IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 November 2022, accepted 10 December 2022, date of publication 20 December 2022,
date of current version 29 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3230985

RESEARCH ARTICLE

An Optimized and Secure Energy-Efficient Blockchain-Based Framework in IoT

MOHAMMED A. AL GHAMDI
Computer Science Department, College of Computers and Information Systems, Umm Al-Qura University, Makkah 715, Saudi Arabia
Corresponding author: Mohammed A. Al Ghamdi (maeghamdi@uqu.edu.sa)

This work was supported by the Data and Artificial Intelligence Scientific Chair at Umm Al-Qura University, Makkah, Saudi Arabia.

ABSTRACT With the rapid development of the IoT (Internet-of-Things), additional smart gadgets may be associated with the Internet, significantly enhancing data transfer and communication. Software-Defined Networking (SDN) is known as a new model that separates the control plane and the data plane, and is anticipated as a favorable solution for implementing Blockchain, to offer the scalability and adaptability required for IoT. The scalability of the network rises in direct proportion to the users' enhanced privacy on the network. Blockchain and SDN are two top innovations utilized to create secure network architectures and provide trustworthy data transmission. They offer a strong and trustworthy platform to deal with dangers and problems, including security, privacy, adaptability, scalability, and secrecy. Unfortunately, the attackers can still inject traffic to disrupt a blockchain node's regular functions. This study provides an optimized Blockchain-based SD IoT architecture for smart networks that is safe and energy-efficient. In this work, it is concentrated on blockchain-based SDN and creates an SDN-Blockchain Classifier. This IDS-based security tool provides a trust-based classifier by handling and reducing harmful traffic through traffic fusion and aggregation. Finally, it is concluded by evaluating the proposed framework SDN-Blockchain Classifier performance against MAC flooding attack in a simulation setting and demonstrating that it can attain optimized average throughput, response time, packet loss of crossing domain path, energy efficiency, end-to-end delay, file transfer operation, energy consumption, and CPU utilization compared to the baselines taken into consideration, thereby achieving efficacy and also security in the proposed smart network.

INDEX TERMS Blockchain, Ethereum, IoT, mininet, openstack, pythereum, software-defined networking.

I. INTRODUCTION

Blockchain [1] is drawing significant interest from numerous industries, including banking, healthcare, and government. Applications operate decentralized owing to blockchain [2]. There is no requirement for a centralized power or an intermediate agency to oversee party trades. Secure trades may be completed even in a trustless distributed system. Earlier, it was impossible to do this. The growth of new tendencies like smart cities, hospitals, businesses, classrooms, etc., has been seen as a result of the creation of the Internet and its supporting technologies like the Blockchain [3] and the Internet of Things (IoT) [4]. The demands of these innovations have also transformed how informative services are presented and received (e.g., e-Business, e-Governance, and e-service brand management). IoT [5] is expected to expand by 30 to 70 billion dollars in the industry by 2025; therefore, it is expected to impact many facets of daily life, especially communication significantly. Unfortunately, IoT [6] encounters many difficulties, such as those brought on by heterogeneity of devices, many assaults, and inconsistency. Due to its distributed aspect, IoT [7] has several problems with scalability, energy efficiency, and security, particularly as the quantity and variety of smart items grow rapidly. For instance, the variety of energy resources and processing in IoT [8] devices has constraints, which might lead to communication blockages and the deployment of security measures. Fog computing has recently been proven to be a practical framework for addressing issues related to IoT [9] source restrictions. But there are still unsolved security problems, including ones that are IoT-specific. For IoT [10] devices to communicate

The associate editor coordinating the review of this manuscript and approving it for publication was Kai Yang.

with one another, new solutions with qualities like secrecy, availability, and high security are needed. The messages are protected from access by ensuring secrecy. A transaction's integrity and the inclusion of all necessary parties guarantee that the message is delivered to its intended recipient and that any tampering is obvious.

Given that most Internet of Things (IoT) [11] devices are low-power with inadequate computation capacity, obtainability guarantees that any data facility is accessible whenever required while considering the energy spent by heterogeneous devices. Another crucial factor for data conduction in an IoT [12] network is reducing energy consumption while maintaining performance and security. A new structure for IoT [13] networks must be created because of this. In essence, an infrastructure with the necessary characteristics needed to ce created to stabilize the lower energy consumption and safety requirements for IoT [14] in the application, physical, and network layers. So, the implementation of such an infrastructure using Blockchain and software-defined networking (SDN) is analyzed [15]. An SDN based IoT architecture with Blockchain [16], which plans to improve resource management in IoT [17] networks, is shown in Figure 1. The orchestration and management architecture make up the three different environments: IoT [18], SDN [19], and Blockchain [20] environments.

A new network design known as SDN [21] includes two primary components, a controller, switches, and network users. The controller operates on a standard protocol, i.e., OpenFlow, to provide rules, management, and programmability of certain network switches, while the switch is responsible for packet routing. Therefore, SDN controller offers network control, programmability, connectivity and remote access, intelligent management, and high flexibility [22]. Additionally, the SDN controller [23] enables the implementation of unified and protected network facilities like security, routing, energy management, and bandwidth usage and may stop unwanted admittance to network resources. IoT and SDN functionalities can be coupled to improve network performance. Additionally, the SDN controller [24] may be used to control and maintain network configuration changes because of the dynamic nature of the IoT devices [25]. The absence of a centralized controller is one of main issues with an IoT network. This problem may be solved by employing an SDN controller to offer a unified controller for interaction with IoT devices. How to increase SDN security is one of the main issues being discussed right now. In SDN [26], the file transfer may be made more secure, for instance, by implementing blockchain. Blockchain's [27] security-by-design feature would be used through the SDN network since it guarantees user confidentiality and resource accessibility from unauthorized users regarding the IoT devices' energy-efficiency strategy.

One of the sophisticated and well-established techniques for securing internet interactions is Blockchain [28], [29], often known as distributed ledger technology (DLT). Additionally, Blockchain offers a framework that stores digital

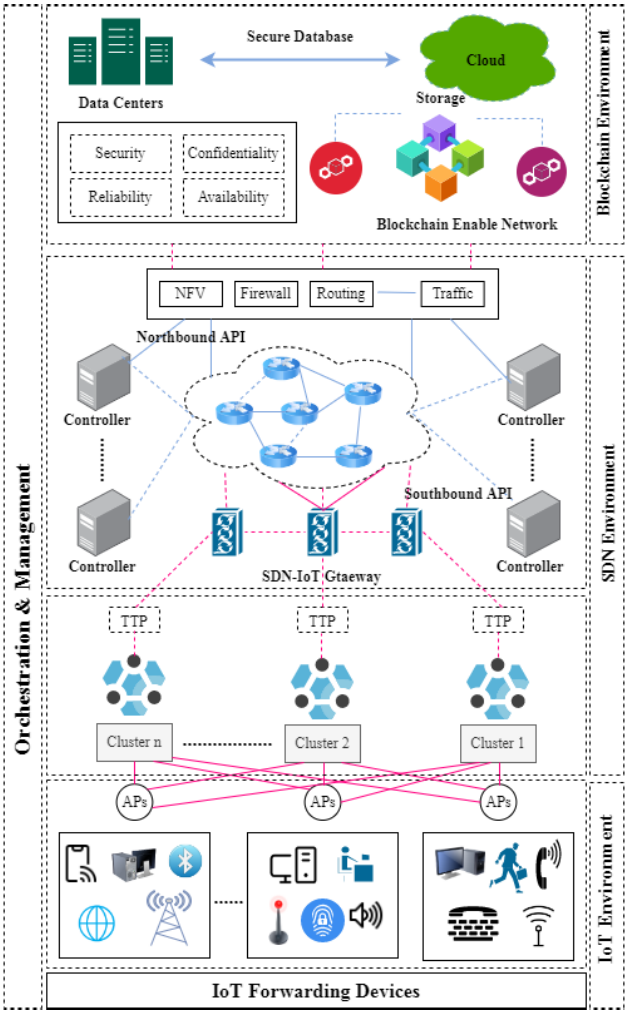

FIGURE 1. An architecture of SDN-based IoT with blockchain.

data and disperses it around the network, but it never permits other parties to change the data. As a result, a lot of financial and data management systems successfully utilize the benefits of Blockchain. Instead of centralized administration, it employs dispersed and adaptable peer-to-peer network management. It is made up of a series of blocks that each holds a specific piece of information. Without a middleman, communication is possible in an entirely secure manner. In order to build a safe and resource-conserving architecture, the problems and constraints of the IoT are addressed by utilizing the abilities of the SDN controller. This technology's P2P nature seeks to safeguard security, confidentiality, solidity, dependability, and the elimination of single point of failure issues.

Due to their distinct characteristics, incorporating Blockchain and SDN with IoT devices can be used to get better management and security. As a result of this inspiration, a novel technique has been developed in which blockchain technology is applied within an SDN network working with OpenStack. The main objective of this work is to introduce the basic concept of SDN, Blockchains, and

OpenStack with the motivation to combine these technologies in order to enhance the management and security as well of the whole system. The Blockchain is used to build a dispersed peer-to-peer (P2P) network where untrustworthy members communicate with one another in a verifiable way without participating of a third party. In this work, the smart contracts scripts (serpent programming for the Ethereum platform) are used in this study to operate the process of Blockchain nodes. The implementation of SDN with Blockchain on OpenStack acting as cloud storage systems makes it possible to share resources and services among hosts safely and privately. Customized SDN network topologies are simulated using the Mininet simulator. OpenStack controller has been integrated with the OpenDaylight controller. The OpenStack platform is used to store data in the cloud. The Ethereum platform added the Pyethereum diagnostic tool for Blockchain testing. The blockchain contracts are created using python programming.

This paper proposes a safe, energy-efficient blockchain-based approach for IoT. The suggested strategy uses SDN architecture to decrease network latency and enhance the quality of service (QoS). The benefit of adopting an SDN-enabled network over a traditional TCP/IP network is that SDN is independent and enables effective and dynamic management of the whole network architecture. The following is a summary of our contributions to this effort.

- We develop SDN-Blockchain Classifier (Graylist-Based Packet, Whitelist-Based Packet), a trust-based security tool for blockchain-based SDN. Through traffic fusion and aggregation, we reduce malicious traffic and protect blockchain nodes from attack.
- The proposed solution offers a classified framework with distributed network management for IoT devices by utilizing an SDN-based controller on blockchain technology devices to improve the trustworthiness between the transactions and securely increase the confidentiality of the data communication.
- Our proposed solution uses the Ethereum platform with proof-of-stake (PoS) method. Ethereum, a decentralized blockchain platform and well-known cryptocurrency has finally finished making the long-awaited transition to proof-of-stake.
- Our approach SDN-Blockchain Classifier outperforms the assumed baseline on energy use and end-to-end latency, according to the experimental assessment. Overall, compared to a traditional Blockchain, the SDN Blockchain-Based IoT framework achieves greater performance (in terms of average throughput, response time, packet loss of crossing domain path, energy efficiency, end-to-end delay, a file transfer operation, energy consumption, and CPU utilization).

The remainder of the paper is organized as follows: the SDN, IoT, Blockchain technology, and OpenStack is introduced in section 3. The related work is presented in section 4, while section 5 explains the proposed framework and its operations. The experimental setup and performance evaluation of the proposed architecture are covered in section 6. In section 7, a conclusion of the work and future work are stated.

II. BACKGROUND

In this section, OpenStack is used to build the tiny data center, OpenDaylight to operate the SDN network, and the Ethereum platform to run the blockchain-based system. In this part, a quick introduction to Blockchain technology, SDN, IoT, OpenStack and Ethereum is presented.

A. BLOCKCHAIN TECHNOLOGY

Blockchain technology has received a lot of interest from both academic and industrial sectors as a result of the success of the Bitcoin application. Building a blockchain that is resistant to temperature changes is the initial objective of blockchain technology. A typical blockchain consists of a set of documents (referred to as blocks) that are arranged in temporal order by distinct time signatures. A cryptographic hash is used to connect every block to the one before it; the first one is known as the genesis block. Various blockchain versions would have different specifications for what a block should include. A block typically contains content, a timestamp, and a strong cryptographic value of each block in the chain before it. Blockchains are often governed by a P2P network offering transparency and authentic data storage. This means that the information stored in any block cannot be changed retrospectively without changing all succeeding blocks. Each block in the chain is depicted in Figure 2 as having a list of transactions and a hash to the block before it. The global status of the data being shared on the network may be determined by any node that has accessibility to the ordered, back-linked list of blocks and can also read it.

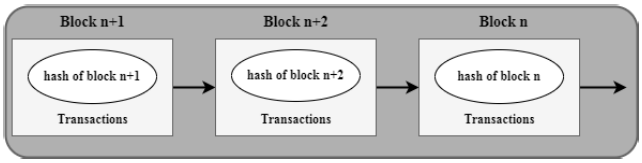

FIGURE 2. Blockchain: every block contains a brief list of transactions.

Blockchains may be divided into two categories: permitted blockchain and permissionless blockchain. The former enables any entity to participate in the consensus mechanism as a reader or writer. These types of blockchains have Bitcoin, Ethereum and Zerocash as some examples. However, the latter restricts the variety of organizations that can participate as readers or authors. Private permitted blockchains can still be dispersed among many places, although a single or centralized body frequently manages them. A consortium blockchain enables the permitted group to make consensus choices, and each contributing entity must register in order to participate in the network. Both private and public permitted blockchains would really impose limitations on authors during the consensus procedure. It is important to note that although private permitted blockchains would not even

provide read access, public permitted blockchains enable everyone to view the state. One example of a permitted blockchain is Hyperledger. In blockchain-based systems, numerous techniques exist to construct a decentralized consensus protocol for block verification.

- Proof of work: In this method, a network node can fruitfully submit a block for acceptance provided it can show that it expended a certain amount of computing resources (also known as ''work'') on it. The Bitcoin blockchain already uses a model protocol based on the cryptography hash algorithm SHA-256.
- Proof of stake: This consensus-building method relies on a blend of chance selection and the participating nodes' power (or ''stake''). It is predicated on the idea that organizations holding sizable stakes in the blockchain-based network have crucial attention in preserving its trustworthiness.
- Proof of passing time: By asking every possible validator for a protected and arbitrary waiting time from a trustworthy runtime situation, consensus based on this approach could be obtained; each participant must wait for the provided time, and the first one takes control of the verification authority.

1) ETHEREUM

The decentralized Ethereum platform enables the use of smart contract software. The applications for programmable smart contracts eliminate all potential for discrimination, deception, or third-party involvement. A custom-built blockchain, a potent worldwide shared structure that can transfer money around and reflect property proprietorship, is where smart contract applications are installed. This function makes it easier for developers to establish marketplaces, keep records of obligations or promises, and transfer money in accordance with instructions without the need for a middleman or taking on third-party risk. The Ethereum wallet is known as a gateway for decentralized applications on the Ethereum blockchain. It maintains and protects files created using the Ethereum platform. The Ubuntu 22.04 LTS operating system includes the Ethereum platform. Figure 3 depicts the platform levels of Ethereum's application of distributed ledger technology, where the Blockchain is created using the Pyethereum tester.

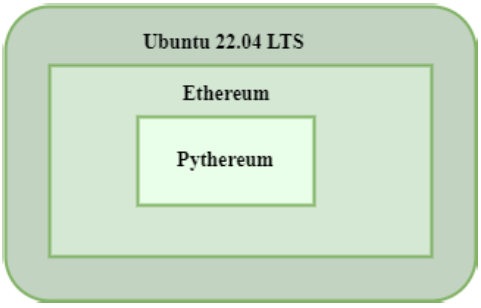

FIGURE 3. Platform levels.

B. INTRODUCTION TO SOFTWARE-DEFINED NETWORKING (SDN)

SDN helps the network personals, such as engineers, administrators and researchers, in adjusting network operations and protocols in a much simpler and more flexible manner, and this is can be happened through separating process that acours between data planes, software-defined networking and network's control planes. As opposed to local control, a centralized controller has a wider understanding of the resources under its control and may make better judgments on their deployment. SDN aims to accomplish two things by allowing switches to forward traffic by predetermined rules. The first one offers open user-controlled management of the forwarding plane while executing network activities without additional software. The second involves moving certain network calculating intricacy away from hardware-based network elements and toward a software-based controller. Numerous programmable switches and control entities make up SDNs, which make it possible to choose the best flow packet forwarding rules from faraway users to virtually generated computational resources. The three layers that make up a typical SDN framework (i.e. application layer, control layer, and infrastructure layer as the three planes) is represented in figure 4. The first layer is in charge of enforcing certain policies via the control layer's supported API. The SDN controller transmits the requests from the applications to the control layer, where it exercises finer control over the network components and provides pertinent data up to the SDN applications. In order to coordinate conflicting application requests for restricted network resources, an SDN controller is used. Network components are included in the infrastructure layer that also defined as the data plane, which can depict their competencies to the control layer through the packet forwarding interface. The controller can connect with lower infrastructure or upper application layers via a Northbound or Southbound interface, as appropriate.

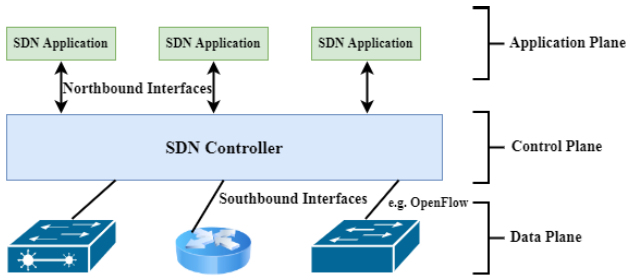

FIGURE 4. Simplified SDN architecture.

C. INTRODUCTION TO INTERNET OF THINGS (IoT)

IoT is anticipated to be the greatest impact after the advent of the Internet. The world of digital gadgets is expanding rapidly, and by 2010, there will be more devices on the planet than people. Similarly, projections indicate that by 2020, there will be around 50 billion gadgets online. In the

IoT era, where every gadget will have a digital identity, all these things will be implemented. The number of devices linked to the Internet has beyond all predictions because to advancements in technologies like low power and resource-constrained devices, which have extended the Internet's scope to the farthest reaches of the globe. Figure 5 showing the IoT platform.

FIGURE 5. Internet of things platform.

D. OPENSTACK

A cloud's computation, storage, and networking resources are managed via the Openstack cloud computing environment. It offers multi-cloud environments unified control of security policies. Either a public cloud or a private cloud can be made. To operate cloud computing services, cloud users can establish Virtual Machines (VMs). Layer 2 switching is used to connect the virtual interfaces of the VM to the physical interfaces of the host. Switch device and virtual Bridge Linux are used to do layer 2 switching, when it comes to forwarding the packets and MAC learning, Linux bridge functions like a regular Ethernet switch. As the default Configuration Controller for Openstack, Neutron is used. It is an API-driven, pluggable network controller. Instead of Neutron, the OpenDaylight controller is utilized in order to manage the SDN network in this research. The OpenStack platform is seen in Figure 6.

III. RELATED WORK

Several scholars have recently contributed to cutting-edge developing technologies, including smart networks, IoT, SDN, and Blockchain. This section discusses the previous research studies that used distributed secure blockchain-based SDN strategies in various contexts along with these works. Takenaka et al. [30] explored how factories might benefit from IoT data and provided an example of a smart network. This study emphasized the value of using the right data format and analytical techniques to achieve goals like mass customization or developing new facilities. Big wireless data (BWD), artificial intelligence (AI), and network function

FIGURE 6. OpenStack platform.

virtualization approaches are all used in the framework for smart networks that Huang et al. [31] suggested. In order to achieve the aim of Quality-of-Service (QoS), the author also let mobile clients access the best network (at a reasonable cost). By the smart-grid paradigm, Kazmi et al. [32] offered the idea of a smart distribution network and analyzed something from a policy point of view. They specifically emphasized the smart distribution network design concept and implementation operations. Etzioni et al. [33] proposed an improved safety and transportation method using ICT and automated vehicles. The authors created a traffic flow model for a smart city that closely resembles real-world traffic behavior to explore the effects of various traffic-related events. The author in [34] presented a contract-based game for energy trading between energy contractors and users to boost the inducement for all of these. They did this by using the idea of operations research. Because they may serve as mobile energy carriers to meet an area's energy needs, EVs are a significant participant in the energy trading market. In order to improve the efficiency of traffic processes in the city, Lv et al. [35] implemented a smart transportation scheme in Macao. The scientists employed support vector regression and deep belief networks to predict traffic conditions. The outcomes demonstrate that their model's inaccuracy in predicting traffic bottleneck was bearable and aided in effectively learning Macao's traffic behavior. ICT is now a crucial component of vehicular networks, which serves as the system's communication hub. The authors of [36] examine the network management and accessibility of linked devices for Wireless Body Area Networks (WBAN) and Ambient Assisted Living (AAL). They suggested that the SDN controller should monitor traffic streams and facilitate a smooth transmission of traffic regulations across network component devices for better addressing and mobility maintenance. In [37], Yang examined the number of controllers, which is the main determining element in any SDN and WBAN merger in the healthcare industry. In order to determine the optimal number of controllers for SDWBAN architecture while considering the quantity of controllers, latency, and SDN-enabled switches, the authors proposed a mathematical model

utilising the convex optimization technique (SDESW). The average of the experimental findings also supported their technical argument. The authors of [38] offer a unique method for decentralized identification and authorized rules control via the deployment of Blockchain in direction to retain a comprehensive view of the system's safety protocols. Their idea is integrated with the FIWARE platform. If opposed to pair methods, it performs better. A blockchain-based system for secure communication and monitoring in a smart energy setting was presented by Aggarwal et al. [39]. In order to verify the network's energy trading operations, the researchers select a few entities from the smart grid network to operate as mining operation nodes. The miner nodes generated a proof of work mapped to the hash function produced by the connected object. Investigating how energy trade inside a smart network organization may be made using blockchain smart contracts while maintaining user privacy, Khalid et al. [40] have launched a crypto trading research.

A. HYBRID ENERGY TRADING

In [41] discussed the viability of integrating Blockchain technology into Sdn controllers for security and privacy considerations. They concentrate on examining how Blockchain technology is now being used in SDN. A similar development gives network infrastructure secrecy, integrity, and scalability. Finnah et al. presented blockchain-based SDN in [42], provided a decentralized, reliable record of SDN data and broke down the division of multi-vendor devices for defect retrieval to lower the cost of fault recovery. Their suggestion was validated through simulation utilizing OpendayLight and Ethereum technologies. The authors proposed a general architecture of blockchain-based SDN in [43] to address the problem of centralized control planes. In this structure, the control plane and the application layer are combined to form one core element, and extra security measures are used to provide security. At that time, decentralized controller security might be improved with the use of blockchain technology. Comparison summary of the related work is shown in Table 1.

Summary: Takenaka et al. [30] focus on IoT's possibilities in B2C enterprises for the consumer electronics industry. In order to better understand how customers behave daily, they provide an analytical example using smart appliance logs from 600 users and their replies to a questionnaire about their lives. By utilizing WBD (Wireless Big Data), CR (Cognitive Radio), and NFV (Network Function Virtualization) approaches, Huang et al. [31] present a framework to convert HetNets to smart networks. Kazmi et al. [32] present and reviews the modern distributed network (SDN) idea inside the SG paradigm from a planning standpoint. On the premise of the SG packet, changes in the SDN planning process have also been examined (SGP). Through the major SG-enabling technologies (SGTF), anticipated functionalities (SGAF), new consumption models (MDC) as potential SDN candidates, related policies and pilot projects, and multi-objective planning (MOP) as a real-world optimization problem, the packet offers a foundation for SDN planning. A discrete distribution is used by Etzioni et al. [33] to depict the taste variability of several latent classes for shared automated vehicles. Latent class assignment is calculated simultaneously with a discrete choice kernel taking into account latent factors, socio-demographics, and travel behaviours. Using a Bayesian D Efficient design, respondents selected their favourite method of commuting to work from a series of stated preference choice tasks depending on the characteristics of their existing commutes. Zhang et al. [34] suggested a blockchain-based energy trading system for a community-based P2P market to solve the problems with demand-response management in P2P energy trading. The proposed energy trading system is prototyped on a cluster network, with a coordinator running as a smart contract on a Hyperledger blockchain. This allows for investigating the system performance in both on-chain and off-chain processing modes. Lv et al. [35] address the security issues with the Digital Twins (DTs) of the Cooperative Intelligent Transportation System (CITS) in a deep learning environment. Convolutional Neural Network (CNN) and Support Vector Regression (SVR) are integrated, the DL method is modified, and DTs technology is incorporated. The primary energy-efficiency strategies currently in use and their limitations in SDN architectures for IoT infrastructures are highlighted by Ainou et al. [36]. In order to achieve a better trade-off between energy consumption and delay, Yang et al. [37] demonstrate a combined sleeping scheduling and opportunistic transmission system in delay-tolerant marine wireless communication networks based on software defined networking (SDN). A charging-station-to-vehicle (CS2V) energy trading method is suggested by Baza et al. [38] The CS2V system is beneficial in densely populated areas where a daily capacity for multiple EV charging is required. The author also suggests a vehicle-to-vehicle (V2V) energy trading model that protects privacy. To control the demand response in a V2G environment, Aggarwal et al. [39] suggest a peer-to-peer (P2P) energy trading mechanism between EVs and the SPs. Khalid et al. [40] provide a safe and effective V2V and V2G energy trading system. The suggested plan also contributes to the advancement of environmental responsibility and the sustainability and dependability of smart cities. A peer-to-peer energy trading system was suggested by Hebal et al. [41] to provide an independent and regulated Energy Internet (EI). Author developed a new energy routing strategy by maximizing the cost of energy and transmission losses, a subscriber matching system is created to decide which producer or producers should be assigned to each customer. Both consumers of single sources and multiple sources will find a solution in this system. Finnah et al. [42] take into account the dilemma of a power producer with two storage systems a battery and a hydrogen-based storage system and sells energy generated by wind turbines on the continuous intraday market. A backwards-approximated

TABLE 1. Summary related work.

Ref.	SDN	Blockchain	IoT	DataSet	Tool	Attack Diversity	Format
Takenaka et al. [30]	×	×	✓	N/A	N/A	No	IoT log data
Huang et al. [31]	✓	×	×	N/A	N/A	No	Packet, Flow
Kazmi et al. [32]	✓	×	✓	N/A	N/A	Yes	Packet, Flow
Etzioni et al. [33]	×	×	✓	Kaggle	MIMIC	No	Other
Zhang et al. [34]	×	✓	×	N/A	Hyperledger	Yes	Packet, Flow
Lv et al. [35]	×	×	×	Kaggle	SVR	Yes	Packet, Flow
Ainou et al. [36]	✓	×	✓	N/A	N/A	No	Packet, Flow
Yang et al. [37]	✓	×	×	N/A	CVX	No	Packet, Flow
Baza et al. [38]	×	✓	×	N/A	N/A	Yes	Other
Aggarwal et al. [39]	×	✓	×	N/A	Ethereum	Yes	Other
Khalid et al. [40]	×	✓	×	N/A	LAG	Yes	Other
Hebal et al. [41]	×	×	×	N/A	Matlab	No	Packet, Flow
Finnah et al. [42]	×	×	×	Hourly wind data from a weather station	Matlab	No	Electricity Flow
Wang et al. [43]	×	✓	×	N/A	N/A	No	Energy Flow

dynamic programming approach with an optimal computational budget allocation solves the issue. Wang et al. [43] present a unique hybrid community P2P market architecture for multi-energy systems. A data-driven market surrogate model-enabled deep reinforcement learning (DRL) technique is suggested to support P2P transactions within the technological limitations of the community delivery networks. To provide privacy protection, a deep belief network (DBN) based market surrogate model is created to define the P2P transaction behaviours of peers in the community without revealing their personal information. However, our proposed technique offers an optimal, secure, and resource-conserving SDN IoT architecture for smart networks. We design an SDN-Blockchain Classifier with a focus on blockchain-based SDN. This IDS-based security technology manages and reduces dangerous traffic through traffic fusion and aggregation, providing a trust-based classifier.

IV. PROPOSED SOLUTION ARCHITECTURE

In this work, we develop SDN-Blockchain Classifier (Graylist-Based Packet, Whitelist-Based Packet), a trust-based security tool for blockchain-based SDN. Through traffic fusion and aggregation, we reduce malicious traffic and protect blockchain nodes from attack. In a practical implementation, an incoming packet must first travel through the graylist-based packet classifier module to determine if its IP address is on the graylist. If a match is discovered, the condition for this packet in the look-up table can be examined. When the first requirement is met, the packets distribute the IP addresses module for validation. If the validation is correct, packets must be sent straight to the secure blockchain environment without going through an IDS check. If the validation is unsuccessful, the payload must be halted. If anyone criterion is not satisfied, the packets will be sent to the whitelist-based categorization module under the second condition. If the packet's IP address is not on the graylist, it must be forwarded to the whitelist-based packet classifier module. Below is a list of the procedures for checking an IP address. The payloads must be compared with the lookup database's signature, if the packet's IP addresses are recognized in the whitelist. These packets will be blocked if a match is found, and a warning will be produced. The general architecture of MAC flooding attack is firstly presented and then propose SDN-Blockchain Classifier and security mechanism IDS-based that reduces harmful traffic and safeguards blockchain-based SDN from flooding assaults. To increase the safety of the SD-IoT system, assaults that originate inside SDN should be identified and halted. Accessibility features should be used to enhance system performance in opposition to system and network failures, incorporating legacy safety demands into account. Vulnerable system components should also be quickly recognized and disabled before they may negatively affect the network. Figure 7 depicts the danger model that is considered while defending against various threats.

Our proposed solution uses the Ethereum platform with the proof-of-stake (PoS) method. Ethereum is a decentralized blockchain platform that creates a peer-to-peer network for safely executing and validating smart contract application code. Participants can do business with one another using

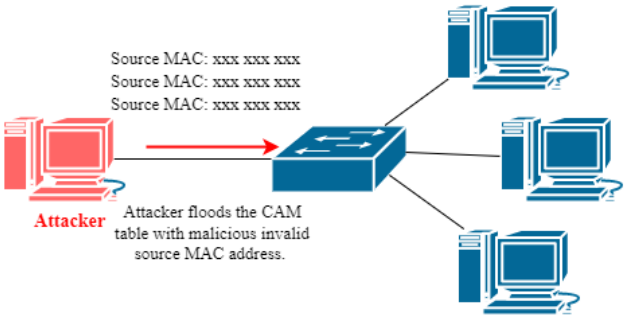

FIGURE 7. MAC flooding attacks.

smart contracts without the need for a reliable central authority. Participants have complete ownership and visibility over transaction data since transaction records are immutable, verifiable, and securely disseminated across the network. Ethereum, a well-known cryptocurrency, has finally finished making the long-awaited transition to proof-of-stake. The Ethereum blockchain formerly relied on proof-of-work, a consensus process that demands a significant amount of computing energy from each decentralized node involved in the network. However, the proof-of-stake methods fundamentally alter how the Ethereum blockchain functions. Since staked ETH and validators now protect the network, there is no longer a need to mine blocks. The proof-of-stake has many advantages: higher energy efficiency, fewer access barriers, fewer hardware needs, and decreased centralization risk.

A. MAC FLOODING ATTACK

MAC flooding attack aims to disrupt the network switches' safety. Typically, network switch has a specific table with entries related to MAC addresses of the nodes called MAC table. The entries in the MAC table are the unique MAC addresses with respect to the switch linked with that device. Using this table, the switches may now guide data out of the ports to the recipients. As already discussed, switches transmit data to the respective computer or machines intended to receive that data. In contrast, hubs disseminate data to all ports that enable the data scope to all hosts on the network. The main objective of the MAC Flooding attack is to abolish this MAC table. During a typical MAC Flooding attack, the attacker tries to transfer a large volume of Ethernet frames. Many Ethernet structures will be sent to the switch, each with a unique sender address. This way, the attacker fills up the memory switches, where the addresses of MAC are stored in database. Due to MAC address storm, the appropriate users will be thrown out MAC addresses from the MAC table. The switches are no lengthier capable of sending the arriving data to the intended ports. Consequently, all ports will be overwhelmed with a sizable quantity of incoming frames.

The MAC Address Table is full and cannot store any further MAC addresses. It will cause the switch to go into a screwup state and function exactly like a network hub from that point on. How to eliminate this situation of MAC flooding attack to resume the normal network traffic? The packet headers meant for the victim's PC would also reach the attacker as they are connected to the network. Because then the attacker will be able to steal private information from the victim's and other machines' communications. Commonly, these confidential data are captured using a packet analyzer. The attacker has the option of initiating an ARP spoofing attack after initiating a MAC Flood assault. As a result, the attacker will be able to continue having accessibility to confidential material even after the switches under attack have improved from the MAC Flooding attack. In an ARP spoofing attack, the attacker attempts to have their MAC address linked with a legitimate network user's IP address by sending fake Address Resolution Protocol (ARP) messages. Using the Address Resolution Mechanism, a process employed by the Internet Protocol, a physical address, such as a MAC address, also known as an Ethernet address, is mapped from a machine's IP address.

1) HOW CAN THE MAC FLOODING ATTACK BE STOPPED?

There are several ways to mitigate the MAC flooding attacks, some of them are as follows:

- Switch Port Security
- Using the AAA security server to authenticate the user
- Security precautions to avoid ARP or IP spoofing
- Put IEEE 802.1X suites into use

B. SDN-BLOCKCHAIN CLASSIFIER

When blockchain installed on SDN controllers, then blockchain gateway must deliver transaction or block to other instances that must be in sync with the present state of the blockchain. Attackers still have an option to damage the blockchain-based SDN's capabilities by performing MAC operations that prevent some nodes from sending or receiving any data at all. An assault of this nature would be particularly dangerous to a consortium blockchain system with few nodes. It is crucial to implement suitable security systems to detect and handle harmful traffic in order to defend consortium blockchains against MAC attacks.

In this research, an SDN-Blockchain classifier, an IDS-based security method which uses a list base traffic fusion and classifier approach are proposed to assist and reduce dangerous traffic. The slightly elevated structure of the SDN-Blockchain classifier is shown in Figure 8. It comprises of a monitoring mechanism, an IP verification mechanism, a graylist-based packet filtering component, and a whitelist-based packet classifier component. In practice, the SDN controller would be installed along with IDS like Cobra to defend the entire network from different threats. The list base packet classifier is frequently used in front of IDS to classify a lot of network data. The two main parts of traffic classification are a graylist-based packet classifier component and a whitelist-based packet classifier. Whereas the latter is in charge of lowering traffic based on a whitelist, the former tries to classify network packets based on a graylist. The installed IDS and the whitelist-based packet classifier provide network

data (such as alerts) used by the monitor engine to calculate IP reputation. In this study, weighted ratio-based whitelist creation allows the monitor engine to update the whitelist regularly. The IP identification module checks whether an agent signifies the facility or the user it privileges to signify.

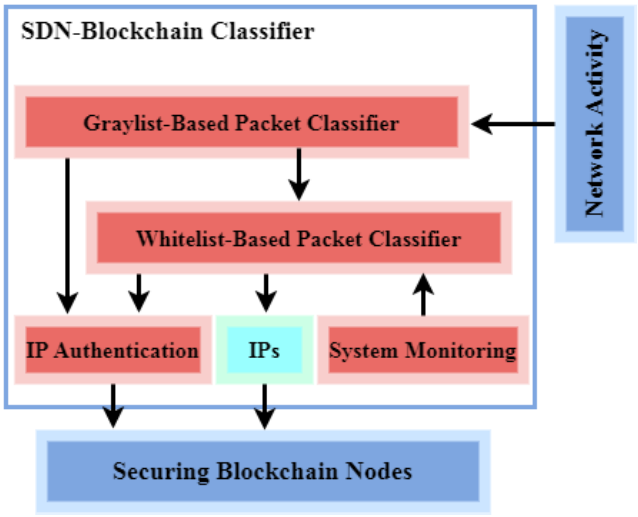

FIGURE 8. Proposed architecture of SDN-blockchain classifier.

1) 5.2.1 GRAYLIST-BASED PACKET CLASSIFIER

This component consists of two sections: a look-up table for comparing and a whitelist containing all whitelisted IP addresses. Security administrators can put up certain special restrictions to mitigate the possible danger and lessen the negative effects in real-world circumstances because of how sensitive whitelists are. To hold both general and particular conditions, the lookup table can include two sub-tables. Firstly, general conditions: This database keeps track of all extra criteria that must be checked with incoming traffic, including Flag data, network conditions, etc. Secondly, particular conditions: This table has the capacity to hold unique circumstances pertaining to a source IP address in a certain network setting. Security managers may often establish these requirements in practice using previous data processing and domain expertise (i.e., the experience of experts). If and only if the arriving packets meet all requirements, they can be sent to the destination network.

Summary: The arriving packet can be immediately compared with generic circumstances if no particular requirements are defined. The packets are transmitted straight to the IP Authentication server if all requirements are satisfied. The whitelist-based packet classifier module must analyze the packet if someone's criteria are not met. If any particular requirements are specified, the arriving packet must first be checked against those requirements. The package can be checked for all normal conditions if all special conditions are satisfied. The whitelist-based packet classifier module must analyze the packet if any criteria are not met. The packet can only get to the IP authentication Module to verify its originating address if and only if all particular and general requirements are met.

2) WHITELIST-BASED PACKET CLASSIFIER

By matching the arriving contents with the IDS rules that have been previously saved based on their IP addresses, this module can assist in filtering network traffic. The supervise engine may determine the IP reliability by gathering relevant network data broadcast by the installed IDS and whitelist-based packet classifier module. It maintains the whitelist by the following weighted ratio-based whitelist generation techniques as indicated in Equation 1.

$$IP\ Certainty = \frac{\Sigma_e^m = 1\ e}{\Sigma_c^n = 1\ 10 * c}\ (m, n\ \in\ M) \tag{1}$$

where 'e' denotes the number of excellent packets, 'c' is the amount of poor packets and ten is the rate of weight. In prior analysis, the rate of weight is varied between 1 to 30. Thus, the stability between the wrong positive and wrong negative rates could be achieved at a load of 20. It can be refer to the earlier work for further information on the creation of whitelists and the effects of the weighting factor. Such trust calculation can have two key benefits: first, it is simple to implement; second, security administrators can easily adjust the rate of weight and manage the effects of errant packets in a practical setting (i.e., it can enhance the rate in complex systems). A look-up table and a whitelist with all banned IP addresses are specifically included in the whitelist-based packet classifier module. The look up table consist further two tables in total, containing sub-tables for both coordinated and for IDS Signature. The prior stores of all IDS Signature are presently in use, whereas the latter keeps track of the entire IDS Signature that has ever been coordinated. If a coordinator is discovered for that IP address, the Signatures must be added to coordinate IDS Signature.

Summary: The table of all IDS Signature must be consulted if the database of matched IDS Signature does not include the desired signature. If a signature is available in the database of verified IDS signatures, it must be compared to the incoming material. The whitelist packet classifier will alert the security supervisor if a match is made, and the database of verified IDS signatures must be streamlined to include the confirmed signatures for the relevant whitelisted IP address. The packet can proceed to the IP verification module to check the source IP address if no match is found for the signatures.

C. SECURITY MECHANISM IDS-BASED

The usage of list-technique makes it susceptible to IP spoofing attacks, in which a hacker may fabricate an IP source on a packet. This module uses the source verification procedure to identify altered IP addresses to reduce this issue. Verifying whether a given entity actually signifies the facility or user it purports to signify is the basic goal of IP verification. The plan is for this module to deliver a reply packet to the client side in response to a connection

request. The verification is successful if the client provides a legitimate answer; the relevant packet should be stopped (and recorded). A popular subject in the literature is how to validate source IPs. This paper only utilizes a simple but effective preventative strategy as part of its investigation and is created on Java open source software which used P2P application. The user module of the programmer installed inside source host which is the IP authentication component of the programmer installed in the destination host. The Security Mechanism IDS main functions are shown in figure 9.

Summary: When a packet reaches, the IP identification module transmits the target client module a challenge that includes a TCP packet and a key N that was created randomly. The user module must send back a reply that includes a TCP packet and a key (N + 1) if it gets the query with key N. The module determines if the incoming key value is accurate by examining the obtained answer, including the TCP packet and the key (N + 1). If yes, the IP source's authenticity has been successfully verified. A faked IP source is one where the answer is incorrect or where the IP validation module does not receive the TCP packet.

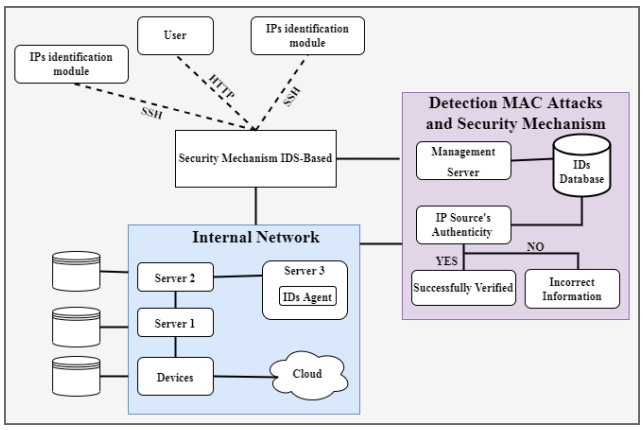

FIGURE 9. Security mechanism IDS main functions.

D. OVERALL DISCUSSION

To verify whether an arriving packet's IP address is on the graylist in a realistic implementation, an arriving packet must first pass through the graylist-based packet classifier module. If a match is found, the look-up table's condition for this packet can then is checked. At first condition, the packets spread the IP addresses module for validation if every condition is satisfied. The packets must be routed directly to the secure blockchain setting without going through an IDS check if the validation is accurate. Payload must be stopped if the validation fails. At second condition, the packets will be routed to the whitelist-based classification module if any one criteria is not met. However, if the IP address of the packet is not included in the graylist, it must be routed to the whitelist-based packet classifier module. The processes to verify an IP address are listed below. At third condition,

Algorithm 1 SDN-Blockchain Classifier

Input: Arriving Packets
Output: Notifications in case of MAC Flooding attacks are found
START
1: Mode = W (Whitelist-Based Packet Classifier)
2: While (A=arriving packets())
3: {
4: Preprocessing (P)
5: **if** (Mode = W)
6: // Graylist-Based Packet Classifier IDs
7: B = CD(A) where B is built form a set of A
8: Classify B using IDs mechanism
9: if (MAC Flooding attack founded)
10: {
11: Mode = G
12: Generate Notifications
13: }
14: **else**
15: {
16: // Whitelist-Based Packet Classifier IDs
17: B = SD(A)
18: Classify B using IDs mechanism
19: if (MAC Flooding attack founded)
20: Generate Notifications
21: if (MAC Flooding attack not founded within specified time)
22: Mode = W
23: }
24: }
END

if the IP addresses of the packet are identified in the whitelist, the payloads must be matched to the signature in the look up database. If a match is discovered, these packets will be stopped, and a warning will be generated. Additionally, a message indicating the generated warning and the state of the illegal IP address will be transmitted from this module to the monitoring engine. The packet must be submitted to the IP identification module if the payload does not match any of the signatures that have been saved. This can only be permitted to access the blockchain ecosystem if and only if it passes the IP validation. At fourth condition, if the IP address of the payload is not included on the whitelist, the installed IDS must handle it conventionally. After packet inspection, the installed IDS must report the packet's status (such as good or bad) and the pertinent IP source to the monitoring engine, which assists in computing the IP trustworthiness and routinely updating the whitelist. The classifier module utilizes the same signatures present in the installed IDS to retain the security level offered by that system. This aims to offer traffic classification without compromising overall network security.

Algorithm IV-D shows the IDS moving between SDN-Blockchain nodes. A two-grain intrusion detection system

(IDS) (Graylist-Based Packet Classifier and Whitelist-Based Packet Classifier) are proposed. The Whitelist-Based Packet Classifier IDs level is ideal in most situations, when intrusions are not discovered, in order to improve IDS performance. The Graylist-Based Packet Classifier IDs can identify potential attack details as soon as a Whitelist-Based Packet Classifier ID detects any intrusion. The mode W denotes the Whitelist-Based Packet Classifier IDs and mode G denotes the Graylist-Based Packet Classifier IDs. Five standard characteristics will be examined, and the Graylist-Based Packet Classifier IDs is represented by lines (6-13). The more aspects of packets it is checked, the more accurate it can be achieved. Because of this, it is suggested that the Whitelist-Based Packet Classifier IDs lines (14-23) check additional packet-related data. In order to input the extracted characteristics directly into the model and quickly ascertain the type of connection, the features are arranged as connection records. This allows the model to respond appropriately. Because it needs to wait for a sufficient number of packets to construct the connection record, the Whitelist-Based Packet Classifier IDs produce findings that are more accurate than those produced by the Graylist-Based Packet Classifier IDs.

V. EXPERIMENTAL SETUP

This section provides an appropriate explanation of our experimental setting. Two controller nodes, two computing nodes, two network nodes, two management nodes and two data network nodes are present in our OpenStack-based system. We use Ethereum platform, Ethereum, a well-known cryptocurrency, has finally finished making the long-awaited transition to proof-of-stake. The Ethereum blockchain formerly relied on proof-of-work, a consensus process that demands a significant amount of computing energy from each decentralized node involved in the network. However, the proof-of-stake methods fundamentally alters how the Ethereum blockchain functions. Since staked ETH and validators now protect the network, there is no longer a need to mine blocks. The proof-of-stake comes with its many advantages: higher energy efficiency, fewer access barriers, fewer hardware needs, and decreased centralization risk. Every PC runs on Linux. The data network's OpenFlow switches are all Open Vswitch 2.17.2 devices. Every port on all Open Vswitches has a 10 Mbps transmission rate. One Red Hat Enterprise Linux (RHEL) server $\times$86-64, version 8.3 computer is set up to install Red Hat OpenStack. A specialized cloud controller node is employed on one system, while a Nova compute node was used on the other. An $\times$86-64 bit processor for the support of Intel 128 CPU extension and Intel VT hardware virtualization capabilities is used to configure the OpenStack controller. In furthermore, 6 $\times$ 2 Gbps Network Interface Card (NIC) is employed, together with 8 GB of RAM and 1 TB of disc capacity. A 64-bit x86 CPU for the OpenStack compute node with provision for the Intel 128 extensions and the Intel VT hardware virtualization extension turned on, are utilized. The computational node has 4 GB RAM, 60 GB of disc space, and 4 $\times$ 2 Gbps NIC card. Red Hat OpenStack is deployed using the PackStack software, with constructing a 410 GB cinder partition. OpenDaylight Lithium is used as an SDN controller, while Mininet is utilized to create 14 SDN hosts in SDN network. Compared to other controllers (e.g., pox, ryu, onos, etc.), the OpenDaylight controller is the most secure and trustworthy. The Linux Foundation created and maintained OpenDaylight (ODL), a Java-based open-source controller. ODL's modular design allows developers to plug in new apps utilizing northbound APIs. ODL supports OpenFlow and other The Internet Engineering Task Force (IETF) standard protocols for southbound communication. We choosing OpenDaylight over many other controllers for four main motives: first, because it is an open-source, the code is continuously reviewed for security flaws and improved security; second, because ODL's modular architecture offers greater performance benefits than those of other controllers under consideration; third, ODL has provided security mechanisms for both NBI and SBI; and forth, ODL is backed by a security team that works round-the-clock to identify, patch, and monitor new and open flaws that hackers might use.

TABLE 2. Simulation parameters.

Simulation Parameters	Values
General Parameters	
Cloud storage platform	OpenStack
Network Simulator	Mininet / Pyethereum
Packet analyzer	Wireshark
Language	Python
IoT Parameters	
Mobile model	Random waypoint model
SDN controller	3
Type SDN controller	OpendayLight
IoT devices	50
Simulation time	200s
IoT devices speed	20 m/s
Area	1500 m * 1500 m
Packet size	1024 byte
SDN Parameters	
SDN controller	3
Type SDN controller	OpendayLight
Routing Protocol	OpenFlow
Blockchain Parameters	
Blockchain platform	Pyethereum tester for Ethereum
Protocol	Proof of Stack, Proof of Work
Block size	Number of connections fitting into a block

Table 2 lists each of the additional simulation parameters in detail, arranged according to the allusion technology (i.e. SDN, Blockchain, and IoT). The devices that make up the IoT-SDN network under test emit packets, and Wireshark is used to collect and analyse those packets. We use X11 tunnelling with ssh to analyze network traffic in real-time. WireShark is launched after starting Mininet in the terminal window, and it is configured to s1-eth1. We open xterm window on ports h1 and h2 to view a TCP connection. Next, we execute netcat 10.0.0.1 5432 on h2

after running netcat −l 5432 on h1. The TCP exchange is seen immediately after the ARP exchange. The IoT environment is specifically replicated for 200 s, with 50 IoT devices dispersed throughout a 1500 m * 1500 m region and moving according to the random waypoint model of mobility. Every device emits packets at a data rate of 10 Mbps that range in size from 1 byte to 1024 bytes. In a virtual computer running an OpenStack-initiated clouds image of Ubunto 22.04.0 LTS the Pyethereum tester for Ethereum is installed. Without interacting with the Blockchain itself, it is used to practice with smart contracts. Our diminishes for participant cataloging in the Blockchain as well as file distribution in the blockchain network are written using cobra programming, which is similar to Python. The Blockchain executes the program code that is generated by the snake code. The Pyethereum tester tool produces the blockchain states. Before starting 2 VM instances in OpenStack, it is important to prepare a safety cluster, generate an SSH key pair (to login to the illustration), and assign floating IP addresses to the vms so that they may be accessed from the outside domain.

VI. PERFORMANCE EVALUATION OF PROPOSED SOLUTION

In this section, the proposed method efficiency using various parameters such as average throughput, response time, packet loss of crossing domain path, energy efficiency, end-to-end delay, file transfer operation, energy consumption, and CPU utilization compared to some methods under flooding attacks are appraised. Blockchain Fundamental (BCF), AS Cooperative Inter-domain Reputation (ASCIR) and Blockchain-based SDN-enabled Secure Routing (BSDNSR) are methods that Yazdinejad et al. and Zeng et al. have used in their scenario. In our scenario, we have compared our proposed solution with BCF, ASCIR and BSDNSR. If we talk about our proposed solution performance, we use the Ethereum platform. A well-known cryptocurrency called Ethereum has completed the long-awaited switch to proof-of-stake. Proof-of-work, a consensus method that requires a considerable amount of computational power from each decentralized node active in the network, was once the foundation of the Ethereum blockchain. The proof-of-stake techniques, however, significantly change how the Ethereum blockchain works. There is no longer a need to mine blocks because staked ETH and validators now secure the network. The proof-of-stake has various benefits, including improved energy efficiency, fewer access restrictions, hardware requirements, and less centralization risk. So, this is the reason why our results are promising.

A. AVERAGE THROUGHPUT

The complete throughput time or complete operation time of transactions refers to the amount of transactions demands made by IoT devices in a network. Additionally, it schematically compares the proposed paradigm with the BCF, ASCIR, and BSDNSR. Moreover, It has been shown that performance is roughly identical amongst nodes with less numbers. However, as the number of nodes grows, so does the throughput. Additionally, It is found that after a certain period of time, the efficiency of the proposed framework with the respect of security and secrecy is significantly superior to BCF, ASCIR, and BSDNSR methods. Following the throughput comparisons, the authors reported that the suggested model outperformed in terms of performance. The processing time and time overhead are reduced by using an efficient algorithm and filter structure for IoT nodes, which increased throughput in comparison to BCF, ASCIR, and BSDNSR methods is shown in Figure 10.

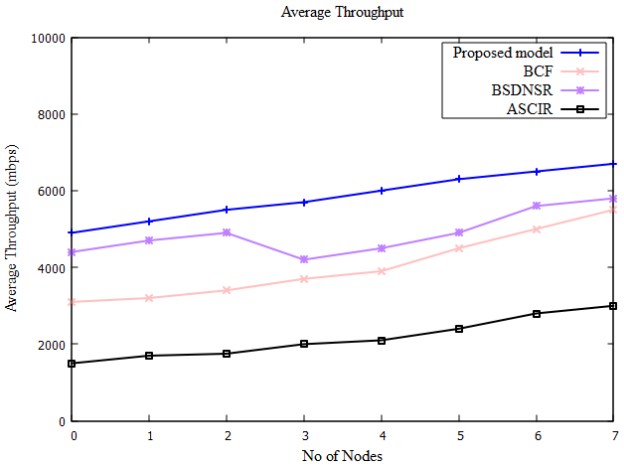

FIGURE 10. Proposed architecture average throughput with BCF, ASCIR, and BSDNSR methods in mobility setup: A comparative summary.

B. RESPONSE TIME

It concerns the amount of time it takes for files to be sent between two Internet of Things devices, and it can be seen that our solution is faster than the conventional one since the controller uses a proprietary routing protocol. The average response time for file transfers of various bulks among IoT nodes is shown in Figure 11, where our technique performs better than the BCF, ASCIR, and BSDNSR methods due to its lower overhead. An efficiency study depending on the number of nodes is successfully shown in Figure 11. Response times for both rise in proportion to the number of nodes. Additionally, It has been stated that the proposed technique achieves better than the BCF, ASCIR, and BSDNSR methods when fewer frequent assaults are involved. By utilizing the cloud platform's offered design, all nodes receive a prompt response.

C. COMPUTATION OF LOSS RATE FOR CROSSING DOMAIN PATH PACKET

Figure 12 depicts the rate of packet loss rate on a crossing domain link as time goes on. The host in the domain of controller c1 kept sending packets to the host in the domain of controller c2. The path passing the domain of the controller

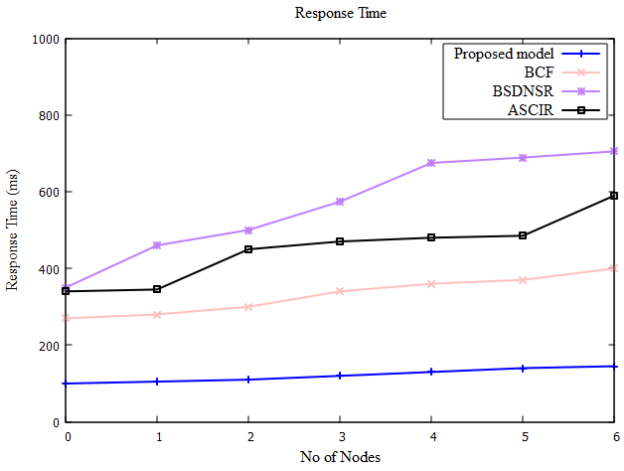

FIGURE 11. Proposed architecture response time with BCF, ASCIR, and BSDNSR methods in mobility scenario.

c3 was first set. The controller c3 eventually developed malignant intent. The findings show that PSDNIR's packet loss rates spiked, reaching almost 99.99%, indicating that all packets that travelled over the malevolent domain were discarded. When the controller c3 turned hostile at time t = 18 min, all BCF, ASCIR, and BSDNSR changed their pathways to traverse the controller c2's domain. However, proposed packet loss has been recovered more quickly than BCF, ASCIR, and BSDNSR methods as shown in Figure 12, and the rate of loss minimization has increased by roughly 23%.

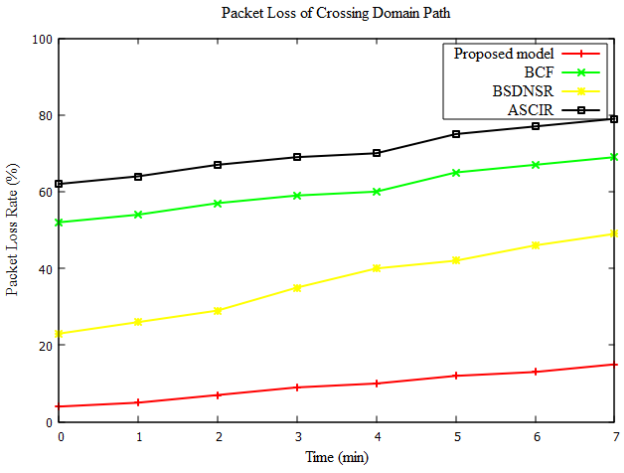

FIGURE 12. Packet loss of crossing domain path of the proposed architecture with that of BCF, ASCIR, and BSDNSR methods in mobility scenario.

D. ENERGY EFFICIENCY

In the Blockchain-enabled SDN-IoT framework, energy efficiency is one of the crucial elements to be monitored and optimized. Energy indicates the proportion of energy used by all IoT devices currently connected to the network. Figure 13 divides the energy consumption into three contributing components to compare the energy dissipation between our architecture BCF, ASCIR, and BSDNSR approaches. Inclusively, our concept beats the BCF, ASCIR, and BSDNSR techniques, which is unable to excuse for the restrictions that IoT devices have on energy usage. On the other hand, by adopting optimal routing methods among the protected collections of IoT devices, It might be efficiently send packets. As a result, the energy efficiency between the proposed model with that of BCF, ASCIR, and BSDNSR methods as showed in Figure 13 uses around 50% less energy. Finally, when compared to the other components, our proposed method uses the best energy efficiency.

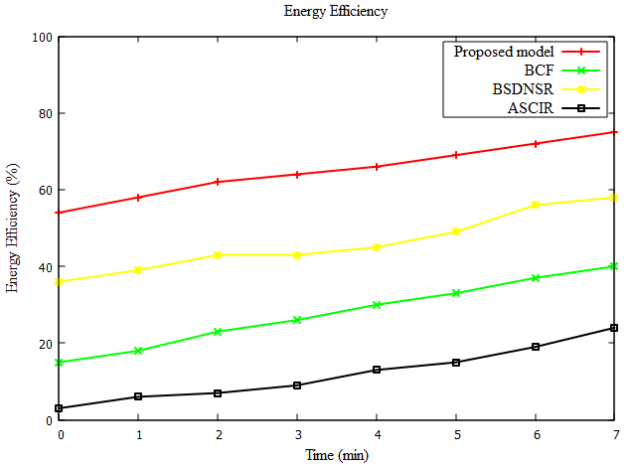

FIGURE 13. Proposed architecture of energy efficiency with BCF, ASCIR, and BSDNSR methods in mobility scenario.

E. END TO END DELAY

The real time systems use IoT applications, it is essential to complete all tasks as quickly as possible. The head of each cluster should therefore be chosen very carefully. To address this problem, an algorithm that quickly chooses the CHs while taking the energy level of the sensors into account is offered by using the Gdist distance measure. When a node is chosen as the CH or when it is connected to another head, it is then tagged. As a result, just one scan for cluster-head selection is performed on each node. It is crucial to look into how a data packet's time in the network affects the end to end delay as it relates to the CH that has been chosen. In fact, the CH must be selected to minimize end-to-end latency during cluster-head selection. The end to end delay as a function of simulation duration is shown in Figure 14 curves when the proposed method, BCF, ASCIR, and BSDNSR methods are performed for seconds. The end to end delay of both strategies converges during the course of the simulation, although the suggested method consistently exhibits lower end-to-end latency than BCF, ASCIR, and BSDNSR methods. As a result, our concept offers adequate performance to choose the CHs correctly and provide effective communication between routing devices.

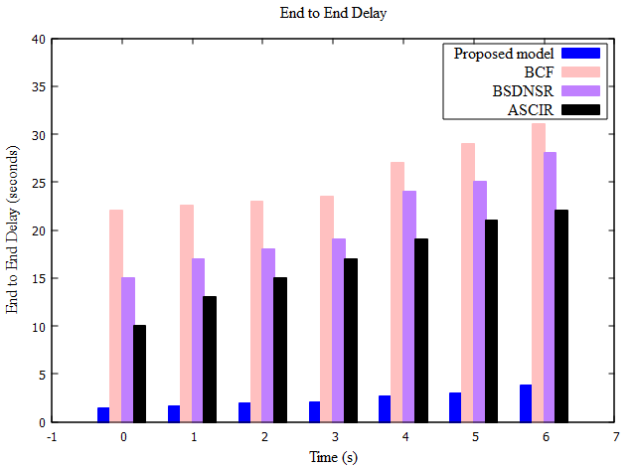

FIGURE 14. Proposed architecture of end-to-end delay with that of BCF, ASCIR, and BSDNSR methods in mobility scenario.

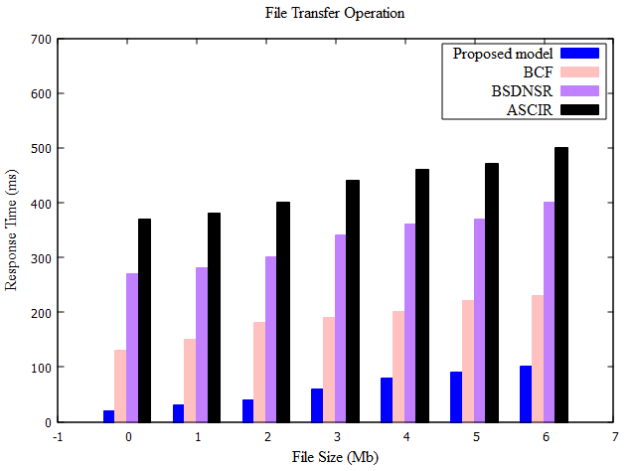

FIGURE 15. File transfer operation of the proposed architecture with that of BCF, ASCIR, and BSDNSR methods in mobility scenario.

F. FILE TRANSFER OPERATION

A TCP-based network, like the Internet, uses the File transfer Protocol (FTP) as a regular network protocol to share data from one computer to another. FTP has distinct control and data interfaces among the user and server and is based on a client-server framework. The local host in an FTP transaction is frequently referred to as the system of the final user. The second computer in an FTP connection is called the remote host, which is frequently a server. Both computers must be networked and configured properly in order to send data using FTP. For clients to utilize these services FTP application is implemented and servers must be configured to execute FTP facilities. A proposed and other method (e.g., BCF, ASCIR, and BSDNSR) of file transfer procedure is shown in Figure 15. Additionally, it displays the file transfer activity accurately in terms of reaction time and file size. The response time increased along with the number of file sizes, and the intended system performed better. The proposed approach can transport huge files more quickly than the current core-based method. As a result, the suggested method enables speedy and safe file transfers.

G. ENERGY CONSUMPTION

Networking components and routing devices often use a lot of energy while transmitting data. Particularly, the device's energy use is inversely associated with the rate of data it transmits (i.e., In other words, the device uses more energy the more bits it transmits). The energy usage may be decreased by utilizing the SDN-Blockchain classifier concept and using the CHs for transmission. The filtering approach is more effective and more efficient. The proposed approach is compared with the BCF, ASCIR, and BSDNSR to determine how much energy each consumes. The results of our 35 second simulation of all methods are depicted in Figure 16. It is clear that our suggested method uses less energy and selects the CHs more effectively than the BCF, ASCIR, and BSDNSR methods. The suggested approach also has a better efficiency in energy usage over longer simulation times, despite the fact that all algorithms have comparable energy-utilization profiles.

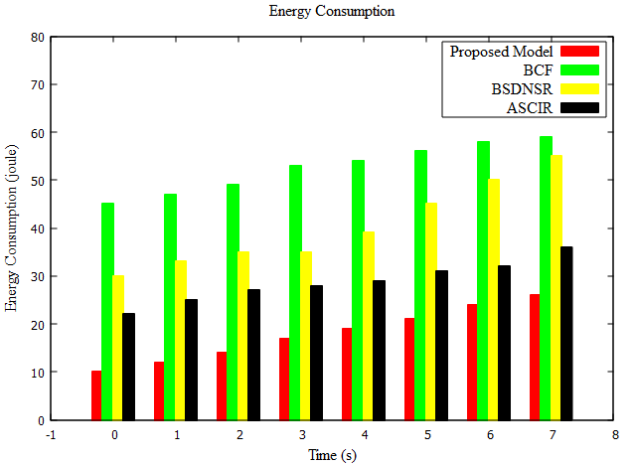

FIGURE 16. Proposed architecture of energy consumption with BCF, ASCIR, and BSDNSR methods in mobility scenario.

H. CPU UTILIZATION

The study of the use of the CPU for flooding attacks in the environment while various applications are running continuously is shown in Figure 17. Furthermore, the learning set was employed to monitor CPU utilization throughout flooding attacks. Moreover, it displays the typical CPU use in different apps built using the proposed strategy during flooding attacks. Then, roughly around point 2.4 value, this violence began; with time, the attack rate also increased. Additionally, it was discovered that proposed model efficiently offers sufficient security against these attacks by carrying on the attack after a predetermined period of time.

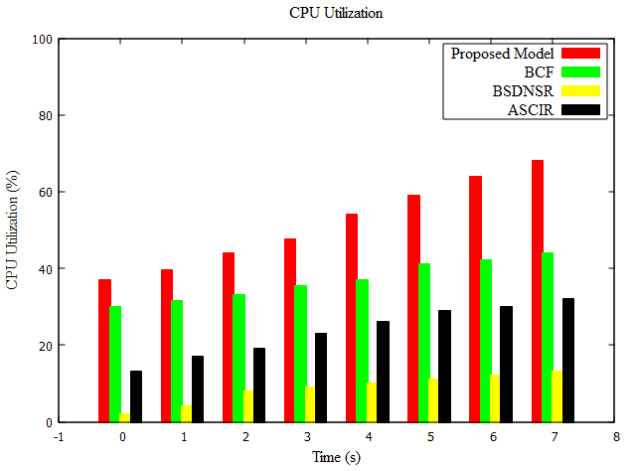

FIGURE 17. Proposed architecture of CPU utilization with BCF, ASCIR, and BSDNSR methods in mobility scenario.

VII. CONCLUSION

Blockchain enable SDN-IoT ecosystems struggle with underdeveloped workflow descriptions in the early stages of development as well as a dearth of resources to effectively install and accomplish this environment. Additionally, only a small amount of earlier studies have looked at and addressed these problems. IoT must deal with challenges as more gadgets nowadays are internet-capable many difficulties with regard to scalability, security, and efficiency to IoT distributed architecture, SDN management is thought to offer the scalability and adaptability required for IoT. SDN controllers may turn into a single point of failure because of the centralized control, making them a prime target for numerous attackers. The study of the convergence between Blockchain and SDN is becoming more prominent due to the growing use of blockchain technology. Consequently, blockchain-based SDN has occurred as a new architecture that enables unreliable parties to communicate with one another in a supportable way without the requirement for a reliable unified authority. Furthermore, MAC flooding attacks might make blockchain nodes incapable of sending or receiving any block information, making blockchain-based SDN insecure. Designing proper security methods is crucial in order to monitor and manage the traffic. Because of this problem, SDN-Blockchain Classifier is created in this work, a trust-based security instrument for blockchain based SDN, by reducing hostile traffic and safeguarding blockchain nodes through traffic fusion and aggregation. Our approach SDN-Blockchain Classifier outperform than assumed baseline on both energy use and end to end latency, according to the experimental assessment (whose implementation scenario is also provided). Overall, compared to a traditional Blockchain, the SDN Blockchain-Based IoT framework achieves greater performance (in terms of average throughput, response time, packet loss of crossing domain path, energy efficiency, end-to-end delay, file transfer operation, energy consumption, and CPU utilization). The processing times displayed by the SDN controllers are also appropriate when compared to the file transfer operation in transactions made on the Ethereum Blockchain. In future, It will be planned to develop the architectures functionality and implement the suggested architecture in a large-scale, real-world situation in the future. In order to assess the responsiveness and flexibility of our SDN-Blockchain Classifier algorithm to environmental changes, it will be also used in a mobility scenario.

REFERENCES

[1] A. Rahman, M. K. Nasir, Z. Rahman, A. Mosavi, and B. Minaei-Bidgoli, "DistBlockBuilding: A distributed blockchain-based SDN-IoT network for smart building management," *IEEE Access*, vol. 8, pp. 140008–140018, 2020.
[2] T. Alharbi, "Deployment of blockchain technology in software defined networks: A survey," *IEEE Access*, vol. 8, pp. 9146–9156, 2020.
[3] Z. A. E. Houda, A. S. Hafid, and L. Khoukhi, "Cochain-SC: An intra-and inter-domain DDoS mitigation scheme based on blockchain using SDN and smart contract," *IEEE Access*, vol. 7, pp. 98893–98907, 2019.
[4] H. Aldabbas and R. Amin, "A novel mechanism to handle address spoofing attacks in SDN based IoT," *Cluster Comput.*, vol. 24, no. 4, pp. 3011–3026, Dec. 2021.
[5] H. Touqeer, S. Zaman, R. Amin, M. Hussain, F. Al-Turjman, and M. Bilal, "Smart home security: Challenges, issues and solutions at different IoT layers," *J. Supercomput.*, vol. 77, no. 12, pp. 14053–14089, Dec. 2021.
[6] O. Flauzac, C. Gonzalez, A. Hachani, and F. Nolot, "SDN based architecture for IoT and improvement of the security," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2015, pp. 688–693.
[7] M. J. Islam, A. Rahman, S. Kabir, M. R. Karim, U. K. Acharjee, M. K. Nasir, S. S. Band, M. Sookhak, and S. Wu, "Blockchain-SDN-based energy-aware and distributed secure architecture for IoT in smart cities," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3850–3864, Mar. 2022.
[8] N. Ravi and S. M. Shalinie, "Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3559–3570, Apr. 2020.
[9] R. Kirichek, A. Vladyko, M. Zakharov, and A. Koucheryavy, "Model networks for Internet of Things and SDN," in *Proc. 18th Int. Conf. Adv. Commun. Technol. (ICACT)*, Jan. 2016, pp. 76–79.
[10] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 812–837, 1st Quart., 2019.
[11] K. S. Sahoo, B. Sahoo, and A. Panda, "A secured SDN framework for IoT," in *Proc. Int. Conf. Man Mach. Interfacing (MAMI)*, Dec. 2015, pp. 1–4.
[12] A. Desai, K. S. Nagegowda, and T. Ninikrishna, "A framework for integrating IoT and SDN using proposed OF-enabled management device," in *Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT)*, Mar. 2016, pp. 1–4.
[13] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, and T. Shah, "Machine learning techniques for spam detection in email and IoT platforms: Analysis and research challenges," *Secur. Commun. Netw.*, vol. 2022, pp. 1–19, Feb. 2022.
[14] J. Li, E. Altman, and C. Touati, "A general SDN-based IoT framework with NVF implementation," *ZTE Commun.*, vol. 13, no. 3, pp. 42–45, 2015.
[15] S. El Jaouhari, A. Bouabdallah, and A. A. Corici, "SDN-based security management of multiple WoT smart spaces," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 10, pp. 9081–9096, Oct. 2021.
[16] L. Xie, Y. Ding, H. Yang, and X. Wang, "Blockchain-based secure and trustworthy Internet of Things in SDN-enabled 5G-VANETs," *IEEE Access*, vol. 7, pp. 56656–56666, 2019.
[17] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "IoT survey: An SDN and fog computing perspective," *Comput. Netw.*, vol. 143, pp. 221–246, Oct. 2018.
[18] C. Vandana, "Security improvement in IoT based on software defined networking (SDN)," *Int. J. Sci., Eng. Technol. Res.*, vol. 5, no. 1, pp. 2327–4662, 2016.
[19] R. Amin, E. Rojas, A. Aqdus, S. Ramzan, D. Casillas-Perez, and J. M. Arco, "A survey on machine learning techniques for routing optimization in SDN," *IEEE Access*, vol. 9, pp. 104582–104611, 2021.
[20] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019.

[21] N. Mazhar, R. Salleh, M. Zeeshan, M. M. Hameed, and N. Khan, "R-IDPS: Real time SDN based IDPS system for IoT security," in *Proc. IEEE 18th Int. Conf. Smart Communities, Improving Quality Life Using ICT, IoT AI (HONET)*, Oct. 2021, pp. 71–76.
[22] P. P. Ray and N. Kumar, "SDN/NFV architectures for edge-cloud oriented IoT: A systematic review," *Comput. Commun.*, vol. 169, pp. 129–153, Mar. 2021.
[23] S. Khorsandroo, A. G. Sánchez, A. S. Tosun, J. Arco, and R. Doriguzzi-Corin, "Hybrid SDN evolution: A comprehensive survey of the state-of-the-art," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 107981.
[24] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proença, "Adversarial deep learning approach detection and defense against DDoS attacks in SDN environments," *Future Gener. Comput. Syst.*, vol. 125, pp. 156–167, Dec. 2021.
[25] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "DistBlockNet: A distributed blockchains-based secure SDN architecture for IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 78–85, Sep. 2017.
[26] S. Ahmad and A. H. Mir, "Scalability, consistency, reliability and security in SDN controllers: A survey of diverse SDN controllers," *J. Netw. Syst. Manage.*, vol. 29, no. 1, pp. 1–59, Jan. 2021.
[27] M. Belotti, N. Bozic, G. Pujolle, and S. Secci, "A vademecum on blockchain technologies: When, which, and how," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3796–3838, 4th Quart., 2019.
[28] S. Xu, X. Chen, and Y. He, "EVchain: An anonymous blockchain-based system for charging-connected electric vehicles," *Tsinghua Sci. Technol.*, vol. 26, no. 6, pp. 845–856, Dec. 2021.
[29] C. Xu, Y. Qu, T. H. Luan, P. W. Eklund, Y. Xiang, and L. Gao, "A lightweight and attack-proof bidirectional blockchain paradigm for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4371–4384, Mar. 2022.
[30] T. Takenaka, Y. Yamamoto, K. Fukuda, A. Kimura, and K. Ueda, "Enhancing products and services using smart appliance networks," *CIRP Ann.*, vol. 65, no. 1, pp. 397–400, 2016.
[31] Y. Huang, J. Tan, and Y.-C. Liang, "Wireless big data: Transforming heterogeneous networks to smart networks," *J. Commun. Inf. Netw.*, vol. 2, no. 1, pp. 19–32, Mar. 2017.
[32] S. A. A. Kazmi, M. K. Shahzad, A. Z. Khan, and D. R. Shin, "Smart distribution networks: A review of modern distribution concepts from a planning perspective," *Energies*, vol. 10, no. 4, p. 501, 2017.
[33] R. Etzioni, N. Urban, S. Ramsey, M. McIntosh, S. Schwartz, B. Reid, J. Radich, G. Anderson, and L. Hartwell, "The case for early detection," *Nature Rev. Cancer*, vol. 3, no. 4, pp. 243–252, 2003.
[34] M. Zhang, F. Eliassen, A. Taherkordi, H.-A. Jacobsen, H.-M. Chung, and Y. Zhang, "Demand–response games for peer-to-peer energy trading with the hyperledger blockchain," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 19–31, Jan. 2022.
[35] Z. Lv, Y. Li, H. Feng, and H. Lv, "Deep learning for security in digital twins of cooperative intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16666–16675, Sep. 2022.
[36] T. E. C. Ainou, S. Ayad, and L. S. Terrissa, "A survey on SDN based energy-efficiency approaches in IoT: Systematic? Survey on energy conservation methods in IoT," in *Proc. 4th Int. Conf. Netw., Inf. Syst. & Secur.*, 2021, pp. 1–7.
[37] T. Yang, L. Kong, N. Zhao, and R. Sun, "Efficient energy and delay tradeoff for vessel communications in SDN based maritime wireless networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3800–3812, Jun. 2021.
[38] M. Baza, A. Sherif, M. M. E. A. Mahmoud, S. Bakiras, W. Alasmary, M. Abdallah, and X. Lin, "Privacy-preserving blockchain-based energy trading schemes for electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9369–9384, Sep. 2021.
[39] S. Aggarwal and N. Kumar, "A consortium blockchain-based energy trading for demand response management in vehicle-to-grid," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9480–9494, Sep. 2021.
[40] R. Khalid, M. W. Malik, T. A. Alghamdi, and N. Javaid, "A consortium blockchain based energy trading scheme for electric vehicles in smart cities," *J. Inf. Secur. Appl.*, vol. 63, Dec. 2021, Art. no. 102998.
[41] S. Hebal, D. Mechta, S. Harous, and M. Dhriyyef, "Hybrid energy routing approach for energy internet," *Energies*, vol. 14, no. 9, p. 2579, Apr. 2021.
[42] B. Finnah and J. Gönsch, "Optimizing trading decisions of wind power plants with hybrid energy storage systems using backwards approximate dynamic programming," *Int. J. Prod. Econ.*, vol. 238, Aug. 2021, Art. no. 108155.
[43] X. Wang, Y. Liu, J. Zhao, C. Liu, J. Liu, and J. Yan, "Surrogate model enabled deep reinforcement learning for hybrid energy community operation," *Appl. Energy*, vol. 289, May 2021, Art. no. 116722.

MOHAMMED A. AL GHAMDI received the bachelor's degree (Hons.) in computer science from King Abdul Aziz University, Jeddah, Saudi Arabia, the master's degree (Hons.) in internet software systems from Birmingham University, Birmingham, U.K., in 2007, and the Ph.D. degree in computer science from The University of Warwick, U.K. Since 2012, he has been with the Department of Computer Science, Umm Al Qura University, Mecca, Saudi Arabia, as an Assistant Professor, and then an Associate Professor. He has authored over 50 papers in international conferences and journals, such as IEEE Systems Journal, IEEE Transactions on Engineering Management, IEEE Access, *Computers, Materials & Continua* (CMC), the IEEE International Conference on Scalable Computing and Communications, and the International Conference on Cloud Computing and Services Science. His research interests include machine learning, data analysis, AI, cloud computer, and cybersecurity. He is the Founder of the Scientific Chair of Data and Artificial Intelligence at Umm Al-Qura University.