IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 January 2024, accepted 31 January 2024, date of publication 6 February 2024, date of current version 15 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3362889

SURVEY

Investigating and Analyzing Simulation Tools of Wireless Sensor Networks: A Comprehensive Survey

GHAIHAB HASSAN ADDAY[1,2], **SHAMALA K. SUBRAMANIAM**[1], **(Member, IEEE), ZURIATI AHMAD ZUKARNAIN**[1], **(Member, IEEE), AND NORMALIA SAMIAN**[1], **(Member, IEEE)**

[1]Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Malaysia

[2]Department of Computer Science, College of Computer Science and Information Technology, University of Basrah, Basrah 61004, Iraq

Corresponding authors: Ghaihab Hassan Adday (ghaihab.aday@uobasrah.edu.iq) and Shamala K. Subramaniam (shamala_ks@upm.edu.my)

This work was supported by the Universiti Putra Malaysia Contract Research under Grant 6300375.

ABSTRACT Wireless Sensor Networks (WSNs) serve as the backbone for the Internet of Things (IoT) and smart cities, enabling the gathering of essential data and vital information. The widespread deployment of sensors created new scenarios that presented new domains for various applications. Due to their numerous advantages, WSN's simulators are still gaining considerable attention as the primary method for testing and evaluating new protocols and approaches for WSN. The scientific community has developed various simulators, some tailored explicitly for WSNs and others for general use. In addition, an old but renovated direction has been enormously grown recently, representing the researcher's attention to building its own simulator. However, many researchers struggle to select the most appropriate tool for performance analysis, which requires extensive research into available options. Choosing a suitable simulator to meet the necessary simulation criteria necessitates exhaustive research into the available options. The published studies on WSN's simulators have limitations, such as the limited number of simulators under examination, ignoring the essential functions of the simulators, and inadequate performance criteria for precise comparison. To get beyond these restrictions in previous studies and offer a fresh study that compares the simulation tools comprehensively, an analysis of thirty-three different simulators was conducted based on a new taxonomy. Furthermore, this study examines the advantages and constraints of each simulator regarding many specific academic areas in WSNs. Moreover, the study presented a detailed comparison among the performance analysis tools according to many famous performance metrics.

INDEX TERMS Discrete event simulation, graphical user interface, MATLAB, NS-2, NS-3, OMNeT++, OPNET, performance analysis tools, specific domain simulators, simulators, TOSSIM, wireless sensor networks.

I. INTRODUCTION

The Internet of Things (IoT) and Wireless Sensor Networks (WSNs) are two industrial technologies embedded in the exponentially growing Industry Revolution 4.0 (IR4.0) concept [1]. Even though WSN is an established technology and not a new one, its importance is increasing exponentially due to the introduction of new and enhanced technologies that depend on sensing [2]. WSN has still garnered significant attention as it forms the foundation of IoT architecture [3], allowing for the detection of surroundings, collection of vital information, and transmission of various types of data to the central station [4], [5]. The attention to WSNs will grow dramatically because the number of sensors

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Seo Kim.

worldwide will be estimated to reach one trillion by 2025 [6]. Moreover, WSNs are crucial today in advancing Next-Generation (XG) wireless communications abilities [7]. In addition, WSN is an essential technology, especially for developing real-time applications such as healthcare applications and environmental monitoring [8]. Although WSNs offer unlimited applications, they often encounter significant challenges due to their distinctiveness from other network types, such as wired and cellular networks. Hardware failures, limited resource shortages, and routing problems are some of the WSN's difficulties. Therefore, performance analysis of previous, current, and proposed algorithms at the design and development stages became vital. While implementing new research, numerous scientific researchers perform evaluations and offer improvements to present new ideas and solutions. Any scientific benchmark's numerical results, figures, and charts must be reproducible to confirm the findings and approve the conclusions. The most often employed methods for performance analysis, specifically for WSNs, are analytical modeling, simulation, and actual deployment [9].

Analytical modeling of systems has typically been done with a mathematical model, which seeks to discover analytical solutions for problems and, as a result, enables the prediction of the system's behavior based on a set of parameters and initial conditions [10]. Analytical modeling is unsuited for WSNs due to their intrinsic complexity and diversity of WSNs. Furthermore, simple models could produce undesirable erroneous findings [11]. The mathematical impossibility of creating complete models for WSNs is a fact that is well-recognized. Every single layer of the protocol stack may be complicated enough to discourage efforts at mathematical analysis. Interactions among the several levels of the protocol stack make this complexity even worse [12]. On the other hand, the testbed is another method to evaluate different algorithms and protocols before implementing them for applications in the real world. WSN's deployment involves the placement of hundreds or thousands of nodes in harsh environments, which is a challenging task [13]. Usually, building an accurate testbed for each predetermined scenario is too costly or impossible. Moreover, most measures are not reproducible and take significant time and effort. In addition, the findings of testbed experiments are significantly impacted by the testing environment, which is sometimes quite unpredictable and difficult to manage [14]. However, WSN testbeds are still necessary for further testing before the final implementation, particularly for applications intended to be used in enclosed spaces.

Compared with previously mentioned approaches, using simulators offers a practical option to analyze algorithms and protocols in an environment under controlled conditions. Simulation is the most common, applicable, and successful method when designing, developing, and testing network protocols for WSNs [15]. The goal of simulators is to reproduce faithfully and precisely anticipate the actual world's behavior in various contexts. Simulation tools provide several benefits, including a reduction in cost, scalability, time, and complexity of implementation [16].

In addition, mathematical modeling can be complemented by simulation as a practical method for validating and enhancing the insights derived from mathematical models. Simulations serve as a complementary tool to mathematical modeling, providing a practical means to verify, validate, and enhance the robustness of models by handling complexity, enabling sensitivity analyses, facilitating iterative improvements, and exploring diverse scenarios. However, the models used by simulators are not guaranteed to be correct; consequently, the outcomes of simulations are not guaranteed to be as accurate as experiments [17]. Many simulators are currently available for modeling different types of networks, such as Wireless Body Area Networks (WBANs), Underwater Wireless Sensor Networks (UWSNs), or even wired networks.

Several network simulators have undergone detailed examination in recent years. Nevertheless, prior studies primarily focused on discussing simulation tools in a general manner, lacking clear explanations on how these tools are suitable for specific domains within WSNs. Additionally, some previous investigations concentrated on a limited number of simulators. However, for a comprehensive review, comparing a minimum of 30 simulators is imperative to provide a precise assessment, given the wide range of available options. Moreover, the time-scale scope of many earlier studies is outdated. Numerous previous studies did not explore the newly developed simulation tools and their updated characteristics.

Therefore, considering the earlier points, presenting a fresh study that builds upon previous works is crucial. This research focuses on thirty-three simulators for small and large-scale WSNs. These simulators were selected because of their widespread applications and popularity, comprehensive features, and extensive published research. The aim is to simplify the difficulty for any researcher in choosing an effective tool for performance assessment. Furthermore, the efforts to improve existing simulators or create new ones require a comprehensive understanding of available performance analysis tools. Therefore, this study offers a thorough examination of WSN's simulators and presents the following contributions:

- The study presents a new taxonomy for thirty-three performance analysis tools, including the most up-to-date WSN simulators. This classification is beneficial in finding the most appropriate network simulator for evaluating the performance processes considering the specific performance metrics.
- The study analyses each of the examined simulators based on various standard functions, including the simulators' architecture, usability, scalability, statistics results, portability type, Graphical User Interface (GUI) availability, and the programming language used for this tool.

- The study examines the selected performance analysis tools based on their features, benefits, drawbacks, and capability to deal effectively with some dominant areas, such as routing algorithms, fault tolerance concepts, security issues, and node deployment.
- The study examines the selected performance analysis tools based on the ability to measure the dominant performance metrics such as network lifetime, throughput, routing overhead, energy consumption, latency, battery lifetime, and network coverage.

The remainder of this work is organized as follows: Most previous studies are discussed in Section II. Section III has proposed the classification and detailed analysis regarding the main features of the available WSN simulators. Section IV illustrates the assessment methodology for the available WSN simulators based on many new perspectives. Discussion and evaluation are presented in Section V. Section VI presented the lessons learned and future directions for the study. Lastly, the paper is concluded in Section VII.

II. RELATED WORKS

Many network simulators have been the subject of extensive studies, case studies, and surveys during the last several years. Institute of Electrical and Electronics Engineers (IEEE Xplore), Association for Computing Machinery (ACM), Multidisciplinary Digital Publishing Institute (MDPI), Science Direct, Springer, and other digital libraries were chosen to increase the odds of getting the best search results. The search phase helped to discover relevant scientific articles regarding the subject of WSN's simulators. This investigation began with selecting various surveys on WSN simulation tools, as shown in Tables 1, 2, and 3, which chronologically present the papers' contributions (2009–2023).

[18] presented a comprehensive overview of thirteen simulation tools designed for congestion detection in IoT networks, comparing them based on various characteristics such as platform, coding, and IoT architecture layer. However, the comparison was limited to the simulation of congestion issues in IoT networks, with a selection of only a few simulators.

In [19], a concise survey on simulation software and simulators for WSN was presented, focusing on the simulators' ability to develop routing protocols in WSNs. Unfortunately, this survey neglected other critical academic areas in WSNs and concentrated solely on eleven performance analysis tools while overlooking recent robust tools.

Reference [20] conducted a comparative analysis of three simulation tools for Long Range (LoRa)/Lower Power Wide Area Network (LoRaWAN) simulations. The analysis considered attributes such as CPU utilization and memory usage. However, for a comprehensive assessment, it is crucial to compare more than just three simulators due to the myriad options available today.

Reference [16] covered twenty-three general network simulators, considering various aspects but lacking precise details on simulation issues in WSNs. Additionally, some simulators evaluated in the study may not be suitable for WSNs due to their specific characteristics or functions. Reference [21] provided a general summary and comparison of communication mechanisms based on Wake-up Receivers for popular simulators. However, the study focused on the comparison of only five simulators, specifically regarding communication mechanisms based on MAC protocols.

Reference [22] presented comparison tables and information on designs for seven performance analysis tools. However, it focused solely on evaluating underwater modeling tools and testing platforms, overlooking the broader spectrum of WSNs.

Reference [23] suggested using six network simulators/emulators to support wireless network simulation. Nevertheless, the investigation was confined to six software solutions suitable for teaching basic principles of wireless networking in educational settings.

Reference [15] provided a general overview and utilization of available simulator methodologies without performance evaluation and dominant domains. Moreover, the work disregarded custom-built simulators, representing a new direction in the simulation area.

Reference [24] showed a brief explanation and comparative study for only five general simulators for modeling Time-Sensitive Networking (TSN). Features of MATLAB/Simulink were presented in [17], focusing on routing terms and a few clustering protocols in WSNs. However, simulation results regarding only a few clustering protocols in WSNs have been discussed.

Reference [25] offered a comprehensive overview of the NS-3 simulator, emphasizing its characteristics, popularity, and flexibility. However, the study does not fully encompass the wide range of WSN simulation tools.

References [26] and [27] systematically described choosing suitable simulation tools for WSNs and their applications, considering factors like scalability. Nevertheless, the comparison was not exhaustive, primarily describing simulators rather than providing a comprehensive analysis.

In [28], the advantages and constraints of seven simulators were examined. Still, the study did not present a detailed comparison among tools according to famous performance metrics, and the discussion was general, not specific to WSNs.

Reference [29] introduced CupCarbon for WSN simulation, evaluating it only for implementing the Dijkstra algorithm and calculating the best route, comparing it with seven simulator tools.

References [30] and [31] compared simulation tools based on relevant WSN criteria, such as modeling energy consumption and extensibility. However, [30] did not examine simulators' suitability for specific academic areas in WSNs, while [31] focused on understanding ubiquitous sensor networks (USNs).

In [32] and [33], performance analysis and comparison of general simulators were presented based on general specifications, scalability, and radio models. However, the

TABLE 1. Previous surveys and studies on network simulators based on the type and scope of the study.

Reference	Year	Study Type	Scope of Study
[18]	2023	Comprehensive Review Paper	Presented a general description of thirteen simulation tools used for congestion detection and control in IoT networks.
[19]	2023	Survey Paper	Presented a brief survey on simulation software and simulators that have been used particularly for developing routing protocols in WSN.
[20]	2022	Survey and Comparative Paper	Presented a comparative analysis of three simulation tools for the Long Range (LoRa)/Lower Power Wide Area Network (LoRaWAN) simulation.
[16]	2022	Survey Paper	Presented a comparison table and performance evaluation using various metrics for twenty-three simulation tools.
[21]	2021	Comparative Study Paper	Presented a general summary, statistical evaluation, and comparison of communication mechanisms based on Wake-up Receivers for five simulation tools.
[22]	2021	Survey Paper	Presented a summary, a comparison table, and information on several experimental platforms' designs and their restrictions for seven performance analysis tools.
[23]	2021	Survey Paper	Presented an overview and suggested use for six network simulators/emulators regarding the support of wireless networks.
[15]	2021	Book	Presented an overview and utilization of general simulators, methodologies, implementation, and a comparison table for twenty-four simulators.
[24]	2020	Comparative Study Paper	Presented brief explanation, summarization, categorization, and comparative study of five network simulators used in Time-Sensitive Networking (TSN).
[17]	2020	Comparative Study Paper	Presented a categorization, comprehensive study for ten network simulators regarding their limitations and powerful features regarding routing algorithms and clustering protocol.
[25]	2020	Review Paper	Presented a comprehensive overview with a comparison table for all the features of the NS-3 simulator and an emphasis on characteristics, popularity, and its flexibility.
[26]	2019	Case Study Paper	Presented an overview, characteristics, assessment metrics, degrees of measurement and appraisal, and a brief comparison table for six network simulation tools.
[27]	2019	Review Paper	Presented an overview, characteristics, and restrictions on experimental tools, including testbeds, simulators, and emulators, and a comparison table for twenty-four simulators.
[28]	2018	Survey Paper	Presented a comparative study including the benefits, drawbacks, the backend environment, and the supported operating system for seven simulators.
[29]	2018	Evaluation Paper	Presented the evaluation of a performance analysis tool called CupCarbon with a comparison of seven other simulators based on the characteristics regarding the routing terms.
[30]	2017	Comparative Study Paper	Presented a comparative study and a table of comparisons for twenty-two simulators based on their features, energy consumption modeling, modeling mobility, and extensibility.
[31]	2017	Review Paper	Presented an overview, characteristics, and assessment methods for many performance analysis tools based on various specifications, operating systems, requirements, and constraints.
[32]	2017	Survey Paper	Presented a performance analysis and comparison of modeling for three simulators.
[33]	2016	Comparison Paper	Presented a brief analysis and comparison table of seven simulation frameworks supporting specific software packages.
[34]	2016	Comparison Paper	Presented a comparative study, classification, list of standard simulation settings, and table of comparisons for six simulation tools.
[35]	2015	Review Paper	Present detailed analysis and comparison based on the architectural and various features, such as animation availability for thirty-one simulators.
[36]	2014	Review Paper	Presented an overview and an investigation of twenty simulation tools of WSN, including their characteristics, benefits, and drawbacks.
[37]	2013	Comparison Paper	Presented a comprehensive analysis of the history and background of six distinct sensor network simulators, as well as a discussion of the benefits and drawbacks of each simulator.
[38]	2012	Comparison Paper	Presented performance evaluation of four performance analysis tools concerning different parameters.
[39]	2012	Comparison Paper	Presented a discussion of four network simulators' performance, dependability, energy use, and reliability.
[40]	2011	Review Paper	Presented a review and comprehensive investigation of modern features, constraints, and strengths for seven simulators.
[41]	2010	Survey Paper	Presented a survey, brief description, and categorization of twenty network simulators.
[42]	2009	Case Study Paper	Presented a comprehensive discussion of the pros and cons of the six most popular performance analysis tools.

TABLE 2. Previous surveys and studies based on the simulators under investigation and the limitations of the study.

Reference	Year	Simulators under investigation	Limitations
[18]	2023	Iotify, OMNeT++, COOJA, MaMMotH, SimIoT, MobIoTSim, IOTSim, ndnSim, NS-3, Bevywise-IoT, EdgeCloudSim, NetSim, and Ansys-IoT	The discussion was limited to the simulation performance of the congestion issue in IoT networks, neglecting many other critical fields of study within WSNs
[19]	2023	NS-3, NS-2, OMNeT++, COOJA, QualNet, Castalia, MiXiM, NetSim, WSNet, OPNET, and TOSSIM	The study briefly reviewed only eleven simulators regarding the routing issues in WSNs and neglected other significant areas such as fault tolerance, security, and data collection.
[20]	2022	NS-3, LoRaSim and OMNeT++	The study presented a comparative analysis for only three simulation tools by evaluating their performance in terms of CPU utilization, memory usage, the number of collisions, etc. The study disregarded many powerful tools that can be used for WSN simulation purposes, such as NS-2 and OPNET.
[16]	2022	PeerSim, PeerThing, RealPeer, Query Cycle, Neurogrid, GPS, 3LS, P2PSim, Overlay Weaver, PlanetSim, Narses, DHTSim, OverSim, OMNeT ++, P2PRealm, NS-2, NS-3, Optimal-Sim, ProtoPeer, Peerfact-Sim, D-P2P-Sim, OPNET, and NetSquid	The study did not provide precise details on simulators used for different fields of WSNs. Moreover, some simulators evaluated in the study may not be suitable for WSNs due to their specific characteristics or functions.
[21]	2021	MATLAB, NS-2, NS-3, NetSim, and OMNeT++	The study discussed the comparison among only five simulators regarding the communication mechanisms based only on the X-MAC and B-MAC protocols.
[22]	2021	NS-3, NS-2, OMNeT++, OPNET, QualNet, J-Sim, and NetSim	The study only concentrated on evaluating underwater modeling tools and experimental platforms, disregarding the broader spectrum of WSNs.
[23]	2021	OMNeT++, Packet Tracer, NS-3, Mininet, CORE, and Komondor	The study investigated a limited selection of six simulation software suitable for teaching the basic principles of wireless networking in educational settings such as classrooms or laboratories.
[15]	2021	NS-2, NS-3, GloMoSim, OPNET, OMNeT++, TOSSIM, ATEMU, Avrora, EmStar, SensorSim, NRL SensorSim, J-Sim, PROWLER /J PROWLER, SENS, SENSE, Shawn, SenSim, PAWiS, MSPsim, Castalia, MiXiM, NesCT, Sunshine, and NetTopo	The study presented a general overview and utilization of available simulator methodologies without performance evaluation and dominant domains in WSNs. Moreover, the book transgressed the custom-built simulators by individuals, representing a new direction.
[24]	2020	OMNeT++, OPNET, NS-2, NS-3, and MATLAB	The study presented only five primary network simulation methods and their requirements for Time-Sensitive Networking (TSN).
[17]	2020	NS-3, NS-2, MATLAB / Simulink, OMNeT++, QualNet, SENSE, OPNET, TOSSIM, NetSim, and GloMoSim	The study highlighted prominent features of MATLAB/Simulink. However, simulation results have only been discussed in routing terms and a few clustering protocols in WSNs.
[25]	2020	NS-3	The study discussed only one simulation tool, NS-3, disregarding many other available options.
[26]	2019	NS-2, NS-3, OMNeT++, GNS3, MATLAB, and OPNET	The study offers a systematic guide for choosing suitable simulation tools for WSNs and their corresponding applications. Nevertheless, the study discussed only six simulators. These six simulators offer insights but do not fully encompass the wide range of WSN simulation tools available to the academic community.
[27]	2019	NS-2, NS-3, OMNeT++, Avrora, OPNET, J-Sim, Qualnet, GloMoSim, PROWLER, Emsim, COOJA, MANNASIM, Castalia, TOSSIM, EmStar, ATEMU, VMNET, FreeMote, MSPSim, WSNTB, INDRIYA, MoteLab, TutorNet, and RealNet	The study provided a descriptive exploration of many specific tools rather than an in-depth comparative evaluation encompassing multiple facets for a comprehensive understanding. The comparison was not exhaustive, focusing primarily on describing the twenty-four mentioned simulators rather than providing a comprehensive analysis across various critical factors.
[28]	2018	NS-2, NS-3, OMNeT++, OPNET, NetSim, Real, and QualNet	The study examined the advantages and constraints of seven simulators. However, the study did not present a detailed comparison among the tools according to many famous performance metrics. Moreover, the study discussed the mentioned simulators from a general perspective, not specifically for WSNs.
[29]	2018	NS-2, NS-3, OMNeT++, Riverbed, NC- TUNS, TOSSIM, and CupCarbon	The study presented a new simulator called CupCarbon for the WSNs simulation. However, the evaluation has been made regarding only the implementation of the Dijkstra algorithm and the calculation for the best route to the destination. In addition, the comparison has been done only regarding seven simulator tools.

TABLE 2. ***(Continued.)*** **Previous surveys and studies based on the simulators under investigation and the limitations of the study.**

[30]	2017	ATEMU, Avrora, Castalia, COOJA, Sidh, Dingo, EmStar, SENS, GloMoSim, J-Sim, JiST/SWANS, OMNeT++, NS-2, NS-3, SENSE, SensorSim, Shawn, ShoX, WsnSimPy, TOSSF, TOSSIM, and VisualSense	The study compared twenty-two simulation tools based on specific criteria in WSNs, such as modeling energy consumption and the extensibility of the simulator. However, the study did not address custom-built simulators. Additionally, some simulators investigated in the study are no longer supported by their developers.
[31]	2017	Ptolemy, NS-2, NS-3, OMNET++, Atarraya, Cell-DEVS, ABMQ, MASON, RepastSNS, NetLogo, SXCS, UbiWise, UbikSim, TATUS, UWSim, SUNSET, SUNRISE, DESERT, RECORDS, Aqua-Net, SeaLinx, Aqua-Net Mate, Aqua-Lab, Aqua-Sim, Aqua-Tune, Aqua- GloMoSim, PROWLER /JPROWLER, Shawn, WSNimPy, SensorSim, OLIMPO, CupCarbon, TimSim, and JSensor	The comparison analysis made by the study is based only on the perspective of the understanding of ubiquitous sensor networks (USNs).
[32]	2017	OPNET, OMNeT++, and NS-2	The study compared simulation modeling for only three simulators based on general specification, scalability, and radio model. Moreover, the study disregarded many powerful simulators such as NS-3, TOSSIM, and Castalia.
[33]	2016	NS-2, PROWLER, OMNeT++, Atarraya, PiccSIM, TrueTime, and MATLAB/Simulink	The study compared only seven simulation frameworks and determined a suitable environment supporting specific software packages such as communication and data collection protocols.
[34]	2016	Castalia, MiXiM, WSNet, PASES, TOSSIM, and COOJA	The study compared only seven simulators based on two simulation scenarios designed and equally implemented in each tool to assess them. Moreover, the study neglected some popular simulation tools such as NS-2, NS-3, and OMNeT++.
[35]	2015	NS-2, NS-3, OMNeT++, J-Sim, MANNASIM, SensorSim, NRLSensorSim, NCTUns, SSFnet, GloMoSim, QualNet, sQualNet, OPNET, SENSE, DRMSim, NetSim, UWSim, VisualSense, Ptolemy, SENS, Shawn, SIDnet-SWANS, WSim/WorldSens/WSNet, WSN Localization Simulator, NetTopo, SIDH, PROWLER, MATLAB, PiccSIM, and LabVIEW	The study presented a detailed analysis and comparison based on the architectures of thirty-one simulators. However, the comparison has been made based on limited features like animation availability and the simulator components.
[36]	2014	NS-2, NS-3, OMNeT++, GNS3, Ethereal, OPNET, QualNet/ GloMoSim, J-Sim, JiST/SWANS, TOSSIM, Ptolemy, Castalia, EmStar, ATEMU, SENSE, SENS, JPROWLER, Avrora, COOJA, and Shawn	The study presented a brief overview of a set of representative twenty WSN simulators regarding only general characteristics such as the platform operating on and the web address.
[37]	2013	NS-2, NS-3, TOSSIM, OMNeT++, Castalia, and QualNet	The study disregarded various simulation tools designed especially for the WSNs. Moreover, the study neglected to compare the six simulators based on many specific academic areas in WSNs and dominated performance metrics.
[38]	2012	NS-2, NS-3, OMNeT++, and GloMoSim	The study discussed the performance evaluation of only four simulator tools. However, the evaluation process was based on a few criteria, such as processor utilization, memory usage, and scalability. Moreover, the simulators under investigation have been compared regarding the MANET routing scenario and neglected other essential simulation scenarios in WSNs
[39]	2012	NS-2, PowerTOSSIM, MATSNL, and MATSNL	The time scale for the study is old, and many new releases have been issued for the investigated simulators.
[40]	2011	NS-2, OMNeT++, GloMoSim, OPNET, SENSE, GTSNetS, and TOSSIM	The study's findings are obsolete due to its timeframe being from a previous period, and multiple updates have been released since. Moreover, many new simulators issued later, such as NS-3, were not discussed.
[41]	2010	SensorSim, Nsrlsensorsim, Castalia, VisualSense, Viptos, Sidh, PROWLER, SENS, TOSSIM, ATEMU, Avrora, SENSE, EmStar, MoteLab, SensorScope, GNOMES, Emulab, SignetLab, SENSONET, and ORBIT	The study's timeframe is antiquated, and many of the simulation tools under investigation within the study have not been used recently.
[42]	2009	NS-2, GloMoSim, QualNet, J-Sim, OMNeT++, and OPNET	The study's findings are outdated due to its timeframe. Moreover, the comparison has been made regarding only six simulators according to a few criteria, such as modules available and graphical support.

comparison excluded new simulators like NS-3 and specific simulation tools for WSNs, such as TOSSIM.

Reference [34] compared seven simulators based on two simulation scenarios, assessing them for specific

TABLE 3. Previous surveys and studies on network simulators based on the target domain and the number of simulators within the study.

Reference	Target Domain			Simulators			
	General	WSN or IoT	Particular Branch of the WSN	Less than 10	10-20	21-30	More than 30
[18]		✓			✓		
[19]			✓		✓		
[20]		✓		✓			
[16]	✓					✓	
[21]		✓		✓			
[22]			✓	✓			
[23]	✓			✓			
[15]		✓				✓	
[24]		✓		✓			
[17]		✓			✓		
[25]	✓			✓			
[26]	✓			✓			
[27]		✓				✓	
[28]		✓		✓			
[29]		✓		✓			
[30]		✓				✓	
[31]			✓				✓
[32]		✓		✓			
[33]		✓		✓			
[34]		✓		✓			
[35]		✓					✓
[36]			✓			✓	
[37]		✓		✓			
[38]		✓		✓			
[39]		✓		✓			
[40]			✓	✓			
[41]		✓					✓
[42]	✓			✓			

performance metrics like throughput, packet loss, and packet delivery ratio. Notably, powerful tools like NS-2 and NS-3 were not included.

A detailed analysis and comparison of thirty-one simulators based on their architecture was presented in [35]. However, the comparison was limited to features like animation availability and the simulator components.

References [36] and [37] provided a brief overview of WSN simulators, focusing on general characteristics like the platform and website address. Yet, the study overlooked simulation tools designed specifically for WSNs and failed to compare them based on their suitability for specific academic areas and dominant performance metrics.

Reference [38] discussed the performance evaluation of four popular simulator tools, considering characteristics such as CPU utilization, memory usage, computational time, and scalability. The simulation scenario involved routing protocols for MANET.

References [39], [40], [41], and [42] discussed characteristics such as speed performance, dependability, energy use, and radio models of various network simulators. However, these studies have an outdated time scale, and many new releases, including NS-3, were not discussed. Additionally, some simulation tools investigated in these studies are rarely used today for WSN simulation.

Tables 1, 2, and 3 show that a substantial number of surveys have been done in this area, providing distinct classifications. However, analyzing these prior surveys in detail has presented several open issues, representing the undoubted need for a new study to address the shortcomings of previous works. The shortcomings of the earlier surveys can be summarized as follows.

- Most prior studies discussed the simulation tools from a general perspective, as shown in Tables 2 and 3. Only limited studies have dealt with the network simulators explicitly designed for the WSNs. It is essential to comprehensively examine the simulation tools that present the WSN environments, concepts, and models, such as SensorSim, TOSSIM, and Castalia. Moreover, it is crucial to introduce a new study investigating the simulation tools regarding critical parameters that reflect new directions in WSNs.

- Powerful simulators like NS-3, OPNET, and OMNeT++ might not have been discussed comprehensively and clearly in previous studies. With the increasing requirements for WSNs and current advanced innovation for IR 4.0 and IoT, many previous simulators have derivations that are being developed regularly. Therefore, these sub-simulators, such as Castalia and SENSE, must be examined separately.
- Many previous studies examined a few performance analysis tools, as shown in Table 3. A complete assessment must compare at least 30 simulators to get an accurate evaluation due to the availability of various simulators nowadays. Thus, this comprehensive study will help guide scientific researchers toward an easy and precise selection of an appropriate tool that can cover the direction of the required experiment.
- The time scale of some previous works under examination is old. Many earlier studies did not examine the newly developed simulators, as shown in Table 2. Furthermore, some previous studies have not examined some unsupported simulation tools recently. New WSN simulators are constantly being developed. Comparing simulation tools that are not supported at the present moment with current simulators is necessary. This will provide reasonable indications about the strengths and limitations of the performance analysis tools and the required new trends regarding the related areas.
- Many previous surveys neglected essential characteristics to discuss when evaluating the simulation tools. Therefore, it is crucial to discuss these critical characteristics for a clear and comprehensive comparison. Many vital features in different simulators will be addressed to fulfill the aim of this work, such as the scalability issue, supporting documents, and the availability of GUI for visualization requirements.
- Many earlier studies examined several simulation tools for WSNs. However, these studies did not clearly explain the tool's suitability regarding specific domains within WSNs, such as fault tolerance, routing, security, and data aggregation. This study will provide detailed information about the different WSN domains that can and cannot be simulated effectively using a specific performance analysis tool.
- Many prior studies did not present the most critical performance metrics that can or cannot be measured accurately using specific performance analysis tools. Therefore, this work shows various performance metrics regarding WSNs that different simulation tools can measure according to currently available research works. Thus, this will help new academic researchers easily select suitable performance analysis tools and have clear insight into similar methodologies.

Based on what was mentioned above regarding the unaddressed shortcomings within previous studies, the presented research comprehensively discusses a wide range of WSN simulators. This work clarifies for new and current academic researchers the pros, cons, architecture, and functionality of every simulator under investigation. In addition, this study illustrates many comparisons among different simulation tools based on new perspectives.

III. WSN SIMULATION TOOLS

The simulation of WSN's behavior is undoubtedly one of the most innovative approaches for testing and validating various algorithms and protocols [43]. While investigating the properties of a new routing protocol, for example, or a new fault tolerance algorithm, the researcher would often utilize a network simulation tool. It is now a standard practice for creating new communication systems and network protocols. WSN simulators enable modeling an arbitrary network by describing the communication channels and the behavior of the sensor nodes [44].

Given that the network topology is nothing more than a collection of simulation parameters, it will eventually be possible to investigate, for example, the routing behavior in various topologies.

On the other hand, when modeling WSNs, the way simulation tools and hardware components interact might differ based on the specific simulation tool and its capabilities. However, the way that simulation tools may interface with hardware components within the framework of WSNs can be summarized by the following stages:

- **Abstraction Stage** The majority of simulation tools, such as OMNeT++ or NS-2, work at an advanced level of abstraction that primarily deals with the modeling of network behavior, communication protocols, and sensor node interactions. Many simulators can skip over specific hardware component information to concentrate on simulations at the network level.
- **Fundamental Hardware Characteristics Stage** Simulation tools can include fundamental hardware properties in sensor nodes, like energy consumption models, transmission range, sensor capabilities, and processing capabilities.
- **Customization Stage:**
  Simulation tools enable researchers to develop customized hardware models. This entails delineating hardware attributes, such as particular processors, energy models, and memory limitations, inside the simulated sensor nodes, enabling a more comprehensive simulation at the hardware level.
- **Collaboration with Additional Tools Stage**
  Some simulation tools can interact with other software or platforms specializing in hardware simulation. This incorporation allows researchers to include specific

hardware functionality or real-world properties in the simulations.

The Discrete Event Simulation (DES) paradigm is the foundation for most network simulations. The concept of DES involves organizing a complex system's behavior into a structured sequence of events. Each event happens at a particular time and represents a specific system status change [45].

The simulated nodes set and arrange the events based on the DES concept. This structured approach enables researchers and engineers to analyze and comprehend how a system behaves over time, allowing them to make more informed decisions and optimizations. For instance, when a packet is transferred from one node to another.

The simulator manages an event queue organized according to the planned event execution time. Fig. 1 illustrates a taxonomy for the most often employed methods for performance analysis in WSNs: analytical modeling, simulation, and actual deployment. In addition, this study proposed a new taxonomy layer that will classify the popular WSN simulators (thirty-three simulators) into three classes: general domain simulators, specific domain simulators (WSN simulators), and custom build simulators. As shown in Fig. 1, each primary category contains several simulation tools that share the main concepts. The general domain simulator class includes 14 different simulators, while the specific domain simulator class, specially designed for WSNs, includes 19 other simulators. Lastly, individuals developed and presented the custom-built simulation tools for particular tasks in WSNs. The three main categories will be explained precisely in the following subsections, including each simulator's work under each category.

A. GENERAL DOMAIN SIMULATORS

General domain simulators offer a wide range of network simulation modules with a versatile design that can be applied to various network types. These simulators serve as valuable tools for researchers and developers working on wired and wireless networks, as they provide a common framework that can be easily adapted for different scenarios. The advantage of general domain simulators lies in their flexibility. End users can modify essential components of the network or application, allowing for customization and experimentation. This adaptability is particularly useful for exploring different network configurations and protocols in network simulation. These simulators' clear hierarchy and modular nature make them highly extendable and maintainable.

New components and applications can be seamlessly integrated, enabling the simulators to evolve with advancements in network technology. For example, although OMNeT++ may not have been explicitly designed for WSNs, it can effectively simulate WSN-related issues [46]. Moreover, researchers can leverage the existing framework or incorporate specialized frameworks; for instance, Castalia can be used to tailor the OMNeT ++ simulator for WSN simulation purposes. One of the primary benefits of using a general network simulator is the availability of standard modules for network simulations. These simulators, such as NS-3, NS-2, OMNeT++, OPNET, and J-Sim, provide a rich set of prebuilt modules that can be readily used, reducing the need for users to develop complex components from scratch [16]. This saves time and effort, especially for common simulation scenarios. However, it is essential to evaluate the characteristics and limitations of these simulators. The following subsections briefly describe the main characteristics, pros, and cons of the network simulators in this category. Additional analysis and evaluation based on three different concepts with more features will be presented in Section IV. The examination process will provide a comprehensive understanding of the capabilities and potential constraints when utilizing general domain simulators in WSN simulation.

1) NETWORK SIMULATOR-3 (NS-3)

NS-3 is a free general simulator developed to meet the simulation requirements of contemporary networking research and promote community participation and software validation [25]. C++ is used in NS-3 to implement simulation models.

NS-3 uses Object-oriented Tool Command Language (OTcl) under the Unix environment but without OTcl scripts to administer the simulation. There is also the option of utilizing Python for implementing network simulations in NS-3. This can be employed to complete the simulation components, as illustrated in Fig. 2, which depicts the simulator architecture. Additional features can be summarized below in terms of both pros and cons:

Advantages of NS-3

- NS-3 provides numerous statistical models for wireless channels, mobility, and traffic creation. Additionally, NS-3 may communicate with external systems, such as real-time Long-Term-Evaluation (LTE) testbed, or external libraries, such as Click [47].
- NS-3 consists of several already developed simulation modules such as Wi-Fi, WiMAX, LTE, and point-to-point, which can reasonably handle matters of the public channel [48].
- NS-3 is compatible with Software Defined Networking (SDN).
- NS-3 itself does not have a built-in virtual machine mechanism. However, NS-3 can be used alongside virtualization technologies or platforms that support virtual machines to enhance the simulation of network scenarios within virtualized environments. For example, researchers can use NS-3 alongside virtualization solutions like VMware, VirtualBox, or other cloud-based virtualization platforms that provide virtual machine capabilities [25].
- Linux essentially powers NS-3. Cygwin may be used to run NS-3 on various operating systems, including Linux and Windows. Cygwin is a set of open-source tools that enables Unix or Linux programs to be built and run on Microsoft Windows.

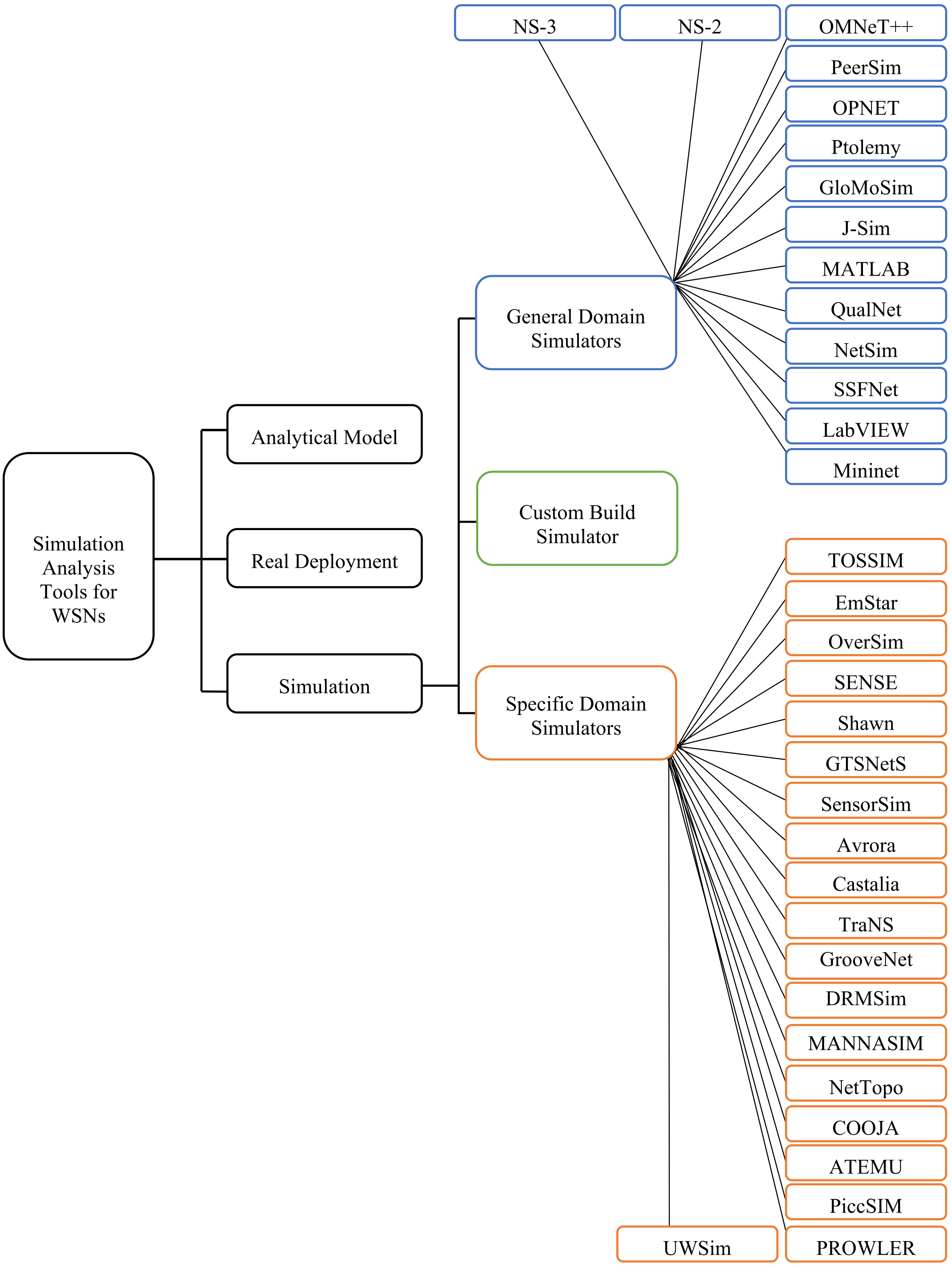

FIGURE 1. Proposed classification of WSN simulation tools.

Disadvantages of NS-3

- The duplicate effort of writing simulation code is prevalent in network simulators. This duplication can be avoided using fast prototyping methodologies, which enable the reuse of simulation code in real prototyping and production environments. Although

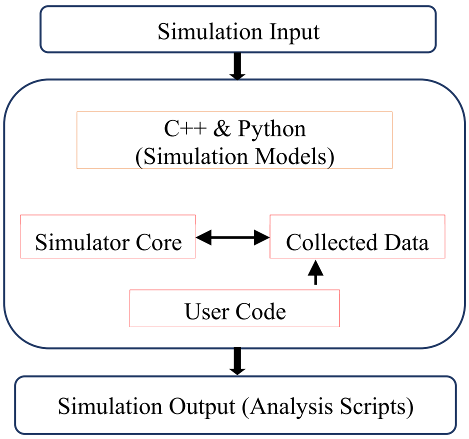

FIGURE 2. The basic architecture of NS-3.

this functionality is already available using NS-3, but there are still limitations regarding the support of real network interfaces and easy configuration of the network settings, such as IP and MAC addresses [49].
- The next-generation mobility management schemes follow the protocol stack of Mobile Internet Protocol version 6 (MIPv6). However, NS-3 does not have any MIPv6 module [50].
- Several Mobile Ad-hoc Network (MANET) routing protocols are available in NS-3. However, it does not include Disruption Tolerant Network (DTN) routing protocols [51].
- NS-3 is primarily designed for discrete-event network simulations, which means it models the behavior of a network over discrete time steps. On the other hand, real-time simulation involves executing a simulation in sync with real-time events. It requires strict timing constraints and the ability to process events and reactions within specific time windows. NS-3 does not provide built-in support for real-time simulation requirements. However, researchers have used external synchronization mechanisms with real-time systems to simulate certain aspects of real-time behavior using NS-3.
- The graphical presentation is not fully complete and is being worked on continuously. Since it is a relatively new simulator that undergoes consistent module updates, NS-3 cannot be used with NS-2 because of incompatibility issues.
- Since energy efficiency is an essential consideration for WSN and IoT applications, one shortcoming of NS-3 is its inability to support the simulation of the energy consumption of Low-Rate Wireless Personal Area Networks (LR-WPAN) [52].
- NS-3 needs continuous system validation, so maintainers must answer the reported bugs and validate the system actively.
- The main drawback of NS-3 is the complexity of learning, and it is described as a time-consuming tool for having complete knowledge about it. Many researchers claim the learning process to utilize NS-3 use is still an open issue.

2) NETWORK SIMULATOR-2 (NS-2)

NS-2 is now one of the famous network simulators used most often in academic and commercial settings. It offers considerable support for the simulation of routing and multicast protocols over wired and wireless networks arranged in a structured or unstructured way, respectively. NS-2 includes Object-oriented Tool Command Language (OTcl) and C++, as shown in Fig. 3.

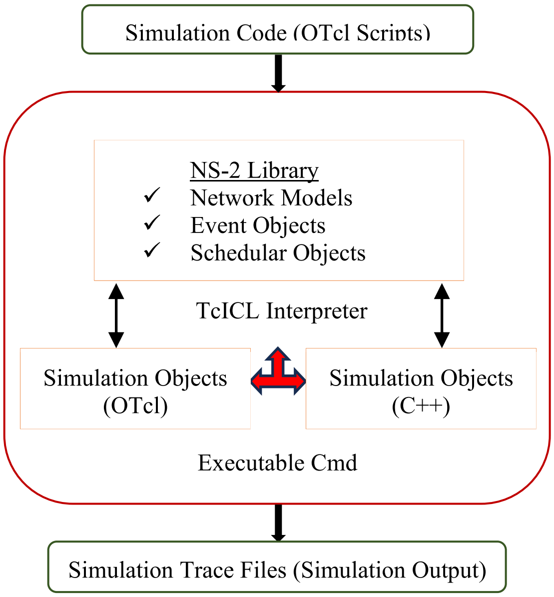

FIGURE 3. The basic architecture of NS-2.

While OTcl scripts handle the tasks of establishing simulators, setting network topologies, constructing network scenarios, and showing simulation results, C++ is mainly used for implementing various protocols and expanding simulation libraries. TclCL is the binding between C++ and OTcl.

Even though NS-2 has been among the most popular WSN simulators for the past ten years, building graphical editors for NS-2 models is almost impossible due to the simulator's architecture. Additional features can be summarized below:

Advantages of NS-2

- It has been shown to simulate as many as 5,000 nodes in a network [53].
- NS-2 is considered the most popular simulation tool that can be used for many fields in WSNs, such as routing, security, fault tolerance, and node localization issues [54], [55].
- NS-2 uses an effective energy model for a straightforward traffic and movement pattern [56].
- NS-2 operates under different operating systems using Cygwin, but Linux is preferable.

- NS-2 supports a variety of wireless Medium Access Control (MAC) protocols, such as IEEE 802.11 (Wi-Fi) and IEEE 802.15.4 (ZigBee).

Disadvantages of NS-2

- NS- does not provide graphical representations of the data produced by simulations. It is necessary to process the raw data using scripting languages.
- Even though NS-2 can simulate more than 5000 sensor nodes, NS-2 suffers from scalability issues and requires high computational resources.
- The main shortcoming of NS-2 is that it is not user-friendly since its interface is text-based; several researchers have voiced their dissatisfaction with the complex learning process of NS-2.

3) OBJECTIVE MODULAR NETWORK TESTBED (OMNET++)

OMNeT++ is a free software general simulator that employs the C++ programming language for simulation model development [57]. To construct their simulator, the OMNeT++ model is a collection of hierarchically layered modules. The module at the highest level is also known as the Network Module, as shown in Fig 4.

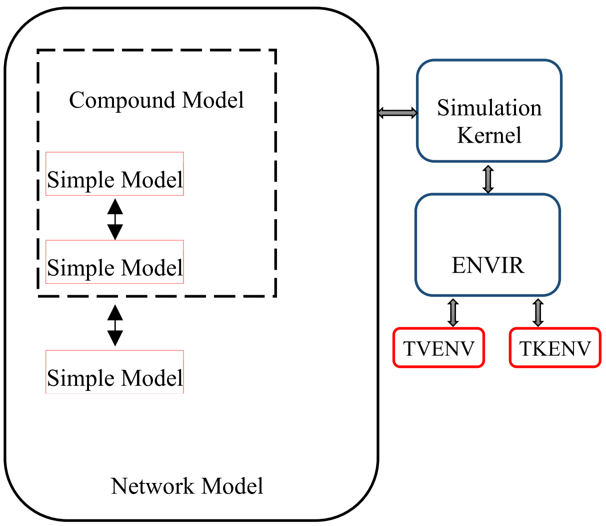

FIGURE 4. The basic architecture of OMNeT++.

This module includes several sub-modules, each of which may contain other sub-modules [58]. At the bottom of the hierarchy are simple modules used to design algorithms and serve as their building blocks. Compound modules consist of a group of basic modules that communicate through messages. OMNeT++'s Network Description Language (NED) is used to create compound modules and configure network simulations. Additional features can be summarized below in terms of both pros and cons:

Advantages of OMNeT++

- OMNeT++ can support DES, which modules communicate through message passing. OMNeT++ also facilitates parallel distributed simulation execution.
- OMNeT++ provides robust GUI support and an embeddable simulation kernel for visualizing user interaction. The interactions between modules are recorded in a log file [59].
- OMNeT++ is scalable for real-time simulation, network emulation, and database integration.
- OMNeT++ is widely used, extensible, and actively maintained by its academic user group, which has also developed extensions for WSN simulation. SensorSim and Castalia are adaptations for the OMNeT++ simulator that can be used for WSNs.
- OMNeT++ supports the best tracing techniques compared to other simulators and offers complete channel controls for WSNs [60].
- OMNeT++ provides many more powerful frameworks to simulate WSNs, such as MiXiM, INET framework, NesT, and PAWiS.

Disadvantages of OMNeT++

- Combining individual models is complex and may result in high-probability bugs.
- OMNeT++ has a steeper learning curve compared to some other simulation tools, especially for those who are not familiar with C++.
- Developing models in OMNeT++ can require advanced programming skills, particularly in C++ or NED.
- Unlike other simulation tools, OMNeT++ alone does not provide an extensive library of pre-built models tailored explicitly for WSNs. This means that developers often need to create their models from scratch or adapt existing models to suit their specific requirements, which can be time-consuming [61].

4) PEER-TO-PEER SIMULATOR (PEERSIM)

PeerSim is a general domain simulator that can be used for large-scale WSN simulation while providing instructional tools for statistical calculations [62]. It provides two simulation models: cycle-based and event-based. Cycle-based is more comprehensive than event-based, with much supporting documentation (user manual and API) [63].

The PeerSim license is open source, and because of its architecture, it is feasible to simulate massive P2P networks thanks to its scalable and dynamic nature. Additional features can be summarized below:

Advantages of PeerSim

- It has high scalability features, and the network density can reach one million nodes.
- PeerSim provides reusable, simple API documentation that may support additional parts per need, especially for cycle-based types.

Disadvantages of PeerSim

- It does not provide debugging facilities and does not have a graphical user interface.
- It offers only a few class packages that support well-known models like random and lattice graphs.
- The most significant drawback is that command base simulator inputs are presented in a text file. Moreover,

the user cannot change the parameters in real-time because there is no GUI [64].
- Does not apply distributed simulation.
- PeerSim is mainly intended for P2P systems and does not have built-in support for WSN simulations.
- PeerSim does not have a comprehensive set of built-in wireless communication models for modeling radio propagation and signal attenuation, which is typical in WSNs [65].

5) OPTIMIZED NETWORK ENGINEERING TOOL (OPNET)

Riverbed Modeler, or OPNET, is a powerful collection of tools for building and testing massive network environments [66]. OPNET employs Object-Oriented Programming (OOP) methodologies to explore WSN's communication and applications flexibly [67]. Fig 5. illustrates the simulator's robust central architecture, making it a flexibly efficient tool for many applications. OPNET covers a variety of network fields, including application performance management, engineering, and research and development. It provides consumers with comparatively strong graphics support. Network topology and entities may be built using the graphical editing interface from the application to the physical layer. Additional features for OPNET can be summarized below:

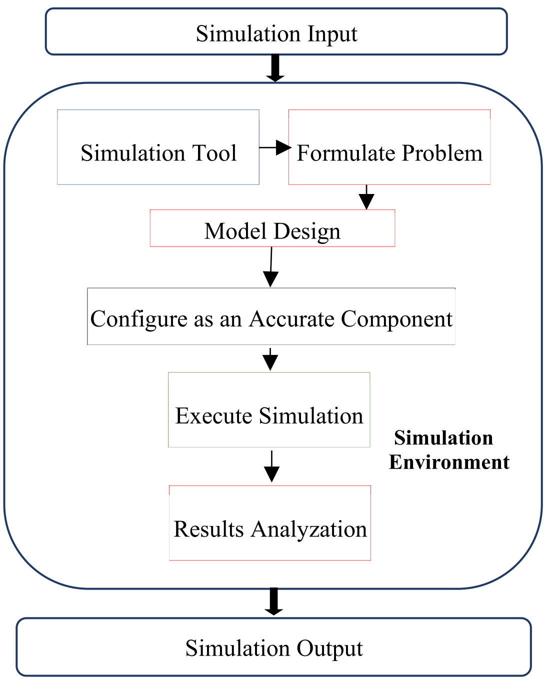

FIGURE 5. The basic architecture of OPNET.

Advantages of OPNET
- OPNET can simulate large-scale WSNs reaching up to a million network nodes.
- OPNET has a comprehensive user manual and a wealth of documentation. Additionally, the source code is accessible for users in the commercial edition.

Disadvantages of OPNET
- The main drawback of OPNET is that the sampling resolution limits the precision of the findings. This is because the simulation is wasteful during prolonged inactivity within the network operations.
- Due to the complicated GUI operation, simulating many nodes inside a single linked device is impossible.
- It is a commercial simulator, and licenses may be costly, especially for individual researchers or small organizations with restricted resources.
- Running simulations with large-scale WSN models in OPNET can be computationally demanding and require substantial computational resources, including CPU power and memory.
- Compared to open-source simulators like NS-3 or OMNeT++, the community support and resource availability for OPNET may be restricted.

6) PTOLEMY

Ptolemy is a broad domain simulator and an open-source software platform that supports network experimentation. A model is a hierarchically linked system of actors that operate simultaneously and exchange messages through related ports [68].

Each hierarchy level may have its director in a model, and different directors can be assembled in order. It is possible to expand the model to replicate the development and spread of WSANs across vast regions, such as a collection of city parks, using the design hierarchy offered by Ptolemy. [69]. The following is a summary of additional features:

Advantages of Ptolemy
- Designing and interpreting the implementation and code is easy. Moreover, Ptolemy consists of various software programs that may be utilized separately [70].
- Ptolemy is platform-neutral, threaded, and network-aware software since it is built on Java language.
- Ptolemy provided detailed API documentation and a user guide for new users.

Disadvantages of Ptolemy
- Creating models in Ptolemy might need extensive programming abilities, particularly in Java or AML. It may include coding complicated behavior and component interactions.
- Ptolemy is a general-purpose modeling and simulation framework that does not have built-in support for WSNs. Consequently, emulating WSN-specific features and protocols may need adaptation and development work.
- Ptolemy has visualization and analysis tools; however, they may be less comprehensive or specialized than those available in other dedicated WSN simulators.

7) GLOBAL MOBILE INFORMATION SYSTEM SIMULATOR (GLOMOSIM)

GloMoSim is an old freeware that simulates wired and wireless network protocols [71]. GloMoSim was used widely for WSNs because it was created as a collection of library modules [72]. GloMoSim covers several necessary network layers, including transport, application, data link, and network layer. Moreover, it supports radio propagation, radio models, and mobility. Additional features can be summarized below:

Advantages of GloMoSim

- Several operating systems, such as Windows, Linux, and Sun SPARC Solaris, supported GloMoSim's operation.
- GloMoSim enables the simulation of networks with a thousand nodes.
- GloMoSim supports many routing protocols, especially those used within MANET, such as AODV and Location-Aided Routing (LAR).

Disadvantages of GloMoSim

- GloMoSim does not present energy usage models for the transport layer and no specific routing protocols for sensor networks.
- All events must be obtained from nearby network nodes since GloMoSim does not handle phenomena outside the simulation context [73].
- GloMoSim is an old simulation tool that has not recently been actively maintained or updated. GloMoSim development and maintenance have slowed dramatically, which may result in restricted bug patches.
- The resource availability of GloMoSim may be limited due to its outdated status compared to other, more actively maintained simulation software. Locating particular WSN-related examples, models, or troubleshooting support may be difficult.
- GloMoSim was designed mainly to simulate mobile ad hoc networks (MANETs) and wireless networks in general. However, it may lack specific WSN-specific capabilities, protocols, or models.
- The documentation for GloMoSim is inadequate and unclear, making the new users struggle to find an easy way to learn to use the simulator.

8) J-SIM

The J-Sim simulator supports many research fields, including WSNs, WBANs, MANETs, and Satellite Networks (SN) [74]. Nowadays, many academics utilize the J-Sim network simulator as one of their reliable tools for their projects. Three main WSN protocols: Localization, Geographic Routing, and Directed Diffusion—have been implemented and presented using J-Sim. J-Sim simulator provides several benefits regarding execution speed, memory allocation, and scalability. Additionally, it offers a solid framework for large-scale WSN simulations involving more than 1000 nodes. Additional features of J-Sim can be summarized below:

Advantages of J-Sim

- J-Sim is a general simulator based on Java and is supported working under many operating systems, including Windows, Linux, and Mac.
- J-Sim gives users access to a powerful energy model. Moreover, J-Sim offers a user interface that is easy to use, making it very straightforward for users to call upon previously defined procedures [75].

Disadvantages of J-Sim

- J-Sim supports running on its platform independently. Thus, the simulation takes much longer than alternative simulation tools because of excessive run-time overhead, which represents one of the main drawbacks of this simulator.
- J-Sim may have difficulties when it comes to handling large-scale WSN simulations or situations with a large number of nodes. Therefore, J-Sim may not be as efficient or scalable as other current simulation tools regarding memory utilization and simulation run-time.
- Finding resources, examples, or community help for J-Sim WSN simulations may be more difficult.

9) MATRIX LABORATORY MATLAB

A general domain tool for calculation and visualization with a very high level of performance and a wide variety of feature-rich options. The program's most valuable attribute is the simplicity of MATLAB's programming capabilities, the foundation for users to build their unique functions quickly [76]. Simulink, an additional significant piece of software, is used as a support system for MATLAB at its backend. Simulink can model non-linear and linear systems in either continuous time, sampled time, or a combination of the two methods. Simulink offers a graphical interface for constructing block diagrams and drag-and-drop functionality for creating diagrams and the components they include [77]. Additional features of MATLAB can be summarized below:

Advantages of MATLAB

- Simulink provides a communication toolbox set that may be used to construct a whole WSN model system [78].
- MATLAB's programming environment comes with various toolboxes, some of which are fuzzy logic and symbolic computations.
- Many academics assert that MATLAB (Simulink) meets the most crucial design requirements and may be utilized successfully to model and create frameworks [79].
- MATLAB can simulate linear and non-linear systems using continuous time, sampled time, or a mix of the two [80].

Disadvantages of MATLAB

- Constructing the hardware architecture of the sending nodes and modeling the communication channel and the receiving node architecture are both required steps in the simulation approach.
- The MATLAB user community is vast and active, which is helpful for general MATLAB-related queries and conversations.
- MATLAB is a commercial product, and obtaining licenses may be costly, especially for individual researchers. The cost factor should be considered when choosing a simulation tool, mainly if budget limits exist.

- MATLAB may have performance and scalability constraints compared to more specialized simulation tools.
- MATLAB is used widely for WSN simulations but lacks specialized capabilities or models designed exclusively for WSNs.

10) QUALNET

A commercial and general-purpose network simulator can simulate various heterogeneous networks, including satellite, wired, wireless, and underwater networks [81]. QualNet was designed originally for quantum networks and offered a graphical tool for creating and visualizing experiments. QualNet has two modes in its architecture: design mode and visualization mode. Design mode is for developing experiments while visualizing performing and visualizing trials. A graphical application for viewing and analyzing packet traces is available from QualNet. Additional features of QualNet can be summarized below:

Advantages of QualNet

- QualNet has a robust and friendly GUI and supports an easy method for debugging and packet tracing [82].
- QualNet can offer numerous 2D and 3D network simulation scenarios.

Disadvantages of QualNet

- QualNet cannot guarantee result correctness due to oversimplified assumptions.
- QualNet is not open source and has a high cost of ownership. QualNet's community support and resource availability are restricted. Locating particular WSN-related examples, models, or troubleshooting support may be difficult.
- Modifying or building new models may need more time and experience because QualNet may not provide the same versatility as other simulators.

11) NETSIM

It is a stochastic discrete event simulator that quickly received much attention due to its unique features and broad simulation library support [83]. NetSim supports various protocols, including Carrier Sense Multiple Access with Collision Detection (CSMA/CD), MANET, and Aloha [84]. NetSim is available in three editions: Academic, Standard, and Pro. The academic version is used in labs and for teaching. The standard version is utilized for research and development at educational institutions, while the NetSim Pro version is used for industrial fields. Additional characteristics of the NetSim simulator are outlined below:

Advantages of NetSim

- NetSim is exceptionally straightforward to use and understand. This is due to its robust GUI and drag-and-drop functionality.
- NetSim supports various routing protocols, such as RIP, OSPF, DSR, AODV, ZRP, and OLSR.
- NetSim is a powerful tool that provides a metrics engine with packet trace, plot generator, and packet animator.
- Users may install, simulate, and evaluate network situations using the command level and GUI.
- The open C code for protocol libraries is accessible for user modification. Doing so may avoid the time-consuming process of developing, customizing, and configuring commercial simulators to suit a customer's particular demands.

Disadvantages of NetSim

- The DES within NetSim only supports one process. A single event queue is employed for the simulation, and it always has one entry for every network station.
- NetSim does not have a full free version.
- Compared to commonly used simulators like NS-2, NS-3, or OMNeT++, the community support and availability of resources for NetSim may be limited.

12) SCALABLE SIMULATION FRAMEWORK NETWORK MODELS (SSFNET)

A discrete-event network simulator is used for network testing and analysis. It is open-source software that may simulate many network situations, including network architecture, protocols, and traffic [85]. It also allows simulating Wide Area Networks (WANs) like the Internet. SSFNet has several benefits that make it famous for simulating WSNs, as outlined below:

Advantages of SSFNet

- SSFNet allows control over complicated traffic patterns.
- SSFNet, like other powerful simulators, supports parallel simulations in its operations.

Disadvantages of SSFNet

- SSFNet is not as actively developed or widely adopted as some other simulation tools. The development and support for SSFNet have declined over time, leading to limited updates, bug fixes, and a smaller user community.
- SSFNet did not give a comprehensive collection of pre-built models optimized for WSNs. End users may need to create or modify models to reflect WSN-specific traits, protocols, or behaviors, which may need more time and effort.
- Understanding the relationships between various abstraction layers is necessary for SSFNet, which is the hardest thing to use in this simulator.
- SSFNet suffers from delayed convergence that could happen in the presence of correlated long-distance traffic.

13) LABVIEW (LABORATORY VIRTUAL INSTRUMENT ENGINEERING WORKBENCH)

LabVIEW is unique among programming environments. Its development was not motivated by creating a new simulator but rather by the need to provide a tool to assist non-programmer scientists and engineers in automating the test and measurement systems they work on [86]. Since its introduction over two decades ago, the LabVIEW development environment has assisted in creating testing, measurement, and control applications, including WSNs simulation purposes [87]. Additional attributes of LabVIEW are outlined below:

Advantages of LabVIEW

- The infrastructure and development environment for LabVIEW is exceptional for visual programming languages.
- Users may create routing and other algorithms in LabVIEW by adding C code to the sensor node [88].

Disadvantages of LabVIEW

- LabVIEW is a graphical programming environment mostly used for data collecting, instrument control, and system monitoring. LabVIEW may be used to simulate WSNs. However, it lacks specialized capabilities and models.
- LabVIEW simulations may be computationally demanding and need a substantial memory and processing capacity, mainly when modeling large-scale WSN situations or dealing with many nodes.
- LabVIEW supports a wide range of operating systems. However, it has some compatibility constraints. It is mainly intended for Windows, so certain features and toolkits may differ based on the operating system.

14) MININET

Mininet is a network emulator that builds an artificial network of connections, switches, controllers, and hosts. Standard Linux network software is also used by Mininet hosts [89]. Mininet uses virtualization tools and the controller for scaling up the network size to hundreds of nodes. Mininet can simulate and test SDN, Wireless Personal Networks (WPANs), and fiber optic applications [90], [91]. Mininet is primarily designed for simulating SDNs and traditional computer networks. However, it is used widely for WSN simulation [92], [93]. Additional characteristics of this simulator are outlined below:

Advantages of Mininet

- Mininet offers a straightforward and affordable network testbed for creating OpenFlow applications.
- Mininet enables complicated topology testing without requiring the actual wiring of a network.
- It provides a basic set of parametrized topologies and supports arbitrary custom topologies.
- Mininet is functional right out of the box without scripting but also gives users access to a simple and flexible Python API for building and experimenting with networks.

Disadvantages of Mininet

- The main drawback of Mininet is a single server's CPU and bandwidth limitations.
- Mininet is currently unable to run applications that are not compatible with Linux.
- Mininet mainly aims to create virtual network topologies and emulate network behavior using virtual hosts and switches. Mininet lacks built-in models for modeling radio propagation, signal attenuation, or interference, which are frequent in WSNs. External interaction with other simulation tools may be required to implement certain wireless-specific behaviors.
- Mininet does not support large-scale WSNs or situations with many nodes.

B. SPECIFIC DOMAIN SIMULATORS (WSN SPECIFIC SIMULATORS)

Much effort has been put into developing specific simulators for WSNs. These simulators represent a high scientific and commercial contribution for WSNs instead of entirely depending on generic simulators. The developed simulation tools provided WSN's layers, WSN's functions, and every restriction related to the sensor node, including the energy consumption terms. Some of these simulation tools take other more specific ways to be related to limited to particular kinds of WSNs such as Vehicular Ad Hoc Networks (VANET), MANET, UWSN, and WBANs. The following subsections represent the study and analysis of the critical characteristics and drawbacks of the network simulators in this category.

1) TINYOS SIMULATOR (TOSSIM)

TOSSIM is mainly thought of as an emulator instead of a simulator. An emulator can imitate a model's software and hardware components. TOSSIM is regarded as a trustworthy and accurate emulator for WSN protocols and applications that use the Tiny Operating System (TinyOS) operating system [94]. TOSSIM is a discrete event simulator with a powerful simulation kernel that offers several ways to interact with WSNs, including monitoring packet traffic and injecting packets flexibly [95], as shown in Fig. 6.

TOSSIM is not necessarily the best simulation option; like any simulator, it makes assumptions about the target hardware platform, and simplifying specific behavior often makes it inaccurate. Additional characteristics of this simulator are outlined below:

Advantages of TOSSIM

- TOSSIM now offers a high-fidelity, scalable simulation of all WSN operations. Additionally, it contains a GUI tool called TinyViz that allows users to interact and see running simulations.
- TOSSIM provides adequate and straightforward to detect hidden terminal issues.
- TOSSIM employed C++ and Python. Python enables dynamic interaction with running simulations, acting as a robust debugger.

Disadvantages of TOSSIM

- TOSSIM's lack of energy measurement is a significant drawback [96].
- TOSSIM requires familiarity with the TinyOS programming framework and the nesC programming language. Users must get acquainted with the syntax, principles, and constraints of nesC, which may have a steeper learning curve than other frequently used programming languages.

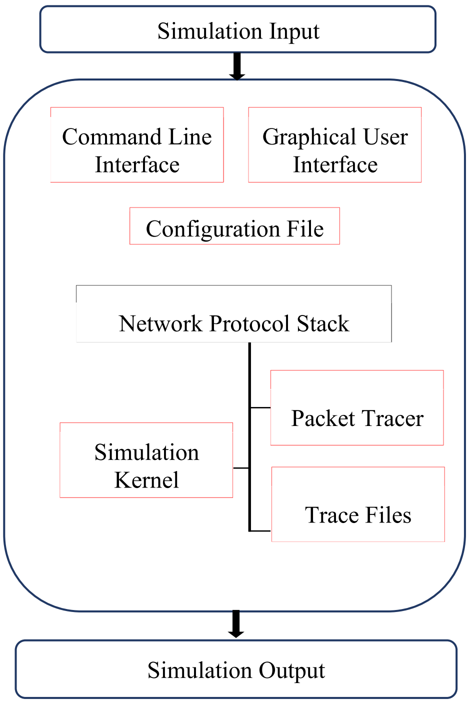

FIGURE 6. The basic architecture of TOSSIM.

- TOSSIM primarily simulates the physical and link layers of WSNs. It supports fewer network layer protocols, routing techniques, and higher-level behaviors. Custom development may be required to add network layer functionality [97].
- TOSSIM may have scalability limits, particularly for large-scale WSN simulations or situations with a large number of nodes. As the number of nodes or simulation complexity grows, simulation performance and memory use may become bottlenecks.
- TOSSIM is firmly incorporated with the TinyOS framework. Therefore, compatibility or interoperability with other WSN simulation frameworks or tools may be restricted. Combining simulations using components from multiple frameworks might be difficult.

2) EMSTAR

EmStar is a real-time, trace-driven emulator created in C, particularly for WSNs. This emulator aids in developing WSN applications on more advanced hardware sensors [98]. Emstar's primary objective is to decrease design complexity so that work may be shared and reused and create new sensor network applications simpler and faster. Emstar offers a straightforward environmental model and network medium to design, build, and deploy heterogeneous sensor network applications [99]. However, Emstar is not as effective and quick as other simulation tools. Additional characteristics of the Emstar simulator are outlined below:

Advantages of EmStar

- EmStar contains services to sustain sensing, message passing, and time synchronization. Hence, it was previously used for WSNs and MANET [100].
- EmStar supports a range of commonly used communication protocols in WSNs, such as IEEE 802.15.4 and ZigBee.
- EmStar is an open-source simulation tool, making it freely accessible to academic researchers.

Disadvantages of EmStar

- The framework of EmStar is less efficient than alternative options because EmStar does not support parallel simulation.
- Simulating a vast number of sensors is beyond the capabilities of EmStar.
- EmStar only offers support of the code for particular sorts of nodes it is intended to operate with.
- EmStar has a potentially high learning curve. Its sophisticated interface or documentation may require much work and time to grasp.
- EmStar provides a graphical interface that simplifies the operation of electrical equipment for end-users. However, EmStar's visualization capabilities may be restricted or less advanced than other simulation tools.

3) OVERSIM

OverSim is a free software and a new overlay framework for simulating WSNs written in C++ [101]. The framework was created to address various issues with the peer-to-peer simulators that are currently available. The OMNeT++ network simulator is the foundation for OverSim, which includes several overlay protocols. Since OverSim is built on OMNeT++, it gains the advantages of features like robust GUI support and a compelling event scheduler [102]. For structured peer-to-peer networks, OverSim offers several standard functions that make it easier to develop new protocols. Based on OMNeT++, OverSim is compatible with a variety of operating systems, including Linux, Windows, and Mac OS X. Additional characteristics of this simulator are outlined below:

Advantages of OverSim

- OverSim framework offers a central module for processing and collecting statistics. The post-processing tools provided by OverSim make it easier to create charts suitable for publishing.
- For large-scale network simulations, several exchangeable models enable modeling more complex heterogeneous underlay networks and simpler networks. It is possible to simulate overlay networks with OverSim with up to 100,000 nodes [103].
- The simulator tracks various statistical information, including packet delivery success or failure, transmitted, received, or forwarded network traffic per node, and packet hop count.
- OverSim's website supports extensive documentation and source code for new users.

Disadvantages of OverSim

- OverSim may lack comprehensive built-in functionality for some WSN communication protocols. Depending on the protocols users want to imitate, users may need to put more effort into developing or integrating them into the framework.
- OverSim may have a smaller user community than other frequently used simulation software. As a result, accessing tools, documentation, or community assistance for WSN simulations utilizing OverSim may be more difficult.

4) SENSOR NETWORK SIMULATOR AND EMULATOR (SENSE)

SENSE is an effective WSN simulator that simulates a large-scale network [104]. SENSE is a C++-based discrete event simulator that operates on top of a general-purpose discrete event simulator known as a Component-Oriented Simulation Toolkit (COST). SENSE is a powerful simulation tool like J-Sim. However, SENSE is still in the early stages of development. Although the simulator's core has steadily been established, it still lacks a complete collection of models, routing protocols, and a broad range of WSN setup templates [105]. Additional characteristics of SENSE are outlined below:

Advantages of SENSE

- SENSE Supports battery and mobility models in WSNs.
- SENSE supports numerous routing protocols, including Ad Hoc On-Demand Distance Vector (AODV), dynamic source routing (DSR), and many self-selecting routing protocols, which are supported by SENSE (SSR).
- SENSE offers parallelization support, much like GloMoSim.
- It is possible to analyze the simulation results and display message flow in the network using a visualization tool called iNSpect and a GUI tool called G-SENSE.
- SENSE includes tools and capabilities for doing in-depth performance analysis. It enables users to collect information on network behavior, such as message passing, energy usage, and latency. This extensive study assists in the identification of bottlenecks, the evaluation of protocol efficiency, and the optimization of network performance.
- The SENSE simulator's guide user interface is designed specifically for WSN simulation. It contains parameters input section, stop time, number of nodes, source nodes, packet size, and interval.

Disadvantages of SENSE

- While SENSE is mainly designed for Linux-based operating systems, running it on other Unix-like operating systems, such as macOS, with the proper setups and dependencies may be feasible. It should be noted, however, that the tool's documentation and support resources are primarily geared toward Linux users.
- Finding specific resources or support for SENSE simulations may be more difficult due to the small user community compared with other simulators like NS-2 and OMNeT++.

5) SHAWN

A high-performance discrete event simulator for sensor networks created in C++. It is speedy, customizable, and may be modified to whatever the simulation or application needs precision. The main goal of Shawn is to substitute replaceable, abstract models for a network's physical layer so that massive networks may be simulated quickly. Additional characteristics of Shawn are outlined below:

Advantages of Shawn

- Shawn is an extremely quick simulator. For example, it can execute the same simulation in less than a minute instead of 25 hours for Ns-2's [106].
- Shawn is designed to simulate immense networks with a few hundred thousand nodes.
- A key component of Shawn's strategy is simulating an impact rather than the occurrence itself. Example: In Shawn, the consequences of packet loss and corruption are represented, not the whole Media Access Control (MAC) layer, which includes a radio propagation model.
- Shawn supports both the Linux Operating System and Windows through Cygwin.
- Shawn comes with a graphical visualization tool named "Viz.".

Disadvantages of Shawn

- The main drawback of this simulator is that no stable release has been produced yet. Shawn is a simulation tool with limited development and updates in recent years. As a result, it may lack certain features or compatibility with newer technologies and protocols.
- Shawn's compatibility with other frameworks and tools is restricted. Integrating Shawn with additional libraries or increasing its functionality may involve extra work or code changes.

6) GEORGIA TECH SENSOR NETWORK SIMULATOR (GTSNETS)

This simulation tool allows the creation and assessment of algorithms for massive WSNs. GTSNetS is constructed on top of an old Georgia Tech Network Simulator (GTNetS), which extends and inherits all of the GTNetS simulator's architectural choices [107]. According to the developers' testimony, the sensor network simulator can handle networks with more than 200000 nodes [108]. Additional features of this simulator are outlined below:

Advantages of GTNetS

- Using GTSNetS, precise data on a particular sensor network may be gathered at the functional unit, node, and network levels.
- GTSNetS presented several energy models for the many sensor types of WSN. This feature allows the user to choose the energy model that best meets his requirements.

- Specifying mobile sensor nodes, moving detected objects, and a mobile base station is possible.
- GTNetS allows the development and evaluation of algorithms for large-scale WSNs. GTSNetS provides different energy and battery models, network and application protocols, and tracing options. Furthermore, the users could easily extend or replace the available models for a specific requirement.

Disadvantages of GTNetS

- The serious drawback of this simulation tool is that it has not been updated and supported for many years. As a result, it may lack certain features or compatibility with newer technologies and protocols for WSNs.
- GTSNetS is an old simulation tool with limited development. As a result, it may lack certain features or compatibility with newer technologies and protocols for WSNs.

7) SENSORSIM

SensorSim is another framework for modeling WSNs based on the NS-2 simulator and includes several add-on capabilities [109]. The primary goal of SensorSim's development is to give more knowledge of sensor networks and a robust foundation for creating novel protocols and performance assessment methods for sensor networks.

The hybrid simulation mode of SensorSim enables additional capabilities, including the interaction of actual nodes, new communication protocols, and real-time user interaction with the GUI interface. Other attributes of the SensorSim simulator are outlined below:

Advantages of SensorSim

- SensorSim supports energy-efficient WSN protocols that optimize energy use and network lifespan, such as LEACH, TEEN, and PEGASIS. These protocols
- SensorSim supports the simulation of IEEE 802.15.4 networks, a low-power, low-data-rate WSN communication technology that forms the basis for protocols like ZigBee.
- SensorSim can simulate data aggregation methods, which combine sensor node data to decrease network data transfer, such as Directed Diffusion and TinyDB.
- SensorSim supports major WSN routing protocols such as AODV, DSDV, and DSR.

Disadvantages of SensorSim

- NS-2 is the foundation upon which SensorSim is constructed. Thus, SensorSim has a scalability issue, and the performance of the simulation might degrade as the network size increases.
- Despite its many benefits and innovative features for modeling WSN networks, SensorSim is not yet accessible to the general public owing to the developers' inability to support the public release.
- SensorSim's visualization and analysis may be less advanced than other simulation tools. Analyzing and understanding simulation findings may involve extra work or other tools, making it harder to glean indications.

8) AVRORA

Avrora offers a straightforward Java API and infrastructure for experimentation, profiling, and analysis. It also includes a flexible framework for emulating and evaluating WSNs. Users may use the monitoring infrastructure to integrate online activity monitoring for greater program comprehension and improvement potential [110]. Additional characteristics of this simulator are outlined below:

Advantages of Avrora

- A complete selection of modeling tools for WSNs is available. This simulator minimizes the problems of TOSSIM and ATEMU while combining their benefits [111].
- AVR-based microcontroller MICA2 sensor nodes can be simulated using Avrora.
- Avrora offers open-source and online documents and supports energy consumption models.
- Avrora is a powerful simulation tool that supports a wide range of protocols such as IEEE 802.15.4, ZigBee, ZigBee, ContikiMAC, Routing Protocol for Low-Power and Lossy Networks (RPL), and IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN).
- Avrora provides more remarkable speed, scalability, and precision than many WSN simulators, such as TOSSIM.

Disadvantages of Avrora

- Avrora's compatibility with other frameworks may be limited. Integrating Avrora with external libraries or extending its functionalities might require extra effort or modifications to the existing codebase.
- The main drawback is it lacks a GUI and is incompatible with many networks' communication tools.
- Avrora simulations can have a steep learning curve. Understanding and configuring the simulation environment, including setting up the software stack and libraries, may require significant effort and time.

9) CASTALIA

Castalia is a popular open-source simulator for WSNs and WBANs. Researchers and academics may use it to assess distributed algorithms and protocols in proper wireless channel and radio models [112]. Castalia is built on OMNeT++ and is not regarded as a sensor-specific platform. Instead, it was developed to provide real-time results of the algorithm being developed by the researcher on any specific sensor platform. Additional characteristics of this simulator are outlined below:

Advantages of Castalia

- Castalia offers a channel model to examine the route between nodes, the mobility of the nodes, and any interference levels concerning signal intensity.
- Castalia supports packet size, module type, and carrier sensing, which may be interpreted using a radio model.

- Castalia works on pathways rather than lines. Therefore, users can simulate the moving nodes anywhere inside the simulation space [113].
- Castalia supports various WSN protocols, such as time synchronization, data aggregation, routing, and MAC protocols.

Disadvantages of Castalia

- Castalia may not support protocols that depend on sensor node hardware. These protocols may need hardware-specific simulation models or libraries.
- Castalia suffers from noticeable execution time for more extensive network sizes followed by higher complexity; thus, Castalia fails to handle the network correctly.

10) TRAFFIC AND NETWORK SIMULATION ENVIRONMENT (TRANS)

It is a free and open-source GUI simulation program that combines the traffic and network simulators SUMO and NS-2 to provide accurate simulations of VANETs [114]. TraNS can create the mobility traces using this architecture before running a network simulation. Trans aims to prevent findings from simulations from materially deviating from those attained by actual tests, as was the case with earlier iterations of mobile ad hoc networks. Using TraNS, the data shared in a VANET may affect how the vehicles behave in the mobility model. Thus, depending on car communication, the network simulator may apply realistic mobility models and affect how the traffic simulator behaves. Additional characteristics of this simulator are outlined below:

Advantages of TraNS

- TraNS's GUI makes setting up all the necessary simulation settings, including the network's topology, easy and fast.
- TraNS is a simulation environment that integrates a mobility generator and a network simulator, providing a tool to build a realistic VANET while providing feedback between the vehicle behavior and the mobility model.
- TraNS provides two ready-to-use VANET applications: Road Danger Warning (safety) and Dynamic Reroute (traffic efficiency).
- TraNS is written in Java and C++ and works under Linux and Windows (trace-generation mode)
- TraNS can provide Google Earth visualization (currently works for TIGER files only)., which is a digital database of geographic features, such as roads, railroads, rivers, lakes, and political boundaries.

Disadvantages of TraNS

- The TraNS simulator has poor API documentation and user guides, leading to a steep learning curve for new users.
- TraNS's development has been put on hold. As a result, TraNS does not support the most recent versions of NS-2 and Sumo.
- TraNS can simulate only a limited number of nodes (tested up to 3000 cars).

11) GROOVENET

GrooveNet is a free software simulator for geographic routing that considers the necessity for a reliable, user-friendly, and realistic simulation [115].

GrooveNet supports the simulation of wireless communication. More significantly, GrooveNet has previously been verified using data from actual applications to guarantee the correctness of the simulation. It is intended to be an opportunistic broadcast protocol with little to no shared state information between neighbors and little handshaking between transmitting and receiving parties. Additional characteristics of this simulator are outlined below:

Advantages of GrooveNet

- GrooveNet has well-laid-out model interfaces that simplify incorporating various network models.
- GrooveNet looks into the broadcast storm issue and employs several rebroadcast rules.
- It can simultaneously accommodate thousands of moving and speaking vehicles with mobility, travel, and communication models.
- The graphical interface makes it easy to auto-generate simulations consisting of thousands of vehicles across any location.
- GrooveNet is used widely for VANET, which consists of groups of moving or stationary vehicles connected by a wireless network [116].
- GrooveNet supports multiple message types, such as GPS messages broadcast periodically to inform neighbors of a vehicle's current position [117].
- GrooveNet supports intelligent WSNs management, which is also based on SDNs.

Disadvantages of GrooveNet

- Configuring the simulation environment, including the complex configuration files and modules, may require significant effort and time due to poor API documentation and a complicated user guide.

12) DYNAMIC ROUTING MODEL SIMULATOR (DRMSIM)

It is an open-source discrete event simulator that seeks the large-scale assessment of the routing model [118]. The primary foundation of the DRMSim simulator is Java software, allowing for flexibility and reusability of the code. The DRMSim simulator currently supports the routing protocols Routing Information Protocol (RIP), Border Gateway Protocol (BGP), and Non_Stop active Routing (NSR).

DRMSim is a simulator designed to minimize the code quality created to enhance extensibility and reusability. Additional characteristics of this simulator are outlined below: The high-level calculation is offered by DRMSim, which computes performance metrics such as communication cost, routing route, and routing table size. Additional characteristics of this simulator are outlined below:

Advantages of DRMSim

- DRMSim can simulate dense, large-scale WSNs. Scalability support lets researchers test data-centric, reliable

multicast algorithms in realistic and difficult network situations.
- DRMSim is a customizable and configurable tool that allows researchers to customize simulations to their needs. Users may specify parameters and alter simulation settings to explore different situations and protocols.

Disadvantages of DRMSim
- DRMSim has limited use and cannot simulate many protocols because it simulates data-centric, reliable multicast protocols in WSNs.
- DRMSim cannot be supported under Windows. In contrast, many operating systems supported this simulator, such as DRMSim UNIX, Linux, and Mac.
- DRMSim does not employ distributed or parallel discrete event simulation methods.
- DRMSim must have at least 4G of RAM to run the bundled software. If memory is inadequate, the software's performance may suffer, or the simulation process may be tainted.

13) MANNASIM
MANNASIM is a framework for WSN simulation based on the NS-2 [119]. By adding additional modules for designing, developing, and analyzing various WSN applications, MANNASIM expands NS-2 to be specific for WSN simulation.

This framework offers a method for configuring the environments by setting different variables, such as the data distribution method, routing hardware capabilities, and sensor nodes' initial power sources [120]. Additional features of this simulator are outlined below:

Advantages of MANNASIM
- MANNASIM can provide different performance metrics, including the amount of energy remaining in the nodes, the types and quantities of simulation faults, and network lifetime.
- MANNASIM offers a complete testbed for simulating several WSN algorithms and protocols and a sophisticated simulation framework for accurately representing sensor nodes [121].
- MANNASIM can simulate the operations of several WSN routing protocols, including AODV, Direct Diffusion, Destination Sequenced Distance Vector (DSDV), Dynamic Source Routing (DSR), Low-Energy Adaptive Clustering Hierarchy (LEACH), and Temporally Ordered Routing Algorithms (TORA).

Disadvantages of MANNASIM
- Users unfamiliar with ns-2 simulations may find MANNASIM simulations challenging to understand.
- MANNASIM's user base and resource availability may be less than those of other simulation tools. Finding MANNASIM simulation materials, documentation, and community assistance may be more complex.

14) NETTOPO
An extendable integrated framework of simulation and visualization for WSNs. The primary goal that guided the design and development of the NetTopo Simulator was to research several different algorithms used in WSNs [122]. In terms of simulation, users can define many parameters over sensor nodes. Some examples of these parameters include energy consumption, bandwidth management, and NetTopo. All of these can be done effectively while considering large-scale heterogeneous networks. The Java programming language was used to develop NetTopo, which includes more than 80 Java classes and 11,000 lines of code [123]. Additional characteristics of this simulator are outlined below:

Advantages of NetTopo
- NetTopo supports two routing protocols: Two Phase Geographic Greedy Forwarding (TPGF) and Greedy Perimeter Stateless Routing (GPSR).
- The topology of the WSN may be easily customized within NetTopo.
- NetTopo presented simple modifications of the characteristics of sensor nodes based on the attributes given by the user.
- NetTopo supports an essential feature regarding files that allow users to store and retrieve their simulated processes.
- NetTopo supports a graphical visualization tool that is simple to use and supports both 2D and 3D.
- The data acquired on the real WSN testbed may be shown using the visualization tool NetTopo provides.

Disadvantages of NetTopo
- NetTopo focuses on visualization and analysis rather than simulation. It visualizes network topologies and sensor node locations but may not simulate protocol modeling or performance assessment.

15) COOJA
COOJA simulator is explicitly built for Contiki, an operating system developed for settings with little memory, such as the nodes used in WSNs [124]. The COOJA simulator enables simultaneous cross-level modeling at three levels: the machine code instruction set, the application, and the operating system. A single simulation is performed using COOJA, which includes both a low-level simulation of the hardware of sensor nodes and a simulation of high-level behavior. Additional characteristics of this simulator are outlined below:

Advantages of COOJA
- It is possible to modify any feature of the COOJA system, including the sensor node platforms, the operating system software, the radio transceivers, and the radio propagation models. This makes the system versatile and expandable.
- COOJA is a Java program, and all interaction with built Contiki code is accomplished via the Java Native Interface (JNI) [125].

- COOJA supports several Contiki-specific communication protocols and stack components, including Routing Protocol for Low-Power and Lossy Networks (RPL), Constrained Application Protocol (CoAP), and IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) [126].
- COOJA allows researchers to evaluate WSN applications' energy efficiency. It monitors and reports node energy use for energy-aware optimizations and performance enhancements.
- Contiki and an active user community support COOJA. Online resources, forums, and documentation enable knowledge exchange, collaboration, and troubleshooting.

Disadvantages of COOJA

- The lack of a GUI is the main drawback of this simulation tool.
- COOJA may struggle to simulate large-scale WSNs with many nodes or dense network deployments.
- COOJA supports several WSN communication protocols, especially those compatible with the Contiki operating system. However, it does not support all protocols, forcing users to build or incorporate them into the simulation, which may increase development time.
- Analyzing and understanding simulation findings may involve extra work and other tools, making it harder to glean insights.

16) (ATMEL EMULATOR) ATEMU

ATEMU is a software emulator for WSNs that is based on AVR microcontrollers. These microcontrollers belonging to the AVR family find widespread use in the MICA sensors. ATEMU aims to bridge the gap between real sensor network deployments and simulations of WSN deployments [127]. The capability of ATEMU to mimic a heterogeneous sensor network is one of the instrument's defining characteristics. That means ATEMU can run different sensor nodes that can execute different codes. Additional characteristics of this simulator are outlined below:

Advantages of ATEMU

- ATEMU can set unlimited breakpoints, and memory watchpoints are supported.
- ATEMU supports the ability to execute code based on TinyOS.
- ATEMU supports symbolic debugging support, including source-level stepping and run-time variable inspection for programs compiled in the ELF format.
- ATEMU simulations are not simple simulation tools for only TinyOS applications. ATEMU may be used to design alternative operating systems for sensor platforms by modeling alternative operating systems for sensor platforms [128].

Disadvantages of ATEMU

- The primary disadvantage of using ATEMU is that the simulation time required is much longer when compared to that required by other simulation tools.
- ATEMU is limited in the number of functions that may be used to simulate network routing and clustering issues, which is another of its shortcomings.
- It may not accurately reflect the intricacies and real-world characteristics of WSNs. Due to abstractions or simplifications introduced in the simulation model, simulated results could not precisely match those of real-world WSN implementations. This might cause a disconnect between simulation results and actual performance.
- ATEMU faces scalability challenges when simulating large-scale WSNs with a high number of nodes or dense network deployments.

17) PLATFORM FOR INTEGRATED COMMUNICATIONS AND CONTROL DESIGN (PICCSIM)

PiccSIM Platform is a unique simulation tool created to provide a comprehensive collection of tools for designing, simulating, and implementing WSNs [129].

PiccSIM uses the Simulink tool to simulate dynamic systems, and NS-2 is used for network simulation. Moreover, a GUI creates and models the network and control system. PiccSIM may be known as the PiccSIM Toolchain when Simulink and NS-2 are combined, and this toolchain can be used to create and simulate actual wireless nodes [130]. Additional characteristics of the PiccSIM simulator are outlined below:

Advantages of PiccSIM

- The PiccSIM simulator makes a complete toolset for designing, simulating, and implementing wireless network control systems (WiNCS) and network-controlled systems (NCS).
- PiccSIM provided a mechanism for unusual data sharing between Simulink and NS-2.
- PiccSIM supports complete automatic network node code creation.

Disadvantages of PiccSIM

- A remote user interface that makes it possible to simulate without a PiccSIM platform.
- PiccSIM simulations may have a challenging learning curve, particularly for users unfamiliar with NS-2 and MATLAB.
- PiccSIM may experience scaling challenges when modeling large-scale WSNs with many nodes. As the network grows, the simulation's performance can slow.

18) PROBABILISTIC WIRELESS SENSOR NETWORK SIMULATOR (PROWLER)

PROWLER is a specific domain simulator for WSN running under MATLAB, which provides collaborative settings with a wide range of reliable and precise implicit numerical capabilities. Additionally, PROWLER offers accessible prototyping applications and can simulate wireless dispersed networks [131]. Additional characteristics of this simulator are outlined below:

Advantages of PROWLER

- PROWLER is oriented towards the MICA platform, which is generally used for real implementations in the field of study of WSNs.
- PROWLER simulation is based on real-time events, allowing easy and direct routing protocol implementation on a real prototype.
- PROWLER provides a library of standard routing components, leading to the possibility of efficiently improving new protocol proposals [132].
- PROWLER can model wireless distributed networks from the application layer to the physical communication layer.
- PROWLER can be used efficiently to verify protocols such as Span free, Collision, Floor 2D, and Floor 1D.
- The main feature of the PROWLER simulator is its capability to simulate sensor node functioning on TinyOS and other generic systems.

Disadvantages of PROWLER

- Customizing the network is complex by using Prowler tools.
- PROWLER does not support protocols for the ZigBee network.
- PROWLER is a robust probabilistic traffic tool, according to the description. However, it often yields inadequate results.
- PROWLER lacks support for the simulation of large-scale networks due to scalability issues.

19) UNDERWATER SIMULATOR (UWSIM) UWSIM WAS CREATED EXPLICITLY FOR UWSNS, A SUBTYPE OF WSN

This simulator was created explicitly for UWSNs, a subtype of WSN. During the past years, the emphasis on network simulators predominantly revolved around ad hoc networks and ground-based sensor systems, leading underwater modeling to be neglected. Research and development in maritime robotics are UWSim's main priorities [133]. The C++-based UWSim simulator uses osgOcean and Open Scene Graph (OSG). Developers utilize the open-source OSG 3D graphics API to create applications for virtual reality, scientific visualization, visual simulation, and other uses. Additional features of this simulator are outlined below:

Advantages of UWSim

- The UWSim can be customized to provide information about the underwater terrain, seabed environment, robots, sensors, and more, configured through XML files [134].
- Underwater communication performance factors, including low bandwidth, low frequency, high transmission, and constrained memory, are primarily the focus of UWSim. Instead of layers or protocols, it is based on component-based techniques [135].
- While Windows was the primary operating system used to support the UWSim previously, only the Ubuntu operating system is supported by the most recent version of UWSim.
- Researchers may incorporate a variety of simulated sensors with the simulation environment, including a camera, object picker, pressure sensor, Inertial measurement unit (IMU) and Doppler velocity log (DVLD), GPS, force sensor, and structured light projector.

Disadvantages of UWSim

- UWSim is focused on modeling underwater robotics and underwater sensor networks and has a narrow range of applications, so other network or scenario types may not be suitable for it to simulate.
- While UWSim makes an effort to simulate underwater environments, the simulation's accuracy and realism might vary depending on the quality and complexity of the models employed.

C. CUSTOM BUILD SIMULATORS

Many researchers faced problems coding their ideas to design new algorithms and propose novel approaches. The researchers' considerable efforts to practice gaining complete knowledge about the existing simulators represent the main challenge. Furthermore, in many cases, the general domain simulators and even the specific domain simulators cannot cover the wide range of ideas researchers produce. In addition, many existing simulation tools are not customizable. This represents the main drawback to dealing with the available simulators. Moreover, many general and specific simulation tools are designed to simulate specific purposes for particular applications in WSNs, so in many situations, these directions do not match the user's aims.

All these reasons make scientific researchers build self-developed simulators independently without depending on the available simulators. The developed simulators indeed serve as excellent performance analysis tools. These custom build simulators have successfully contributed to the repository of tools for WSNs analysis. The basic idea is related to the abstraction process and abstraction level of the WSN operations and how to reflect the same logical concept using the programming language. This direction in the research area is not new at all; many attempts have been recorded to build WSN simulators that can act as performance analysis tools for various fields in the WSN, such as routing algorithms, data gathering approaches, fault tolerance frameworks, energy holes bypassing strategies [107], [136], [137], [138].

This class of simulators has main advantages, such as no time wasted to learn about available simulators and no need to consume time for deep searching of some functions or frameworks. In addition, the researcher will have complete control of the developed simulator, including the designed GUI and the kind of collected results. On the other hand, there are some restrictions to this kind of self-developed simulator. First, the researcher must have adequate knowledge of a programming language, different data structures, and DES concepts. Second, customer build simulators may sometimes be inaccurate due to oversimplification of the WSN basics.

IV. EVALUATION METHODOLOGY FOR THE AVAILABLE WSN SIMULATORS

Various simulators have different preferences for their fundamental constants, and as their functions are tailored to the specific study topics they are intended for, their architectures are also distinct. Therefore, while assessing various simulators, it is crucial to compare them based on their essential constants, their usability for specific domains, and their ability to calculate particular metrics. This assessment procedure is fundamental to ensure the evaluation is comprehensive and precise.

The survey examines and evaluates the dominant simulation tools from three different perspectives: general criteria for any simulator, the prevalent and most popular performance metrics, and the dominant domains that can use the simulation tool under examination. The following subsections illustrate the evaluation process in detail for the three mentioned perspectives.

A. EVALUATION BASED ON THE GENERAL CRITERIA

There are a variety of criteria that emphasize both the strengths and shortcomings of the simulator compared with other simulators. Based on these criteria, simulators may be evaluated. Table 4 illustrates the comparison among different simulation tools based on the general criteria, which are described as follows: -

- **Graphical User Interface (GUI)**
  The provision of GUI support is a subject of great interest and desire from several perspectives. For instance, it may be a valuable tool for debugging and provide a practical approach to rapidly identify undesirable behavior based on visual observation and tracking the step-by-step execution of a simulation [139]. In contrast, a Command Line Interface (CLI) simulation is a text-based user interface used to run simulations. This approach mostly used specific log files to record statistics and simulation output. Network simulators can be classified into two subclasses based on the GUI availability: GUI and CLI simulators.
- **License Type**
  WSN simulators can be classified into accessible and commercial software [140]. Open-free software simulators are made available to the public, and the source code is also available. Users of open free software simulators are not required to pay anything. It is accessible with no licensing fees. In contrast, a valid and permitted license must be purchased to utilize commercial simulators. The source code is secured, and users of commercial simulators must fork out moderate to high costs.
- **Basic Construction Programming Language**
  Simulators are developed based on single or various programming languages such as C++, Java, and Python. The scientific researcher may choose the appropriate network simulator for his work depending on the programming language closest to him and whether he is familiar with its details.
- **Portability**
  Portability refers to the ability of a simulation to be used for multiple topics or classes with little or no modification. WSN simulators, according to this characteristic, can be described as simulators with high or restricted extensibility.
- **User manual and Application Programming Interface (API)**
  High-quality software documentation explains to users and developers what a software system does, how it works, and how to use it [141]. Generally, the term "user manual" describes the instructions given to customers for a product or service. For those not already acquainted with the particular simulator and how it works, many simulators provide several excellent works of literature. The API documentation, conversely, is a technical content deliverable that includes guidelines for efficiently using and integrating with the simulator.
- **Scalability**
  Simulators of WSNs faced challenges in terms of scalability due to the additional complexity brought about by the interaction with the surrounding environment and the complex novel applications. The simulation of several hundred thousand nodes currently remains a challenging problem. Moreover, simulators of WSNs faced challenges in terms of scalability due to the additional complexity brought by the interaction with the surrounding environment and the complex novel applications. Many cases reported that some simulators do not scale more than a specific number of sensor nodes during simulation.
- **Statistical backing**
  The result of a simulator is another crucial factor to consider while studying it. The result must be readily adaptable and near the intended assumptions for statistical analysis and graph development. The simulator should have a replicable way of verifying results again and again. Some WSN simulators provide minimal statistical backing, while others support moderated or fully supported statistical backing.
- **Simulator Type**
  Discrete Event Simulation (DES), Trace Driven Simulation (TDS), and mixed-based simulation are the three leading simulators that are all possible and available. DES represents systems where events happen at particular moments and produce state changes [142]. Time is considered continuous in a DES, but the simulation only advances when discrete events occur. DES simulators keep a priority queue of events and process them in the order in which they appear. DES is ideal for modeling systems with many discrete events, such as computer networks, manufacturing procedures, or transportation systems. Unlike DES, TDS represents systems where time is split into distinct periods or steps. The simulation in a TDS advances in discrete time increments, with state changes occurring at the start and

TABLE 4. The comparison for evaluating the WSN simulators based on the general criteria.

Simulator	Accuracy	Scalability	GUI availability	License type	Programming language	Portability	API documentation and user manual	Statistical backing	Simulator type
NS-3	High	Large scale	Available	Open free software	C++ and optional Python bindings	Extensible	Comprehensive API documentation and user manual	Available	Discrete event
NS-2	High	Limited (5000) nodes	Available	Open free software	C++ and OTCL	Restricted extensibility	Comprehensive API documentation	Limited	Discrete event
OMNeT++	High	Limited (3000) nodes	Available	Open free software	C++	Extensible	Comprehensive API documentation and user manual	Limited	Discrete event
PeerSim	Moderate	Large scale (1,000,000) nodes	Not available	Open free software	Java	Extensible	Comprehensive API documentation and inadequate user guide	Limited	Discrete event
OPNET	High	Large scale	available	Free academic license with use restrictions	C++ and C	Extensible	Comprehensive API documentation and user manual	Available	Discrete event
Ptolemy	Moderate	Limited	available	Open free software	Java	Restricted extensibility	Comprehensive API documentation and user manual	Limited	Discrete event
GloMoSim	Low	Large scale	Limited	Open free software	C and Parsec	Restricted extensibility	Poor API documentation and Poor user guide	Limited	Discrete event
J-Sim	Low	Limited (1000) nodes	Available	Open free software	Java and Jacl	Extensible	Inadequate API Documentation and Poor User Guide	Available	Discrete event
MATLAB	Moderate	Limited	Available	Non-open free software	C and Java	Restricted extensibility	Comprehensive API documentation and user guide	Available	Discrete event
QualNet	Moderate	Large scale (20000) nodes	Available	Commercial and open free licenses only for academicians	C and C++	Extensible	Comprehensive API documentation and user manual	Available	Discrete event
NetSim	Moderate	Large scale	Available	Non-open free software	C and Java	Extensible	Comprehensive API documentation and user manual	Available	Trace driven
SSFNet	Moderate	Large scale (100,000) nodes	Available	Open free software	C++ and Java	Restricted extensibility	Comprehensive API documentation and user manual	Available	Discrete event
LabVIEW	Low	Limited	Available	Non-open free software	C, C++, and Java	Restricted extensibility	Comprehensive API documentation and user manual	Available	Discrete event
Mininet	Low	Limited	Not available	Open free software	Python	Restricted extensibility	Comprehensive API documentation and user manual	Limited	Discrete event
TOSSIM	Low	Limited (1000) nodes	Available	Open free software	C++, Python, and NesC	Extensible	Comprehensive API documentation and inadequate user guide	Available	Discrete event
EmStar	Low	Limited	Available	Open free software	C	Restricted extensibility	Inadequate API documentation and poor user guide	Limited	Trace driven
OverSim	Moderate	Large scale (100,000) nodes	Available	Open free software	C++	Extensible	Comprehensive API documentation and user manual	Available	Discrete event
SENSE	High	Large scale	Available	Open free software	C++	Extensible	Comprehensive API documentation and user manual	Available	Discrete event
Shawn	Low	Large scale (100,000) nodes	Not available	Open free software	C++	Restricted extensibility	Comprehensive API documentation and user manual	Limited	Discrete event
GTSNetS	High	Large scale	Available	Open free software	C++	Extensible	Inadequate API documentation and comprehensive user guide	Available	Discrete event
SensorSim	Moderate	Limited	Not available	Open free software	C++/ Based on NS-2	Restricted extensibility	Inadequate API documentation and poor user guide	Limited	Discrete event
Avrora	Moderate	Large scale	Not available	Open free software	Java	Restricted extensibility	Comprehensive API documentation and user manual	Limited	Discrete event
Castalia	High	Large scale	Available	Open free software	C++/Based on OMNeT++	Extensible	Comprehensive API documentation and user manual	Available	Discrete event
TraNS	Moderate	Limited (3000) nodes	Available	Open free software	C++ and Java	Restricted extensibility	Poor API documentation and poor user guide	Limited	Discrete event
GrooveNet	Moderate	Large scale	Available	Open free software	C++	Restricted extensibility	Inadequate API documentation and inadequate user guide	Limited	Mixed event
DRMSim	Low	Large scale (10,000) nodes	Available	Open free software	Java	Extensible	Comprehensive API documentation and user manual	Available	Discrete event
MANNASIM	Moderate	Large scale	Available	Open free software	Java/Based on NS-2	Extensible	Comprehensive API documentation and inadequate user guide	Available	Discrete event
NetTopo	Moderate	Limited (5000) nodes	Available	Open free software	Java	Extensible	Comprehensive API documentation and inadequate user guide	Available	Discrete event
COOJA	Low	Limited	Not available	Open free software	Java	Restricted extensibility	Poor API documentation and poor user guide	Limited	Discrete event
ATEMU	Low	Limited	Available	Open free software	C	Restricted extensibility	Comprehensive API documentation and user manual	Limited	Discrete event
PiccSIM	Low	Limited	Available	Open free software	MATLAB and NS-2	Restricted extensibility	Comprehensive API documentation and user manual	Limited	Discrete event
PROWLER	Low	Limited	Available	Open free software	MATLAB and Java	Restricted extensibility	Comprehensive API documentation and user manual	Limited	Probabilistic discrete event
UWSim	Moderate	Limited (4000) nodes	Available	Open free software	C++ and NET framework	Restricted extensibility	Comprehensive API documentation and user manual	Limited	Discrete event

TABLE 5. The comparison for evaluating the WSN simulators based on dominant performance metrics.

Simulator	Dominant Performance Metrics									
	Network Throughput	E2E Delay	PDR	Energy Consumption	Network lifetime	Network coverage	Routing Overhead	Latency	Jitter	Battery Life
NS-3	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
NS-2	✓	✓	✓	✓	✓	✓	✓	×	×	×
OMNeT++	✓	✓	×	✓	✓	×	✓	×	×	×
PeerSim	✓	✓	✓	✓	×	×	✓	✓	✓	×
OPNET	✓	✓	✓	✓	×	✓	✓	✓	×	×
Ptolemy	✓	✓	✓	×	×	✓	✓	×	×	×
GloMoSim	✓	✓	✓	✓	×	×	✓	✓	✓	×
J-Sim	✓	✓	✓	✓	✓	×	✓	✓	×	×
MATLAB	✓	✓	✓	×	✓	✓	✓	✓	✓	×
QualNet	✓	✓	✓	✓	✓	×	✓	✓	✓	×
NetSim	✓	✓	✓	✓	✓	✓	✓	×	✓	×
SSFNet	✓	✓	✓	×	×	×	✓	✓	✓	×
LabVIEW	✓	✓	✓	×	×	×	×	✓	✓	×
Mininet	✓		✓	×	×	✓	×	✓	✓	×
TOSSIM	✓	✓	✓	✓	×	✓	×	×	×	✓
EmStar	✓	✓	✓	×	×	✓	✓	✓	✓	×
OverSim	✓	✓	✓	×	×	✓	✓	✓	✓	×
SENSE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Shawn	✓	✓	✓	✓	✓	✓	×	×	×	×
GTSNetS	✓	✓	✓	✓	✓	×	✓	×	×	×
SensorSim	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
Avrora	✓	✓	✓	✓	×	✓	✓	✓	✓	✓
Castalia	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TraNS	✓	✓	✓	×	×	×	✓	✓	✓	×
GrooveNet	✓	✓	✓	×	×	✓	✓	✓	✓	×
DRMSim	✓	✓	✓	×	×	×	✓	✓	✓	×
MANNASIM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NetTopo	✓	✓	✓	✓	×	✓	✓	✓	✓	×
COOJA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ATEMU	✓	✓	✓	×	×	×	✓	✓	✓	×
PiccSIM	✓	×	✓	×	×	✓	✓	×	×	×
PROWLER	✓	✓	✓	✓	✓	×	×	✓	×	×
UWSim	✓	✓	✓	×	×	×	×	✓	✓	×

end of each time step. TDS is often used when modeling systems with continuous state changes, such as physical systems or financial simulations [143]. Discrete-time Markov chains and differential equation models are two examples of TDS.

On the other hand, mixed-based simulators involve DES and TDS features. They are capable of dealing with systems that have both discrete events and continuous state changes. Mixed-based simulators provide more flexible modeling by combining the benefits of DES and TDS.

These simulators are often employed in complicated systems with discrete and continuous behaviors, such as traffic simulations or complex supply chain models. It is vital to note that the kind of simulator used is determined by the system's features being simulated and the unique simulation needs.

B. EVALUATION BASED ON DOMINANT PERFORMANCE METRICS

Various performance metrics can provide suitable indications about any WSN's characteristics, including the network's reliability, integrity, and dependability. This work has chosen the ten most popular performance metrics that will be used to examine the simulator's ability to calculate these metrics accurately, as shown in Table 5. Following are short descriptions of these performance metrics: -

- **Network Throughput**
  Network throughput is a metric representing how much data is received throughout the whole network over time [144].
- **End-to-End (E2E) Delay**
  This criterion represents the time a packet travels from its source to its destination through the network [145].
- **Packet Delivery Ratio (PDR)**
  This metric identifies the ratio of the data received to the data sent over the entire network [146].
- **Energy Consumption**
  This parameter illustrates the average energy usage of network nodes [147].
- **Network Lifetime**
  Is the time taken from the network's initialization to the moment the network's first node dies [54].
- **Network Coverage**
  Network coverage is a criterion of the ability of the sensor nodes to detect events or changes in the environment within their coverage area [148].

TABLE 6. The comparison for evaluating the WSN simulators based dominant domains.

Simulator	WSN's Dominant Domains	Domains not included/Limitations
NS-3	▪ Routing [152][153] ▪ Energy Consumption [154][155] ▪ Security [156][157][158] ▪ QoS [159][160] ▪ Data Aggregation [161] ▪ Transmission Control [162] ▪ Node Deployment [163]	▪ It is not easily customizable for packet formats and energy models. ▪ It does not include any Disruption Tolerant Network (DTN) routing protocols. ▪ It is a time-consuming tool to have complete knowledge about it due to the complexity of learning. NS-3 cannot support the simulation of the energy consumption of LR-WPAN.
NS-2	▪ Routing [54][164][165][166] ▪ Fault Tolerance [54] ▪ Energy Consumption [167] ▪ Security [168][169] ▪ QoS [169][170] ▪ Data Aggregation [167] ▪ Node Deployment [165][170][171]	▪ It does not support the simulation for large-scale WSNs. ▪ Several researchers have voiced their dissatisfaction with the complex learning process of NS-2. ▪ The object-oriented architecture in NS-2 introduces some unnecessary interdependencies between modules that can lead to difficulties in adding new protocols.
OMNeT++	▪ Node Localization [172] ▪ Routing [173][174] ▪ Fault Tolerance [175][176] ▪ Energy Consumption [177][178] ▪ Security [46] ▪ QoS [179] ▪ Data aggregation [180][181]	▪ Computational resource problems, especially when simulating large-scale WSNs or complex protocols. ▪ It has a steep learning curve compared to other simulation tools. ▪ Unlike other simulation tools, OMNeT++ alone does not provide an extensive library of pre-built models tailored explicitly for WSNs. ▪ Combining individual models in OMNeT++ is complex and may result in high-probability bugs.
PeerSim	▪ Routing [182] ▪ Energy Consumption [183] ▪ Security [184] ▪ Node Deployment [182]	▪ Limited availability of WSN's modules. ▪ It does not provide debugging facilities and does not have a graphical user interface. ▪ The most significant drawback is that command base simulator inputs are presented in a text file. ▪ PeerSim does not have a comprehensive set of built-in wireless communication models for modeling radio propagation and signal attenuation, which is typical in WSNs.
OPNET	▪ Fault Tolerance [185] ▪ Node Deployment [185][186] ▪ Node Mobility [187] ▪ Routing [187][188]	▪ OPNET is slow for simulating real-time applications and may require technical hardware or cloud-based aids. ▪ OPNET uses a complicated GUI operation, so simulating many nodes inside a single linked device is impossible. ▪ It is a commercial simulator, and licenses may be costly for individual researchers.
Ptolemy	▪ Routing [120][189] ▪ Energy Consumption [120] ▪ Security [189] ▪ Fault Tolerance [190]	▪ Creating models in Ptolemy might need extensive programming abilities, particularly in Java or AML. ▪ The simulation accuracy depends on the network model used in the simulation. ▪ Simulation WSN-specific features and protocols may need adaptation and development work [68].
GloMoSim	▪ Energy Consumption [191] ▪ Routing [192] ▪ Node Localization [193]	▪ GloMoSim does not present energy usage models for the transport layer and no specific routing protocols for sensor networks. ▪ It does not provide built-in support for security protocols. ▪ GloMoSim is not suitable for simulating WSNs in 3D environments.
J-Sim	▪ Fault Tolerance [194] ▪ Security [194] [195] ▪ QoS [196] ▪ Node Mobility [194]	▪ The simulation takes a much longer time than alternative simulation tools, representing one of this simulator's main drawbacks. ▪ J-Sim can evaluate the performance of WSN protocols, but it may not be the best choice for developing new protocols. ▪ Finding resources, examples, or community help for J-Sim WSN simulations may be more difficult. ▪ J-Sim may have difficulties when it comes to handling large-scale WSN simulations.
MATLAB	▪ Routing [197][198] ▪ Node Deployment [199] ▪ Fault Tolerance [198][200] ▪ Node Localization [201][202] ▪ Security [203]	▪ MATLAB is a commercial product, and getting licenses may be costly. ▪ It does not provide detailed energy usage models for WSN nodes; therefore, the users evaluated it mathematically. ▪ Not suitable for the simulation of security issues against complex attacks. ▪ Does not support the simulation of the cross-layer interaction. ▪ MATLAB has scalability constraints.
QualNet	▪ Routing [204][205] ▪ Node Deployment [206]	▪ It is primarily designed for network layer simulation. Thus, it is not suitable for physical layer simulation. ▪ It does not provide models for the simulation of complex security protocols and attacks. ▪ Result correctness cannot be guaranteed by QualNet owing to oversimplified assumptions. ▪ Modifying or building new models may need more significant time and experience because QualNet may not provide the same amount of versatility as other simulators.
NetSim	▪ Routing [207] ▪ Data Aggregation [208] ▪ Energy Consumption [207][209] ▪ Node Deployment [210]	▪ The DES within the NetSim only supports one process. ▪ NetSim provides specific fundamental security models. However, it may not be appropriate for complex security issues like intrusion detection. ▪ NetSim does not correctly simulate heterogeneous sensor nodes.

TABLE 6. ***(Continued.)*** **The comparison for evaluating the WSN simulators based dominant domains.**

SSFNet	▪ Security [211][212]	▪ SSFNet does not support the simulation of the physical layer of WSNs since it is mainly focused on higher-layer network protocols. ▪ It is designed for stationary WSNs, so it does not support the simulation of mobility WSNs. ▪ SSFNet development and upgrades have been restricted in recent years. This may cause challenges with newer technologies, protocols, or models typically utilized in present WSN research. ▪ SSFNet did not give a comprehensive collection of pre-built models optimized for WSNs.
LabVIEW	▪ Monitoring [88][213][214] ▪ Node Deployment [215] ▪ Energy Consumption [216]	▪ LabVIEW cannot simulate large-scale WSNs. ▪ Not suitable for complex and dynamic routing protocols. ▪ It does not provide any physical layer models for WSN simulation. ▪ It is not appropriate for simulation radio channels and noise. ▪ It does not support the simulation of many security techniques, such as encryption and access control. ▪ LabVIEW may be used to simulate WSNs. However, it lacks specialized capabilities and models because it is a graphical programming environment.
Mininet	▪ Energy Consumption [217]	▪ It is not appropriate for the simulation of large-scale WSNs. ▪ The main drawback of Mininet is a single server's CPU and bandwidth limitations. ▪ It does not provide flexibility for modifying its sensor models. ▪ It is rarely used nowadays for simulation WSNs.
TOSSIM	▪ Energy Consumption [94][97] ▪ Node Mobility [218] ▪ Routing [219] ▪ Node Localization [220]	▪ TOSSIM may have scalability limits, particularly for large-scale WSNs. ▪ It is not appropriate for simulating networks with highly dynamic topologies. ▪ TOSSIM primarily simulates the network layer of WSNs and may not be suitable for simulating interactions between different layers.
EmStar	▪ Node Deployment [221] ▪ Noise Modeling [222]	▪ It does not Simulation of large-scale WSNs. ▪ It is not appropriate for security issues simulation. ▪ It does not support mobility simulation. ▪ It is not appropriate to simulate heterogeneous WSNs. ▪ It does not support any parallel simulation domain. ▪ EmStar's visualization capabilities may be restricted or less advanced than other simulation tools. ▪ Rarely used nowadays for WSNs simulation.
OverSim	▪ Routing [101] ▪ Fault Tolerance [223] ▪ QoS [224]	▪ It does not support the simulation of security issues. ▪ It does not support the simulation of energy consumption and battery life issues. ▪ It does not support the simulation of propagation and radio channel effects. ▪ Finding specific resources, up-to-date documentation, or community support for OverSim simulations could be more challenging.
SENSE	▪ Routing [225]	▪ It does not support high dynamic and complex WSN topologies. ▪ It does not support any custom protocol simulation due to difficulties in supporting external models. ▪ Rarely used nowadays for simulation WSNs
Shawn	▪ Topology Control [226] ▪ Routing [227]	▪ It does not support efficient node mobility simulation. ▪ Shawn cannot simulate interference and other environmental factors. ▪ The main drawback of this simulator is that no stable release has been produced yet. ▪ Shawn is an old simulation tool with limited development and updates in recent years. As a result, it may lack certain features or compatibility with newer technologies and protocols. ▪ Shawn cannot simulate any routing protocols outside the supported models by itself.
GTSNetS	▪ Energy Consumption [228] ▪ Large Scale WSNs [108]	▪ The serious drawback of this simulation tool is that it has not been updating and supporting the simulator for many years. ▪ It is not customizable for some WSN simulations and is not flexible to modify simulation parameters.
SensorSim	▪ Energy Consumption [229] ▪ Fault Tolerance [230] ▪ Data Aggregation [109] ▪ Routing [231]	▪ SensorSim has a scalability issue. ▪ It does not support complex network topologies such as hierarchical topologies. ▪ Even though SensorSim supports several routing protocols, it is not preferable for simulation complex and experimental routing protocols. ▪ It does not support the simulation of the cross-layer interaction.
Avrora	▪ Energy Consumption [232] ▪ Node Deployment [233] ▪ Fault Tolerance [234] ▪ MAC Protocols [235][236] ▪ Security [104][237]	▪ It does not support heterogeneous networks. ▪ It does not provide a clock drift model or support complex node models. ▪ It does not support the simulation of complex security issues.
Castalia	▪ Routing [238][239] ▪ Energy Consumption [240] ▪ Data Aggregation [241] ▪ Security [242]	▪ It is not customizable because it is challenging to modify certain aspects of the simulator to suit specific needs. ▪ Castalia may not support protocols that depend on sensor node hardware. These protocols may need hardware-specific simulation models or libraries. ▪ Castalia suffers from high execution time for more extensive network sizes.

TABLE 6. *(Continued.)* **The comparison for evaluating the WSN simulators based dominant domains.**

TraNS	▪ Realistic Mobility [243] ▪ Security [240]	▪ It does not support energy consumption simulation due to a lack of energy models. ▪ It does not support the simulation of complex security issues. ▪ It does not support the simulation of large-scale WSNs. ▪ TraNS's development has been put on hold. As a result, TraNS does not support the most recent versions of NS-2 and Sumo. ▪ The TraNS simulator has a poor user guide and API documentation, leading to a steep learning curve for new users.
GrooveNet	▪ Security [115][244] ▪ Routing [245]	▪ It does not provide high-accuracy simulation, and it is not customizable at all. ▪ It requires significant computation resources. ▪ Complex configuration files and modules may require considerable effort and time due to poor API documentation and an unclear user guide.
DRMSim	▪ Routing [246]	▪ DRMSim does not employ distributed or parallel DES methods. ▪ It does not support heterogeneous WSNs. ▪ It may not accurately model WSNs' real-world complexity due to simulation model simplifications or abstractions.
MANNASIM	▪ Routing [120][247] ▪ Fault Tolerance [248]	▪ It may be limited by the NS-2 simulator's syntax, settings, and learning curve. ▪ Finding MANNASIM simulation materials, documentation, and community assistance may be more complex. ▪ It does not support the simulation of complex security issues.
NetTopo	▪ Routing [249] ▪ Energy Consumption [250]	▪ NetTopo focuses on visualization and analysis rather than simulation. ▪ It does not support the restricted energy model and does not support complex security issues. ▪ It does not support complex routing protocols.
COOJA	▪ Node Localization [251][252][253] ▪ Energy Consumption [254] ▪ Routing [255]	▪ The lack of a GUI is the main drawback of this simulation tool. ▪ COOJA does not support the simulation of large-scale WSNs with many nodes or dense network deployments Simulation of large-scale WSNs. ▪ It is difficult to customize many routing protocols supported by this simulation tool.
ATEMU	▪ Security [256]	▪ The primary disadvantage of using ATEMU is that the simulation time required is much longer when compared to that required by other simulation tools. ▪ It has limited simulation functionality for complex routing and clustering issues. ▪ Power management cannot be modeled. ▪ It does not support the simulation of complex WSNs because it will require significant computational resources. ▪ It is rarely used nowadays for simulation WSNs.
PiccSIM	▪ QoS [257] ▪ Communication Control [258]	▪ PiccSIM simulations may have a challenging learning curve, particularly for users unfamiliar with NS-2 and MATLAB. ▪ It does not support the simulation of large-scale WSNs. ▪ It may not be suitable for many WSN simulations because its characteristics make it specific only for IEEE 802.15.4 wireless protocol issues.
PROWLER	▪ Node deployment [259] ▪ Security [260] ▪ Routing [132] ▪ Energy Consumption [261]	▪ It does not support Complex WSN topologies like hierarchical topology. ▪ It is a robust probabilistic traffic tool. However, it often yields inadequate results. ▪ It does not support the simulation of large-scale networks due to scalability issues.
UWSim	▪ Mac Protocols [262][263] ▪ Transportation Control [264][265] ▪ Node Coverage [266] ▪ Sonar Image Simulation [267]	▪ It is not easy to customize and does not support the simulation of large-scale WSNs. ▪ UWSim is focused on modeling underwater robotics and underwater sensor networks; other network or scenario types may not be suitable for it to simulate. ▪ UWSim 's accuracy and realism might vary depending on the quality and complexity of the models employed.

- **Routing Overhead**
  It is the percentage of the total number of routing control packets sent by all sensor nodes to the number of data packets received at the final destination [149].
- **Latency**
  This parameter refers to the period it takes for a data packet to transit through a network from point to point.
- **Jitter**
  The variation in packet transit times across a network is known as jitter. It often indicates a signal's deviation from its actual periodicity [150].
- **Battery life**
  Battery life is when a battery can operate a sensor node before needing to be replaced or recharged [151].

C. EVALUATION BASED ON DOMINANT DOMAINS

WSN has various domains under development that are the core of numerous research publications. Routing protocol enhancements, fault tolerance issues, MAC protocol developments, and energy consumption problems are some of these domains. Based on the simulation tool's architecture, some simulators are efficient tools for specific fields, while others may not be suitable for use in the exact domains. Table 6 compares simulation tools based on the most popular disciplines in WSNs.

V. DISCUSSION AND EVALUATION

As a result of the outstanding level of abstraction that they offer, simulators are an excellent option for usage throughout the earliest phases of design and development. Event-driven simulators are common WSN simulators that simulate the network by considering sensor field events such as modifications to sensed data, node failures, and communication occurrences. Event-driven simulators help research localized phenomena like data aggregation, routing protocols, and energy consumption patterns because they allow for fine-grained modeling of WSNs.

The event-driven simulators TOSSIM, COOJA, and NS-2 are a few examples of powerful performance analysis tools. Finding the right balance between realism and processing efficiency is one of the main issues in WSN simulation. Resource limitations, communication protocols, and environmental dynamics must all be considered when simulating a large-scale WSN with thousands of nodes. Simulators must balance precisely simulating the behavior of the network with producing findings in a timely manner.

Additionally, several WSN simulators have visualization features that let users view network behavior during or after simulation. The spatial distribution of sensor nodes, mobility nodes' motion patterns, and the network's data flow may all be understood using visualizations. Visual representations offer a simple method for understanding simulation results and pinpointing possible performance problems. It is important to note that the selection of a WSN simulator is based on the particular needs of the investigation or application. Considerations should be made for elements such as the network's scale, the complexity required, the supporting communication protocols, and the accessibility of modeling libraries. When choosing a simulator for WSN research or development projects, other vital factors to consider are the simulators' documentation, community support, and simplicity of use.

Even though OOP simulators make implementation and expansion more straightforward because of their modular design, these simulators are not scalable. Simulators written in OOP languages such as Java and C++ may run on any platform, although their execution speed is often slower. Simulators that are built using components are more effective, as well as offering better levels of flexibility, reusability, and scalability.

On the other hand, commercial simulators such as the popular OPNET simulator provide a friendly graphical user interface, assistance, and support in simulating sensor-specific hardware. However, these commercial simulators come at a high financial expense. Similarly, specific domain simulators like GloMoSim, QualNet, and TOSSIM offer parallel execution but leave some concerns unaddressed. Due to the simulators' tendency to make too simplistic assumptions, most of them cannot ensure the correctness of the findings.

Table 4 and Table 5 summarize the comparison for evaluating the WSN simulators based on different criteria, and many conclusions can be drawn from these tables. First, some of the general domine simulators are powerful tools and can be used for WSN simulation. NS-3, NS-2, OMNeT++, and OPNET are vital tools for this purpose. Secondly, many WSN-specific simulators are not supported anymore and suffer from many unresolved issues. However, good simulators such as TOSSIM, Castalia, and Avrora can still be used effectively. Third, as shown in Table 6, many simulators have been successfully used for many fields in WSN, NS-2, OMNeT++, MATLAB, TOSSIM, etc. However, some simulators are unsuitable for simulation purposes in specific domains or conditions.

Furthermore, this study investigated over 260 research articles and surveys, which were selected meticulously to encompass the specific simulators being evaluated. Among these simulators, NS-2, NS-3, and OMNeT++ garnered significant attention in publications concerning WSNs domains, as depicted in Fig. 7. Numerous researchers have presented innovative ideas utilizing these simulators, highlighting the robustness of the current simulation tools (NS-2, NS-3, and OMNeT++). As previously stated, scalability plays a crucial role in simulations. Simulators such as NS-2, OMNeT++, J-Sim, and TOSSIM are recognized for their robust capabilities in supporting various WSN domains. Nevertheless, it is worth noting that these simulation tools face challenges when simulating large-scale networks, as demonstrated in Fig. 8.

Therefore, based on those mentioned earlier and the comparison tables presented, there is an unspoken truth that should be clear now. Even though powerful simulation tools such as NS-3, Ns-2, OMNeT++, OPNET, Castalia, and Shawn exist, no perfect simulator can offer all the required functionality for the WSN. Every single simulator, either general or specific domain, has its features and shortcomings. Thus, there is a real need for new simulation tools designed explicitly for WSN simulation purposes. The urgent need arises for a simulator that can be scalable, highly accurate, and provide a friendly graphical user interface.

Moreover, the new simulator must be easy to learn with clear and adequate documentation so the new users do not suffer in the learning process and do not waste a long time gaining this knowledge. This is essential because the prime researcher aims to implement his idea about the WSN field, not to learn or add thoughts about using a network simulator. In conclusion, the old but renovated direction, building custom simulators, has been gaining more and more attention recently due to its flexibility and valuable features, such as the design control process.

VI. LESSONS LEARNED AND FUTURE DIRECTIONS

From the simulation domain perspective, it is clear that simulation is the most essential method for performance analysis of the WSNs. Much hard work has been done to

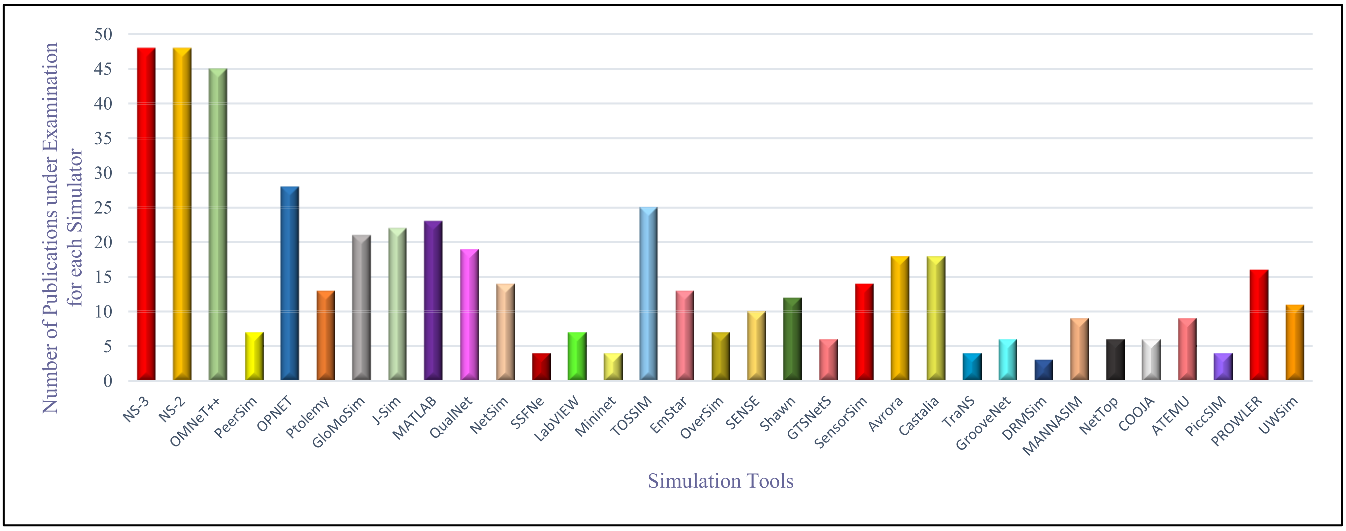

FIGURE 7. Number of publications regarding WSNs simulation tools.

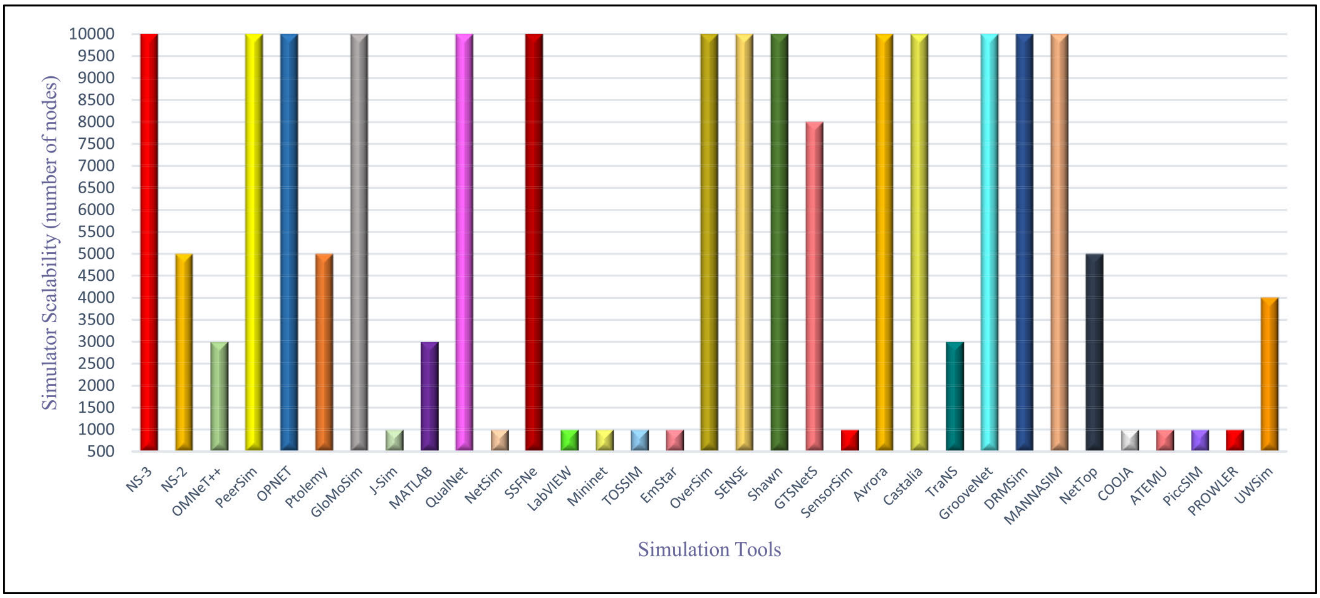

FIGURE 8. Scalability abilities regarding WSNs simulation tools.

present a wide range of simulation tools for the WSNs. Some of these simulation tools are designed specifically for the WSNs' simulation purposes. However, due to the wide variety of WSN fields and their various applications, no single simulator can be used for all areas of the WSNs. Even though there are many powerful simulators, it is not practicable or realistic to claim they can ultimately be used for all WSN's fields. Some simulators can effectively be used for specific areas of WSNs, such as fault tolerance, routing, and data gathering. On the other hand, the same simulators face many difficulties when used in different fields, such as security, QoS, node localization, and realistic mobility, as shown clearly in Table 5. Moreover, many famous and influential simulation tools still face problems, such as poor scalability to simulate large-scale WSNs, the absence of a friendly GUI, poor user guide documentation, and a steep learning curve for new users.

In addition, several performance analysis tools cannot accurately measure many crucial performance metrics in WSNs because of the absence of built-in tools for measuring these performance metrics. Users must implement custom methods to estimate complex metrics such as jitter, battery lifetime, and signal strength, as shown in Table 4.

On the other hand, a new trend has recently risen and gained much attention due to academic researchers' many difficulties using general simulation tools. This trend represents the offering of developed simulation tools by individuals for specific tasks. Based on the concepts of DES and OOP, many researchers, especially postgraduate scholars, presented powerful performance analysis tools that can be used for particular fields within WSNs. These kinds of self-developed simulators successfully contributed to the repository of tools for WSN analysis. However, custom-built simulation tools need a high level of programming knowledge by the developers. Therefore, this study concludes that future advancements regarding simulation tools' workflow have a promising research line for large-scale scientific workflows that would work toward the simulation tools of WSNs.

The scientific community will undoubtedly continue to enhance the current simulators by adding more features. In addition, future directions certainly will be related to present new simulation tools that match the continuous development of WSNs research and applications.

VII. CONCLUSION

WSNs are still gaining attention as an essential stand-alone technology and the backbone for other technologies such as IoT and IR 4.0. Simulation is critical for WSNs as it is a crucial phase before implementation. Simulation costs are the lowest among other alternatives, and it can simulate WSN with thousands of nodes. Various simulation tools for WSNs are designed to be under the service of the researchers and the developers. This work presented a comprehensive search of the previous studies regarding WSN simulators. In addition, this work offered a new taxonomy for WSN simulators that classifies them into three classes: general domain simulators, WSN-specific simulators, and WSN custom build simulators. A total of thirty-three simulators have been discussed based on different criteria. The study reached two critical facts regarding performance analysis tools. First, NS-3, NS-2, OMNeT++, OPNET, MATLAB, Castalia, TOSSIM, and Avrora are the most potent and popular performance analysis tools regarding the comprehensive literature review examined in this work. Second, there is not yet an entirely professional simulator that can be used for all simulation purposes of WSNs. All available simulators suffered from a shortage at least in one or two concepts, such as the lack of customizability or missing some critical WSN models, underscoring the significance of the third category of WSN simulators. Custom-built simulators are crucial because they can enrich the available simulation tools. Based on the OOP programming languages and DES concepts, researchers can build their simulators to simulate the specific domain or dominated performance metric of the WSN. The future work for this study is to construct a WSN simulator explicitly designed for the simulation of routing protocols and fault detection approaches and resolve the problems of the energy holes in WSN topology. The proposed simulator will be able to provide many performance metrics, such as packet error rate, network lifetime, latency, false alarm rate, and node event detection accuracy.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Communication Technology and Network Research Group, Faculty of Computer Science and Information Technology (FSKTM). They would also like to acknowledge the University of Putra Malaysia (UPM) for providing facilities during the preparation of this article.

REFERENCES

[1] M. Majid, S. Habib, A. R. Javed, M. Rizwan, G. Srivastava, T. R. Gadekallu, and J. C. W. Lin, "Applications of wireless sensor networks and Internet of Things frameworks in the industry revolution 4.0: A systematic literature review," *Sensors*, vol. 22, no. 6, p. 2087, Mar. 2022, doi: 10.3390/s22062087.

[2] W. Liang, C. Ma, M. Zheng, and L. Luo, "Relay node placement in wireless sensor networks: From theory to practice," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1602–1613, Apr. 2021.

[3] G. H. Adday, S. K. Subramaniam, Z. A. Zukarnain, and N. Samian, "Fault tolerance structures in wireless sensor networks (WSNs): Survey, classification, and future directions," *Sensors*, vol. 22, no. 16, p. 6041, Aug. 2022, doi: 10.3390/s22166041.

[4] K. Jaiswal and V. Anand, "FAGWO-H: A hybrid method towards fault-tolerant cluster-based routing in wireless sensor network for IoT applications," *J. Supercomput.*, vol. 78, no. 8, pp. 11195–11227, May 2022, doi: 10.1007/s11227-022-04333-6.

[5] M. B. Dowlatshahi, M. Kuchaki Rafsanjani, and B. B. Gupta, "An energy aware grouping memetic algorithm to schedule the sensing activity in WSNs-based IoT for smart cities," *Appl. Soft Comput.*, vol. 108, Sep. 2021, Art. no. 107473, doi: 10.1016/j.asoc.2021.107473.

[6] H. Hayat, T. Griffiths, D. Brennan, R. P. Lewis, M. Barclay, C. Weirman, B. Philip, and J. R. Searle, "The state-of-the-art of sensors and environmental monitoring technologies in buildings," *Sensors*, vol. 19, no. 17, p. 3648, Aug. 2019, doi: 10.3390/s19173648.

[7] B. S. Awoyemi, A. S. Alfa, and B. T. Maharaj, "Network restoration in wireless sensor networks for next-generation applications," *IEEE Sensors J.*, vol. 19, no. 18, pp. 8352–8363, Sep. 2019, doi: 10.1109/JSEN.2019.2917998.

[8] I. Nassra and J. V. Capella, "Data compression techniques in IoT-enabled wireless body sensor networks: A systematic literature review and research trends for QoS improvement," *Internet Things*, vol. 23, Oct. 2023, Art. no. 100806, doi: 10.1016/j.iot.2023.100806.

[9] F. Ojeda, D. Mendez, A. Fajardo, and F. Ellinger, "On wireless sensor network models: A cross-layer systematic review," *J. Sensor Actuator Netw.*, vol. 12, no. 4, p. 50, Jun. 2023, doi: 10.3390/jsan12040050.

[10] A. R. C. Serafini, L. Delforno, J. M. Palma, F. H. Behrens, and C. F. Morais, "Robust static output-feedback control for MJLS with non-homogeneous Markov chains: A comparative study considering a wireless sensor network with time-varying PER," *Sensors*, vol. 21, no. 19, p. 6420, Sep. 2021, doi: 10.3390/s21196420.

[11] P. Nayak, G. K. Swetha, S. Gupta, and K. Madhavi, "Routing in wireless sensor networks using machine learning techniques: Challenges and opportunities," *Measurement*, vol. 178, Jun. 2021, Art. no. 108974, doi: 10.1016/j.measurement.2021.108974.

[12] I. Y. Orea-Flores, M. E. Rivero-Angeles, A. L. Onofre-Soto, E. G. Azpeitia-Rebollar, N. Torres-Cruz, I. Villordo-Jiménez, A. Pretelín-Ricárdez, and R. Menchaca-Mendez, "Teletraffic analysis of energy-efficient intruder detection using hash function techniques in images for remote monitoring in wireless sensor networks," *Comput. Electr. Eng.*, vol. 103, Oct. 2022, Art. no. 108373, doi: 10.1016/j.compeleceng.2022.108373.

[13] E. Moridi, M. Haghparast, M. Hosseinzadeh, and S. J. Jassbi, "Fault management frameworks in wireless sensor networks: A survey," *Comput. Commun.*, vol. 155, pp. 205–226, Apr. 2020, doi: 10.1016/j.comcom.2020.03.011.

[14] P. Y. Dattatraya and J. Agarkhed, "Simulation an art of performance evaluation in wireless sensor networks," in *Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT)*, Mar. 2016, pp. 1–5, doi: 10.1109/ICCPCT.2016.7530235.

[15] H. M. A. Fahmy, *Concepts, Applications, Experimentation and Analysis of Wireless Sensor Networks*. Cham, Switzerland: Springer, 2021.

[16] A. Musa and I. Awan, "Functional and performance analysis of discrete event network simulation tools," *Simul. Model. Pract. Theory*, vol. 116, Apr. 2022, Art. no. 102470, doi: 10.1016/j.simpat.2021.102470.

[17] R. Sharma, V. Vashisht, and U. Singh, "Modelling and simulation frameworks for wireless sensor networks: A comparative study," *IET Wireless Sensor Syst.*, vol. 10, no. 5, pp. 181–197, Oct. 2020, doi: 10.1049/iet-wss.2020.0046.

[18] H. S. Vimala, "Comprehensive review on congestion detection, alleviation, and control for IoT networks," *J. Netw. Comput. Appl.*, vol. 221, Jan. 2024, Art. no. 103749, doi: 10.1016/j.jnca.2023.103749.

[19] S. Verma, S. Kaur, and A. D. Gupta, "A brief survey on simulators for designing routing protocol for wireless sensor network," in *Proc. 14th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, Jul. 2023, pp. 1–6, doi: 10.1109/icccnt56998.2023.10307096.

[20] S. Idris, T. Karunathilake, and A. Förster, "Survey and comparative study of LoRa-enabled simulators for Internet of Things and wireless sensor networks," *Sensors*, vol. 22, no. 15, p. 5546, Jul. 2022, doi: 10.3390/s22155546.

[21] A. Whichi, M. Weber, I. Ketata, S. Sahnoun, and F. Derbel, "Simulation of wireless sensor nodes based on wake-up receivers," in *Proc. 18th Int. Multi-Conference Syst., Signals Devices (SSD)*, Mar. 2021, pp. 235–240, doi: 10.1109/SSD52085.2021.9429306.

[22] T. R. Murgod and S. M. Sundaram, "A comparative study of different network simulation tools and experimentation platforms for underwater communication," *Bull. Electr. Eng. Informat.*, vol. 10, no. 2, pp. 879–885, Apr. 2021, doi: 10.11591/eei.v10i2.1466.

[23] E. Gamess and D. Thornton, "A survey of wireless network simulation and/or emulation software for use in higher education," in *Proc. ACM Southeast Conf.*, Apr. 2021, pp. 63–70, doi: 10.1145/3409334.3452066.

[24] D. Xie, J. Li, and H. Gao, "Comparison and analysis of simulation methods for TSN performance," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 768, no. 5, Mar. 2020, Art. no. 052061, doi: 10.1088/1757-899x/768/5/052061.

[25] L. Campanile, M. Gribaudo, M. Iacono, F. Marulli, and M. Mastroianni, "Computer network simulation with ns-3: A systematic literature review," *Electronics*, vol. 9, no. 2, p. 272, Feb. 2020, doi: 10.3390/electronics9020272.

[26] M. Cuzme-Rodríguez, F. Umaquinga-Criollo, A. Suárez-Zambrano, L. Farinango-Endara, H. Domínguez-Limaico, and H. Mediavilla-Valverde, "Applied technologies," in *Proc. Int. Conf. Appl. Technol.* Cham, Switzerland: Springer, 2020, pp. 271–285, doi: 10.1007/978-3-030-42517-3.

[27] D. Pandey and V. Kushwaha, "Experimental tools and techniques for wireless sensor networks," *Int. J. Recent Technol. Eng.*, vol. 8, no. 3, pp. 1674–1684, Sep. 2019, doi: 10.35940/ijrte.c4441.098319.

[28] R. L. Patel, M. J. Pathak, and A. J. Nayak, "Survey on network simulators," *Int. J. Comput. Appl.*, vol. 182, no. 21, pp. 23–30, Oct. 2018, doi: 10.5120/ijca2018917974.

[29] C. Lopez-Pavon, S. Sendra, and J. F. Valenzuela-Valdes, "Evaluation of CupCarbon network simulator for wireless sensor networks," *Netw. Protocols Algorithms*, vol. 10, no. 2, p. 1, Jun. 2018, doi: 10.5296/npa.v10i2.13201.

[30] M. Saidallah, A. El Fergougui, and A. E. Elalaoui, "A survey and comparative study of open-source wireless sensor network simulators," *Int. J. Adv. Res. Comput. Sci.*, vol. 7, no. 3, pp. 1–7, 2017.

[31] M. Sharif and A. Sadeghi-Niaraki, "Ubiquitous sensor network simulation and emulation environments: A survey," *J. Netw. Comput. Appl.*, vol. 93, pp. 150–181, Sep. 2017, doi: 10.1016/j.jnca.2017.05.009.

[32] A. Mouiz, A. Badri, A. Baghdad, and A. Sahel, "Analysis of modeling performance and simulation tools for wireless sensor networks," *Int. J. Comput. Appl. Technol. Res.*, vol. 6, no. 1, pp. 9–12, Jan. 2017, doi: 10.7753/ijcatr0601.1002.

[33] M. L. Rajaram, E. Kougianos, S. P. Mohanty, and U. Choppali, "Wireless sensor network simulation frameworks: A tutorial review: MATLAB/Simulink bests the rest," *IEEE Consum. Electron. Mag.*, vol. 5, no. 2, pp. 63–69, Apr. 2016, doi: 10.1109/MCE.2016.2519051.

[34] I. Minakov, R. Passerone, A. Rizzardi, and S. Sicari, "A comparative study of recent wireless sensor network simulators," *ACM Trans. Sensor Netw.*, vol. 12, no. 3, pp. 1–39, Aug. 2016, doi: 10.1145/2903144.

[35] A. Nayyar and R. Singh, "A comprehensive review of simulation tools for wireless sensor networks (WSNs)," *J. Wireless Netw. Commun.*, vols. 19–47, no. 1, pp. 19–47, 2015.

[36] M. Živković, B. Nlkolić, J. Protić, and A. Popović, "A survey and classifcation of wireless sensor networks simulators based on the domain of use," *Ad-Hoc Sensors Wireless Netw.*, vol. 20, nos. 3–4, pp. 245–287, 2014.

[37] M. P. Chhimwal, "Comparison between different wireless sensor simulation tool," *IOSR J. Electron. Commun. Eng.*, vol. 5, no. 2, pp. 54–60, 2013, doi: 10.9790/2834-0525460.

[38] A. U. R. Khana, S. M. Bilalb, and M. Othmana, "A performance comparison of network simulators for wireless networks," 2013, *arXiv:1307.4129*.

[39] K. Lahmar, R. Cheour, and M. Abid, "Wireless sensor networks: Trends, power consumption and simulators," in *Proc. 6th Asia Model. Symp.*, May 2012, pp. 200–204, doi: 10.1109/AMS.2012.50.

[40] M. Z. Khan, B. Askwith, F. Bouhafs, and M. Asim, "Limitations of simulation tools for large-scale wireless sensor networks," in *Proc. IEEE Workshops Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2011, pp. 820–825, doi: 10.1109/WAINA.2011.59.

[41] M. Imran, A. M. Said, and H. Hasbullah, "A survey of simulators, emulators and testbeds for wireless sensor networks," in *Proc. Int. Symp. Inf. Technol.*, vol. 2, Jun. 2010, pp. 897–902, doi: 10.1109/ITSIM.2010.5561571.

[42] S. Mehta, N. Ullah, Md. H. Kabir, Mst. N. Sultana, and K. S. Kwak, "A case study of networks simulation tools for wireless networks," in *Proc. 3rd Asia Int. Conf. Model. Simul.*, May 2009, pp. 661–666, doi: 10.1109/AMS.2009.17.

[43] D. Hauweele and B. Quoitin, "Toward accurate clock drift modeling in wireless sensor networks simulation," in *Proc. 22nd Int. ACM Conf. Model., Anal. Simul. Wireless Mobile Syst.*, Nov. 2019, pp. 95–102, doi: 10.1145/3345768.3355911.

[44] K. A. Ngo, T. T. Huynh, and D. T. Huynh, "Simulation wireless sensor networks in Castalia," in *Proc. Int. Conf. Intell. Inf. Technol.*, Feb. 2018, pp. 39–44, doi: 10.1145/3193063.3193066.

[45] F. Chiacchio, J. I. Aizpurua, L. Compagno, and D. D'Urso, "SHyFTOO, an object-oriented Monte Carlo simulation library for the modeling of stochastic hybrid fault tree automaton," *Expert Syst. Appl.*, vol. 146, May 2020, Art. no. 113139, doi: 10.1016/j.eswa.2019.113139.

[46] X. Huang, "A data-driven WSN security threat analysis model based on cognitive computing," *J. Sensors*, vol. 2022, pp. 1–10, Jan. 2022, doi: 10.1155/2022/5013905.

[47] P. Gawłowicz and A. Zubow, "Ns-3 meets OpenAI gym," in *Proc. 22nd Int. ACM Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, Nov. 2019, pp. 113–120, doi: 10.1145/3345768.3355908.

[48] A. Mosaif and S. Rakrak, "Visual node module: An open-source extension to the ns-3 network simulator," *Int. J. Commun. Syst.*, vol. 33, no. 12, pp. 1–18, Aug. 2020, doi: 10.1002/dac.4451.

[49] A. Siddik, J. A. Nitu, N. Islam, A. A. Hasi, J. Ferdous, M. Rahman, and N. Sultan, "Effects of mac parameters on the performance of IEEE 802.11 DCF in NS-3," *Int. J. Wireless Mobile Netw.*, vol. 13, no. 6, pp. 1–20, Dec. 2021, doi: 10.5121/ijwmn.2021.13601.

[50] M. K. Rana, B. Sardar, S. Mandal, and D. Saha, "Implementation and performance evaluation of a mobile IPv6 (MIPv6) simulation model for NS-3," *Simul. Model. Pract. Theory*, vol. 72, pp. 1–22, Mar. 2017, doi: 10.1016/j.simpat.2016.12.005.

[51] J. P. Rohrer and A. N. Mauldin, "Implementation of epidemic routing with IP convergence layer in ns-3," in *Proc. 10th Workshop NS-3*, 2018, pp. 69–76, doi: 10.1145/3199902.3199907.

[52] J. P. A. León, C. L. D. Santos, A. M. Mezher, J. C. Barrera, J. Meng, and E. C. Guerra, "Exploring the potential, limitations, and future directions of wireless technologies in smart grid networks: A comparative analysis," *Comput. Netw.*, vol. 235, Nov. 2023, Art. no. 109956, doi: 10.1016/j.comnet.2023.109956.

[53] Y. Teekaraman, H. Manoharan, R. Kuppusamy, S. Urooj, and F. Alrowais, "Energy efficient multi-hop routing protocol for smart vehicle monitoring using intelligent sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 17, no. 12, Dec. 2021, Art. no. 155014772110391, doi: 10.1177/15501477211039134.

[54] P. Biswas and T. Samanta, "True event-driven and fault-tolerant routing in wireless sensor network," *Wireless Pers. Commun.*, vol. 112, no. 1, pp. 439–461, May 2020, doi: 10.1007/s11277-020-07037-3.

[55] K. Jaiswal and V. Anand, "EOMR: An energy-efficient optimal multi-path routing protocol to improve QoS in wireless sensor network for IoT applications," *Wireless Pers. Commun.*, vol. 111, no. 4, pp. 2493–2515, Apr. 2020, doi: 10.1007/s11277-019-07000-x.

[56] T. Wang, X. Wang, Z. Cui, Y. Cao, and C. Suthaputchakun, "Survey on cooperatively V2X downloading for intelligent transport systems," *IET Intell. Transp. Syst.*, vol. 13, no. 1, pp. 13–21, Jan. 2019, doi: 10.1049/iet-its.2018.5104.

[57] B. Pavkovic, M. Sandic, and N. Teslic, "A genetic simulation strategy: Application to single-fault analysis of TTEthernet synchronization protocol," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102169, doi: 10.1016/j.sysarc.2021.102169.

[58] A. Varga, "A practical introduction to the OMNeT++ simulation framework," in *Recent Advances in Network Simulation: The OMNeT++ Environment and Its Ecosystem*. Cham, Switzerland: Springer, 2019, pp. 3–51.

[59] T. Wang, A. Hussain, M. N. M. Bhutta, and Y. Cao, "Enabling bidirectional traffic mobility for ITS simulation in smart city environments," *Future Gener. Comput. Syst.*, vol. 92, pp. 342–356, Mar. 2019, doi: 10.1016/j.future.2018.10.015.

[60] J. S. Gill, M. Saeedi Velashani, J. Wolf, J. Kenney, M. R. Manesh, and N. Kaabouch, "Simulation testbeds and frameworks for UAV performance evaluation," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, May 2021, pp. 335–341, doi: 10.1109/EIT51626.2021.9491882.

[61] J. S. Weber, M. Neves, and T. Ferreto, "VANET simulators: An updated review," *J. Brazilian Comput. Soc.*, vol. 27, no. 1, p. 8, Dec. 2021, doi: 10.1186/s13173-021-00113-x.

[62] M. B. M. Kamel, P. Ligeti, A. Nagy, and C. Reich, "Distributed address table (DAT): A decentralized model for end-to-end communication in IoT," *Peer-Peer Netw. Appl.*, vol. 15, no. 1, pp. 178–193, Jan. 2022, doi: 10.1007/s12083-021-01221-3.

[63] B. Otero, E. Rodríguez, O. Rojas, J. Verdú, J. J. Costa, M. A. Pajuelo, and R. Canal, "A cost-efficient QoS-aware analytical model of future software content delivery networks," *Int. J. Netw. Manage.*, vol. 31, no. 4, pp. 1–24, Jul. 2021, doi: 10.1002/nem.2137.

[64] E. Kashani, "Investigation of LTE network and removal of unwanted Handover in the network and how to implement it with the NS-3 simulator," *J. Res. Sci. Eng. Technol.*, vol. 8, no. 2, pp. 1–6, Sep. 2020, doi: 10.24200/jmas.vol8iss3pp43-49.

[65] H. Kurdi, B. Alshayban, L. Altoaimy, and S. Alsalamah, "TrustyFeer: A subjective logic trust model for smart city peer-to-peer federated clouds," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/1073216.

[66] J. Li, J. Yao, and D. Huang, "Ethernet-based avionic databus and time-space partition switch design," *J. Commun. Netw.*, vol. 17, no. 3, pp. 286–295, Jun. 2015, doi: 10.1109/JCN.2015.000051.

[67] T. Zhou, C. Yang, H. Chen, Y. Han, W. Bao, and Q. Cheng, "Performance research on ZigBee wireless sensor network self-organizing network for 220 kV four-circuit transmission lines on the same tower," *Sustain. Energy Technol. Assessments*, vol. 53, Oct. 2022, Art. no. 102302, doi: 10.1016/j.seta.2022.102302.

[68] L. J. Pérez and J. Salvachúa, "Simulation of scalability in cloud-based IoT reactive systems leveraged on a WSAN simulator and cloud computing technologies," *Appl. Sci.*, vol. 11, no. 4, p. 1804, Feb. 2021, doi: 10.3390/app11041804.

[69] N. Tampouratzis, P. Mousouliotis, and I. Papaefstathiou, "A novel integrated simulation framework for cyber-physical systems modelling," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 10, pp. 2684–2698, Oct. 2023, doi: 10.1109/TPDS.2023.3300081.

[70] M. R. S. Marques, L. Brisolara, P. R. Ferreira, and L. S. Indrusiak, "Eboracum: An extensible framework for high-level modeling and evaluation of reactive and adaptable WSNs," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–8, doi: 10.1109/ETFA.2016.7733569.

[71] R. Kaur, E. G. Singh, A. Kumar, and S. Kour, "A review study of vanet, mobility models and traffic generator tools," in *Proc. 5th Int. Conf. Contemp. Comput. Informat.*, Dec. 2022, pp. 1055–1060, doi: 10.1109/IC3I56241.2022.10072874.

[72] T. O. Olasupo and C. E. Otero, "A framework for optimizing the deployment of wireless sensor networks," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 1105–1118, Sep. 2018, doi: 10.1109/TNSM.2018.2851925.

[73] K. Kotis, S. Stavrinos, and C. Kalloniatis, "Review on semantic modeling and simulation of cybersecurity and interoperability on the Internet of Underwater Things," *Future Internet*, vol. 15, no. 1, p. 11, Dec. 2022.

[74] J. A. Delport, K. Jackman, P. L. Roux, and C. J. Fourie, "JoSIM—Superconductor SPICE simulator," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, pp. 1–5, Aug. 2019, doi: 10.1109/TASC.2019.2897312.

[75] N. Cao, P. Liu, G. Li, C. Zhang, S. Cao, G. Cao, M. Yan, and B. B. Gupta, "Evaluation models for the nearest closer routing protocol in wireless sensor networks," *IEEE Access*, vol. 6, pp. 77043–77054, 2018, doi: 10.1109/ACCESS.2018.2825441.

[76] Q. He, A. Rezaei, and S. Pursiainen, "Zeffiro user interface for electromagnetic brain imaging: A GPU accelerated FEM tool for forward and inverse computations in MATLAB," *Neuroinformatics*, vol. 18, no. 2, pp. 237–250, Apr. 2020, doi: 10.1007/s12021-019-09436-9.

[77] T. D. Hasseler, A. Ramachandran, W. A. Tarpeh, M. Stadermann, and J. G. Santiago, "Process design tools and techno-economic analysis for capacitive deionization," *Water Res.*, vol. 183, Sep. 2020, Art. no. 116034, doi: 10.1016/j.watres.2020.116034.

[78] R. Shavelis and K. Ozols, "Bluetooth low energy wireless sensor network library in MATLAB simulink," *J. Sensor Actuator Netw.*, vol. 9, no. 3, p. 38, Aug. 2020, doi: 10.3390/jsan9030038.

[79] N. Sharma, B. M. Singh, and K. Singh, "QoS-based energy-efficient protocols for wireless sensor network," *Sustain. Comput., Informat. Syst.*, vol. 30, Jun. 2021, Art. no. 100425, doi: 10.1016/j.suscom.2020.100425.

[80] J. López-Luna, L. E. Ramírez-Montes, S. Martinez-Vargas, A. I. Martínez, O. F. Mijangos-Ricardez, M. D. C. A. González-Chávez, R. Carrillo-González, F. A. Solís-Domínguez, M. D. C. Cuevas-Díaz, and V. Vázquez-Hipólito, "Linear and nonlinear kinetic and isotherm adsorption models for arsenic removal by manganese ferrite nanoparticles," *Social Netw. Appl. Sci.*, vol. 1, no. 8, p. 950, Aug. 2019, doi: 10.1007/s42452-019-0977-3.

[81] L. Hong, H. Guo, J. Liu, and Y. Zhang, "Toward swarm coordination: Topology-aware inter-UAV routing optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10177–10187, Sep. 2020.

[82] B. Patel and P. Shah, "Simulation, modelling and packet sniffing facilities for IoT: A systematic analysis," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 3, p. 2755, Jun. 2020, doi: 10.11591/ijece.v10i3.pp2755-2762.

[83] A. Rajeswari, "Simulation and performance analysis of CIT college campus network for realistic traffic scenarios using NETSIM," *Proc. Comput. Sci.*, vol. 171, pp. 2635–2644, Jan. 2020, doi: 10.1016/j.procs.2020.04.286.

[84] P. Thakur and G. Singh, "Potential simulation frameworks and challenges for Internet of Vehicles networks," in *Proc. Int. Conf. Artif. Intell., Big Data, Comput. Data Commun. Syst. (icABCD)*, Aug. 2020, pp. 1–6, doi: 10.1109/icABCD49160.2020.9183840.

[85] H. Kavak, J. J. Padilla, D. Vernon-Bido, S. Y. Diallo, R. Gore, and S. Shetty, "Simulation for cybersecurity: State of the art and future directions," *J. Cybersecur.*, vol. 7, no. 1, pp. 1–13, Feb. 2021, doi: 10.1093/cybsec/tyab005.

[86] A. Parizad, H. R. Baghaee, M. E. Iranian, G. B. Gharehpetian, and J. M. Guerrero, "Real-time simulator and offline/online closed-loop test bed for power system modeling and development," *Int. J. Electr. Power Energy Syst.*, vol. 122, Nov. 2020, Art. no. 106203, doi: 10.1016/j.ijepes.2020.106203.

[87] A. Farooq and X. Wu, "Review of edutainment immersive visualization (IV) development tools for simulating renewable energy systems (RESs)," *Energy Strategy Rev.*, vol. 44, Nov. 2022, Art. no. 101000, doi: 10.1016/j.esr.2022.101000.

[88] F. Idris, N. Hashim, A. F. Kadmin, and L. B. Yee, "Intelligent fire detection and alert system using labVIEW," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 3, p. 1842, Jun. 2019, doi: 10.11591/ijece.v9i3.pp1842-1849.

[89] Y. Liu, A. Dowling, C. Page, and W. Smith, "Building a mininet-based fiber optic simulator," in *Proc. 30th Wireless Opt. Commun. Conf. (WOCC)*, Oct. 2021, pp. 279–283, doi: 10.1109/WOCC53213.2021.9603202.

[90] N. Gupta, M. S. Maashi, S. Tanwar, S. Badotra, M. Aljebreen, and S. Bharany, "A comparative study of software defined networking controllers using mininet," *Electronics*, vol. 11, no. 17, p. 2715, Aug. 2022, doi: 10.3390/electronics11172715.

[91] S. Buzura, V. Dadarlat, A. Peculea, H. Bertrand, and R. Chevalier, "Simulation framework for 6LoWPAN networks using mininet-WiFi," in *Proc. IEEE Int. Conf. Autom., Quality Test., Robot. (AQTR)*, May 2022, pp. 1–5, doi: 10.1109/AQTR55203.2022.9802017.

[92] P. Kumar, A. Baliyan, K. R. Prasad, N. Sreekanth, P. Jawarkar, V. Roy, and E. T. Amoatey, "Machine learning enabled techniques for protecting wireless sensor networks by estimating attack prevalence and device deployment strategy for 5G networks," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–15, Apr. 2022, doi: 10.1155/2022/5713092.

[93] H. Yahyaoui, M. Majdoub, M. F. Zhani, and M. Aloqaily, "On minimizing TCP retransmission delay in softwarized networks," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2022, pp. 1–6, doi: 10.1109/NOMS54207.2022.9789762.

[94] A. Al-Roubaiey and H. Al-Jamimi, "Online power tossim simulator for wireless sensor networks," in *Proc. 11th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jun. 2019, pp. 1–5, doi: 10.1109/ECAI46879.2019.9042005.

[95] Y. Yigit, V. K. Akram, and O. Dagdeviren, "Breadth-first search tree integrated vertex cover algorithms for link monitoring and routing in wireless sensor networks," *Comput. Netw.*, vol. 194, Jul. 2021, Art. no. 108144, doi: 10.1016/j.comnet.2021.108144.

[96] G. Krivulya, I. Skarga-Bandurova, Z. Tatarchenko, O. Seredina, M. Shcherbakova, and E. Shcherbakov, "An intelligent functional diagnostics (don't short) of wireless sensor network," in *Proc. 7th Int. Conf. Future Internet Things Cloud Workshops (FiCloudW)*, Aug. 2019, pp. 135–139, doi: 10.1109/FiCloudW.2019.00037.

[97] B. Sundaram, N. Srinivas, K. Raja, and M. Mishra, "Renewable energy sources efficient detection in triangulation for wireless sensor networks," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1055, May 2021, Art. no. 012135.

[98] M. Kaur, "Performance and scalability evaluation of the wireless body area network using Castalia simulator," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 2, pp. 543–554, Apr. 2021, doi: 10.17762/turcomat.v12i2.882.

[99] G. Chen, W. Dong, F. Qiu, G. Guan, Y. Gao, and S. Zeng, "Scalable and interactive simulation for IoT applications with TinySim," *IEEE Internet Things J.*, vol. 10, no. 23, pp. 20984–20999, Dec. 2023, doi: 10.1109/JIOT.2023.3285244.

[100] Y. Abdul-Wahab, A.-B. Alhassan, and A.-M. Salifu, "Extending the lifespan of wireless sensor networks: A survey of LEACH and non-LEACH routing protocols," *Int. J. Comput. Appl.*, vol. 177, no. 45, pp. 6–21, Mar. 2020, doi: 10.5120/ijca2020919447.

[101] R. Ruslan, A. S. M. Zailani, N. H. M. Zukri, N. K. Kamarudin, S. J. Elias, and R. B. Ahmad, "Routing performance of structured overlay in distributed hash tables (DHT) for P2P," *Bull. Electr. Eng. Informat.*, vol. 8, no. 2, pp. 389–395, Mar. 2019, doi: 10.11591/eei.v8i2.1449.

[102] T.-Y. Lee, Y.-J. Chen, and G.-J. Horng, "Forming high-availability cloud mechanism for secure peer-servicing networks," *Wireless Pers. Commun.*, vol. 109, no. 1, pp. 361–391, Nov. 2019, doi: 10.1007/s11277-019-06569-7.

[103] P. Zhang, W. Guo, Z. Liu, M. Zhou, B. Huang, and K. Sedraoui, "Optimized blockchain sharding model based on node trust and allocation," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 3, pp. 2804–2816, Sep. 2023, doi: 10.1109/TNSM.2022.3233570.

[104] A. Mohsin, S. Aurangzeb, M. Aleem, and M. T. Khan, "On the performance and scalability of simulators for improving security and safety of smart cities," in *Proc. IEEE 27th Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2022, pp. 1–8, doi: 10.1109/ETFA52439.2022.9921600.

[105] R. Chéour, M. W. Jmal, O. Kanoun, and M. Abid, "Evaluation of simulator tools and power-aware scheduling model for wireless sensor networks," *IET Comput. Digit. Techn.*, vol. 11, no. 5, pp. 173–182, Sep. 2017, doi: 10.1049/iet-cdt.2017.0003.

[106] S. P. Fekete, A. Kroller, S. Fischer, and D. Pfisterer, "Shawn: The fast, highly customizable sensor network simulator," in *Proc. 4th Int. Conf. Networked Sens. Syst.*, Jun. 2007, p. 299, doi: 10.1109/inss.2007.4297441.

[107] M. L. Silva, L. N. S. Júnior, A. L. L. Aquino, and J. D. C. Lima, "JSensor: A parallel simulator for huge wireless sensor networks applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 10, pp. 2296–2308, Oct. 2019, doi: 10.1109/TPDS.2019.2908845.

[108] M. Kudelski, L. M. Gambardella, and G. A. Di Caro, "RoboNetSim: An integrated framework for multi-robot and network simulation," *Robot. Auto. Syst.*, vol. 61, no. 5, pp. 483–496, May 2013, doi: 10.1016/j.robot.2013.01.003.

[109] M. E. Pellenz, R. Lachowski, E. Jamhour, G. Brante, G. L. Moritz, and R. D. Souza, "In-network data aggregation for information-centric WSNs using unsupervised machine learning techniques," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Sep. 2021, pp. 1–7, doi: 10.1109/ISCC53001.2021.9631416.

[110] C. Haas, J. Wilke, and V. Stöhr, "Realistic simulation of energy consumption in wireless sensor networks," in *Proc. Eur. Conf. Wireless Sensor Netw.* Berlin, Germany: Springer, 2012, pp. 82–97.

[111] U. Meena and A. Agarwal, "Analysis of simulators used in wireless sensor network," in *Proc. 6th Int. Conf. Trends Electron. Informat. (ICOEI)*, Apr. 2022, pp. 678–683, doi: 10.1109/ICOEI53556.2022.9776877.

[112] E. Caballero, V. Ferreira, R. A. Lima, J. C. H. Soto, D. Muchaluat-Saade, and C. Albuquerque, "BNS: A framework for wireless body area network realistic simulations," *Sensors*, vol. 21, no. 16, p. 5504, Aug. 2021, doi: 10.3390/s21165504.

[113] U. Amozarrain and M. Larrea, "Using publish/subscribe for message routing in mobile environments," *Wireless Netw.*, vol. 29, no. 4, pp. 1831–1842, May 2023, doi: 10.1007/s11276-023-03233-8.

[114] M. S. Sheikh and J. Liang, "A comprehensive survey on VANET security services in traffic management system," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–23, Sep. 2019.

[115] B. Ching, M. Amoozadeh, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Enabling performance and security simulation studies of intelligent traffic signal light control with VENTOS-HIL," *Veh. Commun.*, vol. 24, Aug. 2020, Art. no. 100230, doi: 10.1016/j.vehcom.2020.100230.

[116] M. Amoozadeh, B. Ching, C.-N. Chuah, D. Ghosal, and H. M. Zhang, "VENTOS: Vehicular network open simulator with hardware-in-the-loop support," *Proc. Comput. Sci.*, vol. 151, pp. 61–68, Jan. 2019.

[117] B. Feroz, A. Mehmood, H. Maryam, S. Zeadally, C. Maple, and M. A. Shah, "Vehicle-life interaction in fog-enabled smart connected and autonomous vehicles," *IEEE Access*, vol. 9, pp. 7402–7420, 2021, doi: 10.1109/ACCESS.2020.3049110.

[118] M. Scazzariello, L. Ariemma, G. Di Battista, and M. Patrignani, "Megalos: A scalable architecture for the virtualization of large network scenarios," *Future Internet*, vol. 13, no. 9, p. 227, Aug. 2021, doi: 10.3390/fi13090227.

[119] N. Dharini, N. Duraipandian, and J. Katiravan, "ELPC-trust framework for wireless sensor networks," *Wireless Pers. Commun.*, vol. 113, pp. 1709–1742, Aug. 2020, doi: 10.1007/s11277-020-07288-0.

[120] V. K. Venkatesan, I. Izonin, J. Periyasamy, A. Indirajithu, A. Batyuk, and M. T. Ramakrishna, "Incorporation of energy efficient computational strategies for clustering and routing in heterogeneous networks of smart city," *Energies*, vol. 15, no. 20, p. 7524, Oct. 2022, doi: 10.3390/en15207524.

[121] H. Huang, H. Yin, G. Min, J. Zhang, Y. Wu, and X. Zhang, "Energy-aware dual-path geographic routing to bypass routing holes in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1339–1352, Jun. 2018, doi: 10.1109/TMC.2017.2771424.

[122] D. Jabba and P. Acevedo, "ViTool-BC: Visualization tool based on cooja simulator for WSN," *Appl. Sci.*, vol. 11, no. 16, p. 7665, Aug. 2021, doi: 10.3390/app11167665.

[123] L. Shu, Y. Zhang, L. T. Yang, Y. Wang, M. Hauswirth, and N. Xiong, "TPGF: Geographic routing in wireless multimedia sensor networks," *Telecommun. Syst.*, vol. 44, nos. 1–2, pp. 79–95, Jun. 2010, doi: 10.1007/s11235-009-9227-0.

[124] E. N. Kaur and I. K. Aulakh, "Evaluation and implementation of cluster head selection in WSN using Contiki/Cooja simulator," *J. Statist. Manage. Syst.*, vol. 23, no. 2, pp. 407–418, Feb. 2020, doi: 10.1080/09720510.2020.1736324.

[125] Y. Tanaka, T. Ito, and F. Teraoka, "6TiSCH scheduling function design suite founded on Contiki-NG," *J. Inf. Process.*, vol. 30, no. 10, pp. 669–678, 2022, doi: 10.2197/ipsjjip.30.669.

[126] A. Seyfollahi and A. Ghaffari, "A lightweight load balancing and route minimizing solution for routing protocol for low-power and lossy networks," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107368, doi: 10.1016/j.comnet.2020.107368.

[127] Z. Padrah, A. Pastrav, T. Palade, O. Ratiu, and E. Puschita, "Development and validation of an ISA100.11a simulation model for accurate industrial WSN planning and deployment," *Sensors*, vol. 21, no. 11, p. 3600, May 2021, doi: 10.3390/s21113600.

[128] S. M. Nam and H. J. Kim, "WSN-SES/MB: System entity structure and model base framework for large-scale wireless sensor networks," *Sensors*, vol. 21, no. 2, p. 430, Jan. 2021, doi: 10.3390/s21020430.

[129] W. Han, G. Wang, and A. M. Stankovic, "Active disturbance rejection control in fully distributed automatic generation control with co-simulation of communication delay," *Control Eng. Pract.*, vol. 85, pp. 225–234, Apr. 2019, doi: 10.1016/j.conengprac.2019.01.016.

[130] G. C. Marchesan, E. A. Carara, C. Müller, and L. L. de Oliveira, "GCoL—A general co-simulator applied to wireless sensor networks and RTL design," in *Proc. 17th IEEE Int. New Circuits Syst. Conf. (NEWCAS)*, Jun. 2019, pp. 1–4, doi: 10.1109/NEWCAS44328.2019.8961220.

[131] A. Rady, E. L. M. El-Rabaie, M. Shokair, and N. Abdel-Salam, "Comprehensive survey of routing protocols for mobile wireless sensor networks," *Int. J. Commun. Syst.*, vol. 34, no. 15, pp. 1–10, Oct. 2021, doi: 10.1002/dac.4942.

[132] M. Baghouri, S. Chakkor, Z. Cheker, and A. E. Oualkadi, "Real-time implementation and evaluation of the LEACH protocol by PROWLER simulator," in *Proc. Int. Conf. Innov. Intell. Informat., Comput., Technol. (3ICT)*, Nov. 2022, pp. 387–392, doi: 10.1109/3ICT56508.2022.9990868.

[133] Q. Tang, J. Li, D. Jin, T. Ngogia, M. Liu, and Y. Li, "Three-dimension visual simulation for formation tracing control of multiple underwater vehicles," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Oct. 2021, pp. 878–883, doi: 10.1109/ICUS52573.2021.9641116.

[134] X. Lin, N. Jha, M. Joshi, N. Karapetyan, Y. Aloimonos, and M. Yu, "OysterSim: Underwater simulation for enhancing oyster reef monitoring," in *Proc. OCEANS*, Oct. 2022, pp. 1–6, doi: 10.1109/OCEANS47191.2022.9977233.

[135] L. Salmon, P.-Y. Pillain, G. Guillou, and J.-P. Babau, "CARES, a framework for CPS simulation : Application to autonomous underwater vehicle navigation function," in *Proc. Forum Specification Design Lang. (FDL)*, Sep. 2021, pp. 01–08, doi: 10.1109/FDL53530.2021.9568380.

[136] G. H. Adday, S. K. Subramaniam, Z. A. Zukarnain, and N. Samian, "Friendship degree and tenth man strategy: A new method for differentiating between erroneous readings and true events in wireless sensor networks," *IEEE Access*, vol. 11, pp. 127651–127668, 2023, doi: 10.1109/ACCESS.2023.3332476.

[137] M. Ghaleb, E. Felemban, S. Subramaniam, A. A. Sheikh, and S. B. Qaisar, "A performance simulation tool for the analysis of data gathering in both terrestrial and underwater sensor networks," *IEEE Access*, vol. 5, pp. 4190–4208, 2017, doi: 10.1109/ACCESS.2017.2684539.

[138] J. A. Barriga, P. J. Clemente, E. Sosa-Sánchez, and Á. E. Prieto, "SimulateIoT: Domain specific language to design, code generation and execute IoT simulation environments," *IEEE Access*, vol. 9, pp. 92531–92552, 2021, doi: 10.1109/ACCESS.2021.3092528.

[139] I. S. Gavrilov, "Development of a wireless sensor network simulator using energy-efficient traffic routing methods," *Bull. Transilvania Univ. Brasov. III, Math. Comput. Sci.*, vol. 2023, pp. 211–228, Jul. 2023, doi: 10.31926/but.mif.2023.3.65.1.16.

[140] U. D. Ani, J. M. Watson, M. Carr, A. Cook, and J. R. Nurse, "A review of the use and utility of industrial network-based open source simulators: Functionality, security, and policy viewpoints," *J. Defense Model. Simul., Appl., Methodol., Technol.*, vol. 19, no. 3, pp. 263–286, Jul. 2022, doi: 10.1177/1548512920953499.

[141] E. Aghajani, C. Nagy, O. L. Vega-Márquez, M. Linares-Vásquez, L. Moreno, G. Bavota, and M. Lanza, "Software documentation issues unveiled," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng. (ICSE)*, May 2019, pp. 1199–1210, doi: 10.1109/ICSE.2019.00122.

[142] E. Volnes, S. Kristiansen, and T. Plagemann, "Improving the accuracy of timing in scalable WSN simulations with communication software execution models," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107855, doi: 10.1016/j.comnet.2021.107855.

[143] G. Lugaresi, S. Gangemi, G. Gazzoni, and A. Matta, "Online validation of digital twins for manufacturing systems," *Comput. Ind.*, vol. 150, Sep. 2023, Art. no. 103942, doi: 10.1016/j.compind.2023.103942.

[144] A. M. Shantaf, S. Kurnaz, and A. H. Mohammed, "Performance evaluation of three mobile ad-hoc network routing protocols in different environments," in *Proc. Int. Congr. Hum.-Comput. Interact., Optim. Robot. Appl. (HORA)*, Jun. 2020, no. 1, pp. 1–6, doi: 10.1109/HORA49412.2020.9152845.

[145] G. H. Kumar, "Node localization algorithm for detecting malicious nodes to prevent connection failures and improve end-to-end delay," *Comput. Commun.*, vol. 190, pp. 37–47, Jun. 2022, doi: 10.1016/j.comcom.2022.04.001.

[146] R. Krishnan and G. Perumal, "H2B2H protocol for addressing link failure in WSN," *Cluster Comput.*, vol. 22, no. S4, pp. 9687–9696, Jul. 2019, doi: 10.1007/s10586-017-1355-9.

[147] Y. Huang, B. Tang, L. Deng, and C. Zhao, "Fuzzy analytic hierarchy process-based balanced topology control of wireless sensor networks for machine vibration monitoring," *IEEE Sensors J.*, vol. 20, no. 15, pp. 8256–8264, Aug. 2020, doi: 10.1109/JSEN.2020.2966049.

[148] Y. Yue, L. Cao, and Z. Luo, "Hybrid artificial bee colony algorithm for improving the coverage and connectivity of wireless sensor networks," *Wireless Pers. Commun.*, vol. 108, no. 3, pp. 1719–1732, Oct. 2019, doi: 10.1007/s11277-019-06492-x.

[149] H. Alwan and A. Agarwal, "MQoSR: A multiobjective QoS routing protocol for wireless sensor networks," *ISRN Sensor Netw.*, vol. 2013, pp. 1–12, Jun. 2013, doi: 10.1155/2013/495803.

[150] A. M. Viswa Bharathy and V. Chandrasekar, "A novel virtual tunneling protocol for underwater wireless sensor networks," in *Soft Computing and Signal Processing* (Advances in Intelligent Systems and Computing), vol. 900. Singapore: Springer, 2019, pp. 281–289.

[151] L. J. Chien, M. Drieberg, P. Sebastian, and L. H. Hiung, "A simple solar energy harvester for wireless sensor networks," in *Proc. 6th Int. Conf. Intell. Adv. Syst. (ICIAS)*, Aug. 2016, pp. 1–6, doi: 10.1109/ICIAS.2016.7824104.

[152] A. Kurniawan, P. Kristalina, and M. Z. S. Hadi, "Performance analysis of routing protocols AODV, OLSR and DSDV on MANET using NS3," in *Proc. Int. Electron. Symp. (IES)*, Sep. 2020, pp. 199–206, doi: 10.1109/IES50839.2020.9231690.

[153] N. AlMansour and S. Alahmadi, "Secure ad hoc on-demand distance vector routing protocol in WSN," in *Proc. 1st Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, Apr. 2018, pp. 1–4, doi: 10.1109/CAIS.2018.8441991.

[154] S. Khriji, R. Cheour, M. Goetz, D. E. Houssaini, I. Kammoun, and O. Kanoun, "Measuring energy consumption of a wireless sensor node during transmission: PanStamp," in *Proc. IEEE 32nd Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, May 2018, pp. 274–280, doi: 10.1109/AINA.2018.00050.

[155] W. A. Aliady and S. A. Al-Ahmadi, "Energy preserving secure measure against wormhole attack in wireless sensor networks," *IEEE Access*, vol. 7, pp. 84132–84141, 2019, doi: 10.1109/ACCESS.2019.2924283.

[156] F. Wu, X. Li, A. K. Sangaiah, L. Xu, S. Kumari, L. Wu, and J. Shen, "A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 727–737, May 2018.

[157] M. B. M. Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, Jan. 2019.

[158] G. Kasturi, A. Jain, and J. Singh, "Detection and classification of radio frequency jamming attacks using machine learning," *J. Wireless Mob. Netw. Ubiquitous Comput. Dependable Appl.*, vol. 11, no. 4, pp. 49–62, 2020.

[159] F. Palmese, A. E. C. Redondi, and M. Cesana, "Adaptive quality of service control for MQTT-SN," *Sensors*, vol. 22, no. 22, p. 8852, Nov. 2022, doi: 10.3390/s22228852.

[160] N. Islam, M. I. Hossain, and A. Rahman, "A comprehensive analysis of quality of service (QoS) in ZigBee network through mobile and fixed node," *J. Comput. Commun.*, vol. 10, no. 3, pp. 86–99, 2022, doi: 10.4236/jcc.2022.103006.

[161] P. Movva and P. T. Rao, "Novel two-fold data aggregation and MAC scheduling to support energy efficient routing in wireless sensor network," *IEEE Access*, vol. 7, pp. 1260–1274, 2019, doi: 10.1109/ACCESS.2018.2888484.

[162] W. Quan, N. Cheng, M. Qin, H. Zhang, H. A. Chan, and X. Shen, "Adaptive transmission control for software defined vehicular networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 653–656, Jun. 2019, doi: 10.1109/LWC.2018.2879514.

[163] A. H. Allam, M. Taha, and H. H. Zayed, "Enhanced zone-based energy aware data collection protocol for WSNs (E-ZEAL)," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 2, pp. 36–46, Feb. 2022, doi: 10.1016/j.jksuci.2019.10.012.

[164] M. Z. Ghawy, G. A. Amran, H. AlSalman, E. Ghaleb, J. Khan, A. A. Al-Bakhrani, A. M. Alziadi, A. Ali, and S. S. Ullah, "An effective wireless sensor network routing protocol based on particle swarm optimization algorithm," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–13, May 2022.

[165] U. S. Verma and N. Gupta, "Wireless sensor network path optimization using sensor node coverage area calculation approach," *Wireless Pers. Commun.*, vol. 116, no. 1, pp. 91–103, Jan. 2021, doi: 10.1007/s11277-020-07706-3.

[166] H. Abdulrab, F. A. Hussin, A. A. Aziz, A. Awang, I. Ismail, and P. A. M. Devan, "Reliable fault tolerant-based multipath routing model for industrial wireless control systems," *Appl. Sci.*, vol. 12, no. 2, p. 544, Jan. 2022, doi: 10.3390/app12020544.

[167] K. Fathima Shemim and U. Witkowski, "Energy efficient clustering protocols for WSN: Performance analysis of FL-EE-NC with LEACH, k means-LEACH, LEACH-FL and FL-EE/D using NS-2," in *Proc. 32nd Int. Conf. Microelectron. (ICM)*, Dec. 2020, pp. 1–5, doi: 10.1109/ICM50269.2020.9331768.

[168] R. Fotohi and S. Firoozi Bari, "A novel countermeasure technique to protect WSN against denial-of-sleep attacks using firefly and Hopfield neural network (HNN) algorithms," *J. Supercomput.*, vol. 76, no. 9, pp. 6860–6886, Sep. 2020, doi: 10.1007/s11227-019-03131-x.

[169] K. Karthigadevi, S. Balamurali, and M. Venkatesulu, "Based on neighbor density estimation technique to improve the quality of service and to detect and prevent the sinkhole attack in wireless sensor network," in *Proc. IEEE Int. Conf. Intell. Techn. Control, Optim. Signal Process. (INCOS)*, Apr. 2019, pp. 1–4, doi: 10.1109/INCOS45849.2019.8951406.

[170] M. Pundir and J. K. Sandhu, "A systematic review of quality of service in wireless sensor networks using machine learning: Recent trend and future vision," *J. Netw. Comput. Appl.*, vol. 188, Aug. 2021, Art. no. 103084, doi: 10.1016/j.jnca.2021.103084.

[171] S. Sinha and A. Paul, "A novel neuro-fuzzy-based localisation system for WSN using node proximity," *Int. J. Internet Protocol Technol.*, vol. 14, no. 1, p. 49, 2021, doi: 10.1504/ijipt.2021.113908.

[172] H. Alazzam and W. Almobaideen, "Enhancing the lifetime of wireless sensor network using genetic algorithm," in *Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS)*, Jun. 2019, pp. 25–29, doi: 10.1109/IACS.2019.8809109.

[173] Y. H. Robinson, R. S. Krishnan, K. L. Narayanan, Y. Reeginal, G. V. Rajkumar, and P. S. R. Malar, "Hybrid data forwarding technique for enhanced lifetime in wireless sensor networks," in *Proc. 5th Int. Conf. Trends Electron. Informat. (ICOEI)*, Jun. 2021, pp. 657–660.

[174] N. T. T. Hang, N. C. Trinh, and N. T. Ban, "Energy aware event driven routing protocol and dynamic delivering scheme for multievent wireless sensor network," in *Proc. 2nd Int. Conf. Recent Adv. Signal Process., Telecommun. Comput. (SigTelCom)*, Jan. 2018, pp. 224–229, doi: 10.1109/SIGTELCOM.2018.8325795.

[175] C. Boucetta, O. Flauzac, B. S. Haggar, A. M. Nassour, and F. Nolot, "Smart grid networks enabled electricity-theft detection and fault tolerance," in *Proc. 8th Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2020, pp. 1–6, doi: 10.1109/WINCOM50532.2020.9272446.

[176] M. El Fissaoui, A. Beni-Hssane, and M. Saadi, "Energy efficient and fault tolerant distributed algorithm for data aggregation in wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 2, pp. 569–578, Feb. 2019, doi: 10.1007/s12652-018-0704-8.

[177] M. K. Baazaoui, I. Ketata, A. Fakhfakh, and F. Derbel, "Modeling of packet error rate distribution based on received signal strength indications in OMNeT++ for wake-up receivers," *Sensors*, vol. 23, no. 5, p. 2394, Feb. 2023, doi: 10.3390/s23052394.

[178] S. A. Abdulzahra, A. K. M. Al-Qurabat, and A. K. Idrees, "Compression-based data reduction technique for IoT sensor networks," *Baghdad Sci. J.*, vol. 18, no. 1, p. 0184, Mar. 2021, doi: 10.21123/bsj.2021.18.1.0184.

[179] M. Baazaoui, I. Ketata, G. Fersi, A. Fakhfakh, and F. Derbel, "Implementation of RSSI module in OMNeT++ for investigation of WSN simulations based on real environmental conditions," in *Proc. 11th Int. Conf. Sensor Netw.*, 2022, pp. 281–287, doi: 10.5220/0011012600003118.

[180] M. A. Merzoug, A. Boukerche, A. Mostefaoui, and S. Chouali, "Spreading aggregation: A distributed collision-free approach for data aggregation in large-scale wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 125, pp. 121–134, Mar. 2019, doi: 10.1016/j.jpdc.2018.11.007.

[181] I. D. I. Saeedi and A. K. M. Al-Qurabat, "Perceptually important points-based data aggregation method for wireless sensor networks," *Baghdad Sci. J.*, vol. 19, no. 4, p. 0875, Aug. 2022, doi: 10.21123/bsj.2022.19.4.0875.

[182] R. Azimi and H. Sajedi, "A decentralized gossip based approach for data clustering in peer-to-peer networks," *J. Parallel Distrib. Comput.*, vol. 119, pp. 64–80, Sep. 2018, doi: 10.1016/j.jpdc.2018.03.009.

[183] P. S. Uma Priyadarsini and P. Sriramya, "Disaster management using evidence-based interactive trust management system for wireless sensor networks by Internet of Things," *Comput. Electr. Eng.*, vol. 75, pp. 164–174, May 2019, doi: 10.1016/j.compeleceng.2019.02.020.

[184] M. B. M. Kamel, P. Ligeti, and C. Reich, "Lamred: Location-aware and privacy preserving multi-layer resource discovery for IoT," *Acta Cybernetica*, vol. 25, no. 2, pp. 319–349, Aug. 2021, doi: 10.14232/acta-cyb.289938.

[185] A. M. Almajidi and V. P. Pawar, "A new system model for sensor node validation by using OPNET," *Wireless Pers. Commun.*, vol. 108, no. 4, pp. 2389–2401, Oct. 2019, doi: 10.1007/s11277-019-06527-3.

[186] P. Mounika, "Performance analysis of wireless sensor network topologies for zigbee using riverbed modeler," in *Proc. 2nd Int. Conf. Inventive Syst. Control (ICISC)*, Jan. 2018, pp. 1456–1459, doi: 10.1109/ICISC.2018.8399050.

[187] O. Alsaif, I. Saleh, and D. Ali, "Evaluating the performance of nodes mobility for zigbee wireless sensor network," in *Proc. Int. Conf. Comput. Inf. Sci. Technol. Their Appl. (ICCISTA)*, Mar. 2019, pp. 1–5, doi: 10.1109/ICCISTA.2019.8830652.

[188] H. Zemrane, A. N. Abbou, Y. Baddi, and A. Hasbi, "Wireless sensor networks as part of IoT: Performance study of WiMax–Mobil protocol," in *Proc. 4th Int. Conf. Cloud Comput. Technol. Appl. (Cloudtech)*, Nov. 2018, pp. 1–8, doi: 10.1109/CloudTech.2018.8713351.

[189] C. Liu, C. Zhang, F. Chen, and C. Zhao, "Towards IPv6-based architecture for big data processing of community medical Internet of Things," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 1333–1338, doi: 10.1109/IWCMC.2018.8450268.

[190] G. Rajeswari and K. Murugan, "Healing of large-scale failures in WSN by the effectual placement of relay nodes," *IET Commun.*, vol. 14, no. 17, pp. 3030–3038, Oct. 2020, doi: 10.1049/iet-com.2020.0320.

[191] M. Bakni, L. M. Moreno Chacón, Y. Cardinale, G. Terrasson, and O. Curea, "WSN simulators evaluation: An approach focusing on energy awareness," *Int. J. Wireless Mobile Netw.*, vol. 11, no. 6, pp. 1–20, Dec. 2019, doi: 10.5121/ijwmn.2019.11601.

[192] K. Prabha, "Performance assessment and comparison of efficient ad hoc reactive and proactive network routing protocols," *Social Netw. Comput. Sci.*, vol. 1, no. 1, p. 13, Jan. 2020, doi: 10.1007/s42979-019-0015-y.

[193] S. Ghildiyal, D. Srivastava, S. Mall, V. Sharma, A. Gupta, M. Manish, and S. Kumar, "Performance optimization of multi-node density using random mobility model in mobile adhoc network," in *Proc. AIP Conf.*, 2023, p. 020059, doi: 10.1063/5.0152315.

[194] M. Jamshidi, M. Esnaashari, A. M. Darwesh, and M. R. Meybodi, "Using time-location tags and watchdog nodes to defend against node replication attack in mobile wireless sensor networks," *Int. J. Wireless Inf. Netw.*, vol. 27, no. 1, pp. 102–115, Mar. 2020, doi: 10.1007/s10776-019-00469-0.

[195] M. Jamshidi, S. S. A. Poor, N. N. Qader, M. Esnaashari, and R. M. Mohammad, "A lightweight algorithm against replica node attack in mobile wireless sensor networks using learning agents," *IEIE Trans. Smart Process. Comput.*, vol. 8, no. 1, pp. 58–70, Feb. 2019, doi: 10.5573/ieiespc.2019.8.1.058.

[196] Z. Alansari, N. B. Anuar, A. Kamsin, M. R. Belgaum, and S. Soomro, "Quality of service in wireless sensor networks using cellular learning automata," in *Proc. Int. Conf. Comput., Electron. Commun. Eng. (iCCECE)*, Aug. 2020, pp. 15–20, doi: 10.1109/iCCECE49321.2020.9231123.

[197] H. M. Jawad, A. M. Jawad, R. Nordin, S. K. Gharghan, N. F. Abdullah, M. Ismail, and M. J. Abu-AlShaeer, "Accurate empirical path-loss model based on particle swarm optimization for wireless sensor networks in smart agriculture," *IEEE Sensors J.*, vol. 20, no. 1, pp. 552–561, Jan. 2020, doi: 10.1109/JSEN.2019.2940186.

[198] H. Mohapatra and A. K. Rath, "Fault tolerance in WSN through PE-LEACH protocol," *IET Wireless Sensor Syst.*, vol. 9, no. 6, pp. 358–365, Dec. 2019, doi: 10.1049/iet-wss.2018.5229.

[199] S. Alsafi and S. A. Talab, "Implementation of DEEC, DDEEC, EDEEC and TDEEC protocols using MATLAB in wireless sensor network," *Int. J. Adv. Netw. Appl.*, vol. 12, no. 3, pp. 4596–4600, 2020, doi: 10.35444/ijana.2020.12304.

[200] A. Shankar, N. R. Sivakumar, M. Sivaram, A. Ambikapathy, T. K. Nguyen, and V. Dhasarathan, "Increasing fault tolerance ability and network lifetime with clustered pollination in wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 2, pp. 2285–2298, Feb. 2021, doi: 10.1007/s12652-020-02325-z.

[201] P. Singh, P. Singh, N. Mittal, U. Singh, and S. Singh, "An optimum localization approach using hybrid TSNMRA in 2D WSNs," *Comput. Netw.*, vol. 226, May 2023, Art. no. 109682, doi: 10.1016/j.comnet.2023.109682.

[202] M. K. Kumar and V. K. Prasad, "TASLT: Triangular area segmentation based localization technique for wireless sensor networks using AoA and RSSI measures—A new approach," in *Proc. IEEE 18th Int. Conf. Mobile Ad Hoc Smart Syst. (MASS)*, Oct. 2021, pp. 585–590, doi: 10.1109/MASS52906.2021.00083.

[203] N. M. Samir, M. Musni, Z. M. Hanapi, and M. R. Radzuan, "Impact of denial-of-service attack on directional compact geographic forwarding routing protocol in wireless sensor networks," *Baghdad Sci. J.*, vol. 18, no. 4, p. 1371, Dec. 2021, doi: 10.21123/bsj.2021.18.4(suppl.).1371.

[204] A. V. Jha, B. Appasani, and A. N. Ghazali, "Performance evaluation of network layer routing protocols on wireless sensor networks," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2019, pp. 1862–1865, doi: 10.1109/ICCES45898.2019.9002570.

[205] N. H. Ismail and B. Y. Mohsin, "Qualnet performances of grid-based clustering for WSN's routing protocols," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 17, no. 1, p. 448, Jan. 2020, doi: 10.11591/ijeecs.v17.i1.pp448-456.

[206] B. S. Hassen, S. A.-S. Lafta, H. M. Noman, and A. H. Ali, "Analyzing the performances of WSNs routing protocols in grid-based clustering," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 9, no. 4, p. 1211, Aug. 2019, doi: 10.18517/ijaseit.9.4.8900.

[207] A. Kaushik, S. Indu, and D. Gupta, "A grey wolf optimization approach for improving the performance of wireless sensor networks," *Wireless Pers. Commun.*, vol. 106, no. 3, pp. 1429–1449, Jun. 2019, doi: 10.1007/s11277-019-06223-2.

[208] R. K. Sundararajan and U. Arumugam, "Event detection and information passing using LEACH protocol in wireless sensor networks," *Wireless Pers. Commun.*, vol. 101, no. 3, pp. 1703–1714, Aug. 2018, doi: 10.1007/s11277-018-5785-3.

[209] H. Sharma, A. Haque, and Z. A. Jaffery, "Maximization of wireless sensor network lifetime using solar energy harvesting for smart agriculture monitoring," *Ad Hoc Netw.*, vol. 94, Nov. 2019, Art. no. 101966, doi: 10.1016/j.adhoc.2019.101966.

[210] M. Farsi, M. A. Elhosseini, M. Badawy, H. A. Ali, and H. Z. Eldin, "Deployment techniques in wireless sensor networks, coverage and connectivity: A survey," *IEEE Access*, vol. 7, pp. 28940–28954, 2019, doi: 10.1109/ACCESS.2019.2902072.

[211] J. Yun, K. Sohn, H. Y. J. Kang, and D. Lee, "Dynamic simulation on network security simulator using SSFNET," in *Proc. Int. Conf. Converg. Inf. Technol.*, Nov. 2007, pp. 1168–1172, doi: 10.1109/ICCIT.2007.373.

[212] A. Shehu and R. Kushe, "A cyber attack scenario using SSFNet," in *Proc. 14th Int. Conf. Netw.-Based Inf. Syst.*, Sep. 2011, pp. 690–693, doi: 10.1109/NBiS.2011.116.

[213] F. Abdallah, M. T. Yaseen, and Y. M. Hussien, "Portable heartbeat rate monitoring system by WSN using LABVIEW," *Int. J. Comput. Digit. Syst.*, vol. 10, no. 1, pp. 353–360, Feb. 2021, doi: 10.12785/ijcds/100135.

[214] P. J. Y. Piera and J. K. G. Salva, "A wireless sensor network for fire detection and alarm system," in *Proc. 7th Int. Conf. Inf. Commun. Technol.*, Jul. 2019, pp. 1–5, doi: 10.1109/ICoICT.2019.8835265.

[215] D. Chandirasekaran and T. Jayabarathi, "Cat swarm algorithm in wireless sensor networks for optimized cluster head selection: A real time approach," *Cluster Comput.*, vol. 22, no. S5, pp. 11351–11361, Sep. 2019, doi: 10.1007/s10586-017-1392-4.

[216] M. Nurgaliyev, A. Saymbetov, Y. Yashchyshyn, N. Kuttybay, and D. Tukymbekov, "Prediction of energy consumption for LoRa based wireless sensors network," *Wireless Netw.*, vol. 26, no. 5, pp. 3507–3520, Jul. 2020, doi: 10.1007/s11276-020-02276-5.

[217] S. Almuntasheri and M. J. F. Alenazi, "Software-defined network-based energy-aware routing method for wireless sensor networks in industry 4.0," *Appl. Sci.*, vol. 12, no. 19, p. 10073, Oct. 2022, doi: 10.3390/app121910073.

[218] L. Guezouli, K. Barka, S. Bouam, and A. Zidani, "Implementation and optimization of RWP mobility model in WSNs under TOSSIM simulator," *Int. J. Commun. Netw. Inf. Secur.*, vol. 9, no. 1, pp. 1–11, Apr. 2022, doi: 10.17762/ijcnis.v9i1.2116.

[219] X. Zhong and Y. Liang, "Scalable downward routing for wireless sensor networks actuation," *IEEE Sensors J.*, vol. 19, no. 20, pp. 9552–9560, Oct. 2019, doi: 10.1109/JSEN.2019.2924153.

[220] F. Mezrag, S. Bitam, and A. Mellouk, "Secure routing in cluster-based wireless sensor networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6, doi: 10.1109/GLOCOM.2017.8254138.

[221] L. Girod, N. Ramanathan, J. Elson, T. Stathopoulos, M. Lukac, and D. Estrin, "Emstar: A software environment for developing and deploying heterogeneous sensor-actuator networks," *ACM Trans. Sensor Netw.*, vol. 3, no. 3, p. 13, Aug. 2007, doi: 10.1145/1267060.1267061.

[222] H. Lee, A. Cerpa, and P. Levis, "Improving wireless simulation through noise modeling," in *Proc. 6th Int. Conf. Inf. Process. sensor Netw.*, 2007, p. 21, doi: 10.1145/1236360.1236364.

[223] A. Poenaru, R. Istrate, and F. Pop, "AFT: Adaptive and fault tolerant peer-to-peer overlay—A user-centric solution for data sharing," *Future Gener. Comput. Syst.*, vol. 80, pp. 583–595, Mar. 2018, doi: 10.1016/j.future.2016.05.022.

[224] K. Pal, M. C. Govil, and M. Ahmed, "FLHyO: Fuzzy logic based hybrid overlay for P2P live video streaming," *Multimedia Tools Appl.*, vol. 78, no. 23, pp. 33679–33702, Dec. 2019, doi: 10.1007/s11042-019-08010-4.

[225] N. Sabor, S. Sasaki, M. Abo-Zahhad, and S. M. Ahmed, "A graphical-based educational simulation tool for wireless sensor networks," *Simul. Model. Pract. Theory*, vol. 69, pp. 55–79, Dec. 2016, doi: 10.1016/j.simpat.2016.09.004.

[226] K. Chantzis, D. Amaxilatis, I. Chatzigiannakis, and J. Rolim, "Symmetric coherent link degree, adaptive throughput-transmission power for wireless sensor networks," in *Proc. IEEE Int. Conf. Distrib. Comput. Sensor Syst.*, May 2014, pp. 26–34, doi: 10.1109/DCOSS.2014.13.

[227] G. Serpen and Z. Gao, "Empirical model development for message delay and drop in wireless sensor networks," *Proc. Comput. Sci.*, vol. 36, pp. 353–358, Jan. 2014, doi: 10.1016/j.procs.2014.09.005.

[228] M. Asim, H. Mokhtar, and M. Z. Khan, "An energy efficient management scheme for wireless sensor networks," *Int. J. Crit. Comput.-Based Syst.*, vol. 6, no. 2, p. 133, 2015, doi: 10.1504/ijccbs.2015.073541.

[229] X.-X. Ding, T.-T. Wang, H. Chu, X. Liu, and Y.-H. Feng, "An enhanced cluster head selection of LEACH based on power consumption and density of sensor nodes in wireless sensor networks," *Wireless Pers. Commun.*, vol. 109, no. 4, pp. 2277–2287, Dec. 2019, doi: 10.1007/s11277-019-06681-8.

[230] M. Rafiuzzaman, J. Gascon-Samson, K. Pattabiraman, and S. Gopalakrishnan, "Failure prediction in the Internet of Things due to memory exhaustion," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 292–301, doi: 10.1145/3297280.3297311.

[231] X.-X. Ding, Y.-N. Liu, and L.-Y. Yang, "An optimized cluster structure routing method based on LEACH in wireless sensor networks," *Wireless Pers. Commun.*, vol. 121, no. 4, pp. 2719–2733, Dec. 2021, doi: 10.1007/s11277-021-08845-x.

[232] N. C. Resmi and S. Chouhan, "An enhanced methodology for energy-efficient interdependent source-channel coding for wireless sensor networks," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 4, pp. 1072–1080, Dec. 2020, doi: 10.1109/TGCN.2020.3008079.

[233] R. A. Reddy and N. V. Ram, "Data aggregation and precedence by delay sensitivity (DAP-DS): Data transmission over wireless body sensor networks," *Microprocessors Microsyst.*, vol. 77, Sep. 2020, Art. no. 103165, doi: 10.1016/j.micpro.2020.103165.

[234] J. Grosso and A. Jhumka, "Fault-tolerant ant colony based-routing in many-to-many IoT sensor networks," in *Proc. IEEE 20th Int. Symp. Netw. Comput. Appl. (NCA)*, Nov. 2021, pp. 1–10, doi: 10.1109/NCA53618.2021.9685935.

[235] A. Khan, S. Siddiqui, and S. Ghani, "Optimizing MAC layer performance for wireless sensor networks in eHealth," in *Proc. IEEE 44th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Madrid, Spain, Jul. 2020, pp. 1679–1682.

[236] S. Siddiqui and A. A. Khan, "Achieving energy efficiency for bursty traffic in wireless sensor networks," in *Proc. 22nd Int. Multitopic Conf. (INMIC)*, Nov. 2019, pp. 1–4, doi: 10.1109/INMIC48123.2019.9022731.

[237] J. Zhao, Y. Sun, Q. Deng, M. Cheng, and Q. Kuang, "Towards understanding bugs in WSN applications," *J. Comput.*, vol. 32, no. 1, pp. 1–10, 2021, doi: 10.3966/199115992021023201001.

[238] N. Moussa, A. El Belrhiti El Alaoui, and C. Chaudet, "A novel approach of WSN routing protocols comparison for forest fire detection," *Wireless Netw.*, vol. 26, no. 3, pp. 1857–1867, Apr. 2020, doi: 10.1007/s11276-018-1872-3.

[239] A. Attir, F. Naït-Abdesselam, and K. M. Faraoun, "Lightweight anonymous and mutual authentication scheme for wireless body area networks," *Comput. Netw.*, vol. 224, Apr. 2023, Art. no. 109625, doi: 10.1016/j.comnet.2023.109625.

[240] R. Elkamel, A. Messouadi, and A. Cherif, "Extending the lifetime of wireless sensor networks through mitigating the hot spot problem," *J. Parallel Distrib. Comput.*, vol. 133, pp. 159–169, Nov. 2019, doi: 10.1016/j.jpdc.2019.06.007.

[241] H. A. A. Al-Kashoash, Z.-A.-S. A. Rahman, and E. Alhamdawee, "Energy and RSSI based fuzzy inference system for cluster head selection in wireless sensor networks," in *Proc. Int. Conf. Inf. Commun. Technol.*, Apr. 2019, pp. 102–105, doi: 10.1145/3321289.3321319.

[242] A. Yahyaoui, T. Abdellatif, and R. Attia, "Hierarchical anomaly based intrusion detection and localization in IoT," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 108–113, doi: 10.1109/IWCMC.2019.8766574.

[243] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 760–776, Feb. 2019, doi: 10.1109/TITS.2018.2818888.

[244] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "Trust model for secure group leader-based communications in VANET," *Wirel. Netw.*, vol. 25, no. 8, pp. 4639–4661, 2019, doi: 10.1007/s11276-018-1756-6.

[245] R. Chaves, C. Senna, M. Luís, S. Sargento, A. Moreira, D. Recharte, and R. Matos, "EmuCD: An emulator for content dissemination protocols in vehicular networks," *Future Internet*, vol. 12, no. 12, p. 234, Dec. 2020, doi: 10.3390/fi12120234.

[246] M. Jaganath, R. Vasanth, and G. Malarselvi, "An exhaustive consideration of wired and wireless network simulators," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2S4, pp. 96–102, Aug. 2019, doi: 10.35940/ijrte.B1017.0782S419.

[247] A.-N. Ateeq, I. Obaid, O. Othman, and A. Awad, "Lifetime enhancement of WSN based on improved LEACH with cluster head alternative gateway," in *Proc. 4th Int. Conf. Future Netw. Distrib. Syst. (ICFNDS)*, Nov. 2020, pp. 1–6, doi: 10.1145/3440749.3442615.

[248] P. Srividya, L. Nirmala Devi, and A. Nageswar Rao, "A trusted effective approach for forecasting the failure of data link and intrusion in wireless sensor networks," *Theor. Comput. Sci.*, vol. 941, pp. 1–13, Jan. 2023, doi: 10.1016/j.tcs.2022.08.004.

[249] S. Aswale and V. R. Ghorpade, "EEMGR: Energy and ETX aware multipath geographic routing in WMSN," in *Proc. 3rd Int. Conf. Internet Things, Smart Innov. Usages (IoT-SIU)*, Feb. 2018, pp. 1–6, doi: 10.1109/IoT-SIU.2018.8519870.

[250] M. Mukherjee, L. Shu, R. V. Prasad, D. Wang, and G. P. Hancke, "Sleep scheduling for unbalanced energy harvesting in industrial wireless sensor networks," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 108–115, Feb. 2019, doi: 10.1109/MCOM.2019.1700811.

[251] S. Vinay Prasad and S. Sinha, "RSSI based improved weighted centroid localization method using indirect transmission and error estimation," in *Proc. IEEE Int. Conf. Data Sci. Inf. Syst. (ICDSIS)*, Jul. 2022, pp. 1–6, doi: 10.1109/ICDSIS55133.2022.9915992.

[252] S. Sinha, "Range based improved localization scheme in densely populated wireless sensor network," in *Proc. 6th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2021, pp. 792–797, doi: 10.1109/ICCES51350.2021.9489164.

[253] V. Srivastava, S. Tripathi, K. Singh, and L. H. Son, "Energy efficient optimized rate based congestion control routing in wireless sensor network," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 3, pp. 1325–1338, Mar. 2020, doi: 10.1007/s12652-019-01449-1.

[254] H. S. Zenalabdin, A. Buhari, and T. E. Nyamasvisva, "Performance analysis of IoT protocol stack over dense and sparse mote network using COOJA simulator," *J. Phys., Conf. Ser.*, vol. 1529, no. 5, May 2020, Art. no. 052007, doi: 10.1088/1742-6596/1529/5/052007.

[255] M. T. Dandjinou, H. Tall, and T. Yélémou, "Fault-tolerant routing protocol for ad hoc networks," in *Proc. IEEE Int. Conf Natural Eng. Sci. for Sahel's Sustain. Develop.-Impact Big Data Appl. Soc. Environ. (IBASE-BF)*, Feb. 2020, pp. 1–6, doi: 10.1109/IBASE-BF48578.2020.9069590.

[256] M. Pucher, C. Kudera, and G. Merzdovnik, "AVRS: Emulating AVR microcontrollers for reverse engineering and security testing," in *Proc. 15th Int. Conf. Availability, Rel. Secur.*, Aug. 2020, pp. 1–10, doi: 10.1145/3407023.3407065.

[257] B. K. Chejerla and S. K. Madria, "QoS guaranteeing robust scheduling in attack resilient cloud integrated cyber physical system," *Future Gener. Comput. Syst.*, vol. 75, pp. 145–157, Oct. 2017, doi: 10.1016/j.future.2017.02.034.

[258] V. G. Švenda, A. M. Stankovic, A. T. Saric, and M. K. Transtrum, "Influence of communication irregularities and co-simulation on hybrid power system state estimation," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Europe)*, Oct. 2018, pp. 1–6, doi: 10.1109/ISGTEUROPE.2018.8571859.

[259] S. Shukla, "Angle based critical nodes detection (ABCND) for reliable industrial wireless sensor networks," *Wireless Pers. Commun.*, vol. 130, no. 2, pp. 757–775, May 2023, doi: 10.1007/s11277-023-10308-4.

[260] K. Boikanyo, A. M. Zungeru, A. Yahya, and C. K. Lebekwe, "Performance evaluation of termite Hill routing algorithm on mobile and static sources of RPMS," in *Proc. Int. Conf. Smart Appl., Commun. Netw. (SmartNets)*, Nov. 2022, pp. 01–06, doi: 10.1109/SmartNets55823.2022.9994002.

[261] G. K. Ijemaru, L. M. Ang, and K. P. Seng, "Optimizing energy consumption and provisioning for wireless charging and data collection in large-scale WRSNs with mobile elements," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 17585–17602, Oct. 2023, doi: 10.1109/JIOT.2023.3277667.

[262] D. Centelles, A. Soriano, J. V. Marti, R. Marin, and P. J. Sanz, "UWSim-NET: An open-source framework for experimentation in communications for underwater robotics," in *Proc. OCEANS*, Jun. 2019, pp. 1–8, doi: 10.1109/OCEANSE.2019.8867460.

[263] D. Centelles, A. Soriano, J. V. Martí, and P. J. Sanz, "Underwater multirobot cooperative intervention MAC protocol," *IEEE Access*, vol. 8, pp. 60867–60876, 2020, doi: 10.1109/ACCESS.2020.2983641.

[264] Q. Tang, D. Chen, Y. Li, and L. Tang, "Three-dimensional visual simulation for trajectory tracking of autonomous underwater vehicle based on UWSim," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2019, pp. 1175–1180, doi: 10.1109/AIM.2019.8868418.

[265] S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, "A predictive control approach for cooperative transportation by multiple underwater vehicle manipulator systems," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 3, pp. 917–930, May 2022, doi: 10.1109/TCST.2021.3085121.

[266] Z. Shen, J. P. Wilson, and S. Gupta, "An online coverage path planning algorithm for curvature-constrained AUVs," in *Proc. OCEANS MTS/IEEE SEATTLE*, Seattle, WA, USA, Oct. 2019, pp. 1–5, doi: 10.23919/OCEANS40490.2019.8962629.

[267] D.-H. Gwon, J. Kim, M. H. Kim, H. G. Park, T. Y. Kim, and A. Kim, "Development of a side scan sonar module for the underwater simulator," in *Proc. 14th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Jun. 2017, pp. 662–665, doi: 10.1109/URAI.2017.7992789.

GHAIHAB HASSAN ADDAY was born in Basrah, Iraq, in 1981. He received the B.S. degree in computer science and the M.S. degree in routing in wireless networks from the University of Basrah, Basrah, in 2008 and 2012, respectively. He is currently pursuing the Ph.D. degree in wireless sensor networks with Universiti Putra Malaysia (UPM), Selangor, Malaysia.

From 2012 to 2014, he was a Lecturer Assistant with the Department of Computer Science, College of Science, University of Basrah. Since 2014, he has been a Lecturer with the Department of Computer Science, College of Computer Science and Information Technology, University of Basrah. From 2016 to 2019, he was a Secretary with the College Council, College of Computer Science and Information Technology, University of Basrah. From 2019 to 2020, he was the Assistant Dean of Administrative and Financial Affairs with the College of Computer Science and Information Technology, University of Basrah. His current research interests include routing protocols in wireless sensor networks, fault tolerance framework in wireless networks, trust management in wireless networks, the Internet of Things (IoT), and information security.

SHAMALA K. SUBRAMANIAM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Universiti Putra Malaysia (UPM), Selangor, Malaysia, in 1996, 1999, and 2002, respectively. She is currently a Professor with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, UPM. Her research interests include computer networks, simulation, and modeling.

ZURIATI AHMAD ZUKARNAIN (Member, IEEE) received the B.S. and M.S. degrees in physics and education from Universiti Putra Malaysia (UPM), Selangor, Malaysia, in 1997 and 2000, respectively, and the Ph.D. degree in quantum computing and communication from the University of Bradford, U.K. in 2005.

Since 2001, she has been an Academic Staff Member with the Faculty of Computer Science and Information Technology, UPM. From 2006 to 2011, she was the Head of the Department of Communication Technology and Networks. From 2012 to 2015, she was also the Head of the Section of High-Performance Computing Institute of Mathematical Research, UPM. She taught several courses for undergraduate students, such as data communication and networks, distributed systems, mobile and wireless, network security, computer architecture, and assembly language. She taught a few courses for postgraduate students, such as the advanced distributed and research method. Her research interests include computer networks, distributed systems, mobile and wireless, network security, quantum computing, and quantum cryptography. She is a member of the IEEE Computer Society.

NORMALIA SAMIAN (Member, IEEE) received the Ph.D. degree from Universiti Putra Malaysia (UPM), Selangor, Malaysia, in 2017, with a focus on cooperation in wireless multi-hop networks.

She is currently a Senior Lecturer with the Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM). She is also the Leader of the Wireless, Mobile, and Quantum Computing (WiMoQ) Research Group and the Head of the Academic Advisor with the Faculty of Computer Science and Information Technology. She is leading a grant project on securing the IoT networks using blockchain technology. Her research interests include ad hoc network security, cooperation, and trust management in wireless networks, the Internet of Things (IoT), and blockchain technology. She has published several impact factors journals and tier-A conferences related to her fields.

Dr. Samian received the N2 Women Young Researcher Fellowship from IEEE LCN2016, Dubai, during the Ph.D. degree. She has served as a reviewer/technical program committee for international journals/conferences.