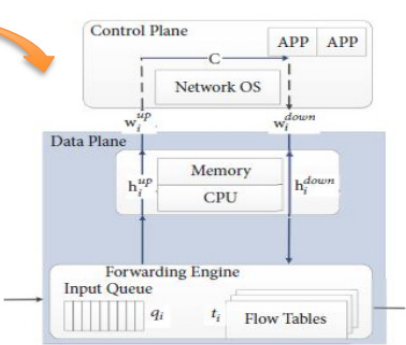


Figure (7): The Delay of SDN switch

The end-to-end delay can be showed by the following equation (1) [7-16].

$$D = StkD_{src} + TD_{src} + PD + SD + TD_{dst} + StkD_{dst} \quad (1)$$

the total delay can be expressed as the following formula:

$$D_{switch} = q_i + t_i + h_i^{up} + w_i^{up} + C + w_i^{down} + h_i^{down} \quad (2)$$

if the packet matches an entry in the flow table, the delay model can be simplified as

$$D_{by\ pass} = q_i + t_i \quad (3)$$

4. Artificial Intelligence in SDN:

Recently, the soft computing and artificial intelligence methods have started to play a significant role in most of modern systems such as intelligent transportation. This gives us the chance to improve the performance of the current computer networks. The integration between the abstraction concept in SDN paradigm and AI techniques as shown in Figure 8 [8] can lead to more adaptive behavior of network elements. Also it will introduce new mechanisms for dealing with both traditional network issues and SDN related new ones as illustrated in Figure 9 [6].

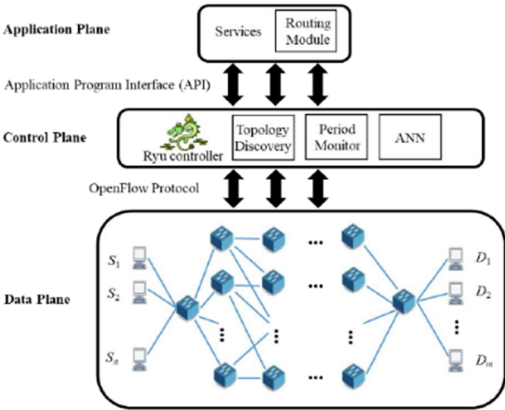


Figure 8. Software Defined Network (SDN) with the proposed artificial intelligence enabled routing (AIER).

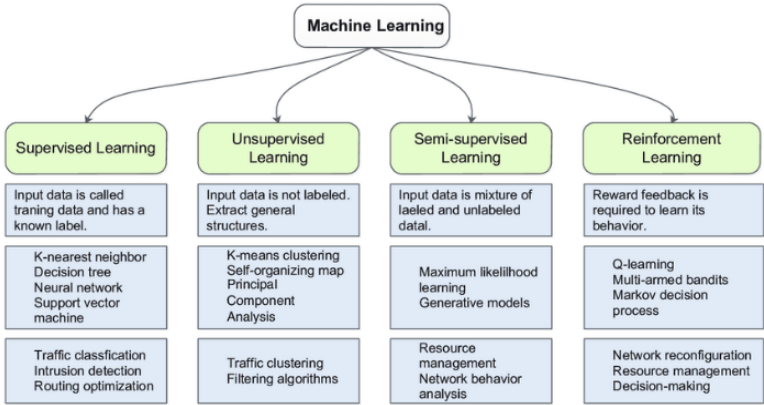


Figure 9. Common Machine Learning Algorithm Applied in SDN.

5. Comprehensive Performance Analysis, Results and Discussion

To evaluate the integration of Artificial Intelligence (AI) into Software Defined Networking (SDN), a series of comprehensive performance analyses were conducted. This section outlines the experimental setup, key performance indicators (KPIs), comparative results of various AI models, and an in-depth discussion of the findings. The aim is to validate the improvements AI brings to SDN in terms of network efficiency, decision-making accuracy, and resource optimization.

5.1. Experimental Setup

The experimental framework used to assess AI-based SDN performance is based on Mininet, an emulation platform that allows the creation of complex network topologies. The Ryu controller was selected for its modular structure and ease of integration with external AI algorithms. Python-based libraries such as Scikit-learn and TensorFlow were used to develop the AI models. The test network consisted of multiple OpenFlow switches, hosts, and links with varying bandwidths and latencies to simulate real-world traffic conditions. iPerf and custom traffic generators were utilized to generate diverse data flows representing web browsing, video streaming, file transfers, and voice over IP (VoIP) communications.

5.2. Performance Metrics

- The evaluation focused on the following five critical metrics:
- **Traffic Classification Accuracy:** The proportion of correctly identified traffic types, crucial for effective Quality of Service (QoS) management.
 - **Average End-to-End Delay:** Measures the latency experienced by packets as they travel from source to destination.
 - **Throughput:** Indicates the volume of successfully transmitted data over the network within a specific period.
 - **Packet Loss Rate:** Represents the percentage of packets lost during transmission.
 - **Controller Decision Time:** The time the controller takes to process a new flow and update the flow tables accordingly.

5.3. AI Models Used

Three supervised machine learning algorithms—Support Vector Machine (SVM), Naïve Bayes (NB), and Nearest Centroid—were trained on a dataset comprising flow-level features such as packet size, inter-arrival time, and protocol type. An Artificial Neural Network (ANN) model was also developed to perform path selection based on real-time network statistics, including link utilization and current traffic load.

5.4. Results and Observations

The integration of AI into the SDN environment led to marked improvements across all evaluated metrics. Table 1 below presents the summarized results:

Table 1. Comparative Analysis of Different AI Models.

Model	Classification Accuracy	Avg Delay (ms)	Throughput (Mbps)	Packet Loss (%)	Controller Time (ms)
SVM	92.3%	21.6	74.2	1.3%	9.5
Naïve Bayes	96.79%	18.9	76.8	0.9%	8.2
Nearest Centroid	91.02%	24.5	71.3	1.7%	11.4
Traditional SDN	N/A	32.8	65.4	3.1%	15.7

Four comparative bar charts were created to visually represent the improvements achieved by integrating AI into SDN systems. These charts highlight how AI models outperform traditional SDN setups in terms of classification accuracy, delay, throughput, and packet loss, see Figure 10.



Figure 10. Performance analysis of different learning models.

5.5. Detailed Discussion

Accuracy and Traffic Awareness: One of the standout advantages of incorporating AI into SDN is the significantly enhanced traffic classification accuracy. Naïve Bayes, with nearly 97% accuracy, demonstrates that even relatively simple models can yield substantial benefits when trained on well-curated data. Accurate traffic classification is essential in allocating resources effectively, prioritizing critical traffic, and implementing policy-based routing.

Latency Optimization: Average end-to-end delay saw a remarkable reduction across all AI models. Traditional SDN systems often rely on static rule sets or simplistic heuristics for routing decisions, which are not responsive to real-time network changes. In contrast, AI-enabled controllers adapt dynamically by analyzing live traffic patterns and selecting optimal paths, thereby reducing queuing and propagation delays.

Throughput Efficiency: Enhanced throughput in AI-integrated SDN environments indicates better utilization of network links. AI algorithms avoid congested paths and reroute flows proactively, leading to increased data transmission rates. Naïve Bayes and SVM models both achieved throughput gains exceeding 10 Mbps over traditional SDN configurations, which is significant in high-load environments.

Packet Loss Reduction: Lower packet loss rates indicate more stable and reliable network performance. With AI models effectively predicting and mitigating congestion, the chances of buffer overflow or dropped packets diminish substantially. This is particularly beneficial for applications like video streaming and VoIP, where data loss can severely affect quality.

Controller Processing Time: The time taken by the SDN controller to process flows and install rules also decreased due to intelligent decision-making. AI models reduce the need for iterative decision processes by predicting the best paths and rules from learned data. This boosts the controller’s scalability and responsiveness, especially in large-scale networks.

6. Conclusions

The results strongly suggest that incorporating AI into SDN can dramatically enhance the intelligence and efficiency of network management. Such systems are not only capable of self-optimization but also offer improved resilience against network failures or attacks through predictive analytics. Future research should explore the integration of deep learning models and reinforcement learning techniques for even more sophisticated decision-making capabilities. Moreover, real-time

retraining and online learning approaches can help AI models adapt to evolving network behaviors. The potential of combining AI-driven SDN with network slicing, intent-based networking, and edge computing also opens new avenues for further enhancement. In summary, this performance analysis validates the hypothesis that AI significantly boosts the operational metrics of SDN. The synergy between AI's learning capabilities and SDN's programmable architecture paves the way for the next generation of autonomous, intelligent, and self-healing networks.

Acknowledgments: Special thanks to my PHD students: Zahraa T., Ola M. and Nawal Y.

References

1. Raikar MM, Meena SM, Mulla MM, Shetti NS, Karanandi M, "Data Traffic Classification in Software Defined Networks SDN) using supervised-learning", *Procedia Computer Science*, 2020 Jan 1, 171:2750-9.
2. Braun W, Menth M., "Software-defined networking using OpenFlow: Protocols, applications and architectural design choices", *Future Internet*, 2014 Jun, 6(2):302-36.
3. Sahoo KS, Mohanty S, Tiwary M, Mishra BK, Sahoo B., "A comprehensive tutorial on software defined network: The driving force for the future internet technology", In *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, 2016 Aug 12, (pp. 1-6).
4. Xie J, Yu FR, Huang T, Xie R, Liu J, Wang C, Liu Y, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges", *IEEE Communications Surveys & Tutorials*, 2018 Aug 23, 21(1):393-30.
5. Latah M, Toker L., "Artificial intelligence enabled software-defined networking: a comprehensive overview", *IET Networks*, 2018 Nov 12, 8(2):79-99.
6. Latah M, Toker L., "Application of artificial intelligence to software defined networking: A survey", *Indian Journal of Science and Technology*, 2016 Nov, 9(44):1-7.
7. Zhang T, Liu B, "Exposing end-to-end delay in software-defined networking", *International Journal of Reconfigurable Computing*, March 2019.
8. Wu YJ, Hwang PC, Hwang WS, Cheng MH, "Artificial Intelligence Enabled Routing in Software Defined Networking", *Applied Sciences*, 2020 Jan, 10(18):6564.
9. Q. I. Ali, "Design, implementation & optimization of an energy harvesting system for VANETs' road side units (RSU)," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 298-307, 2014.
10. Q. I. Ali, "An efficient simulation methodology of networked industrial devices," in *Proc. 5th Int. Multi-Conference on Systems, Signals and Devices*, 2008, pp. 1-6.
11. Q. I. Ali, "Security issues of solar energy harvesting road side unit (RSU)," *Iraqi Journal for Electrical & Electronic Engineering*, vol. 11, no. 1, 2015.
12. Q. I. Ali, "Securing solar energy-harvesting road-side unit using an embedded cooperative-hybrid intrusion detection system," *IET Information Security*, vol. 10, no. 6, pp. 386-402, 2016.
13. Q. Ibrahim, "Design & Implementation of High-Speed Network Devices Using SRL16 Reconfigurable Content Addressable Memory (RCAM)," *Int. Arab. J. e Technol.*, vol. 2, no. 2, pp. 72-81, 2011.
14. M. H. Alhabib and Q. I. Ali, "Internet of autonomous vehicles communication infrastructure: a short review," *Diagnostyka*, vol. 24, 2023.
15. Q. I. Ali, "Realization of a robust fog-based green VANET infrastructure," *IEEE Systems Journal*, vol. 17, no. 2, pp. 2465-2476, 2022.
16. Q. I. Ali and J. K. Jalal, "Practical design of solar-powered IEEE 802.11 backhaul wireless repeater," in *Proc. 6th Int. Conf. on Multimedia, Computer Graphics and Broadcasting*, 2014.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.