Received: 16 December 2023 | *Revised:* 25 March 2024 | *Accepted:* 13 April 2024
DOI: 10.37917/ijeee.22.1.12 *Vol. 22 | Issue 1 | June 2026*

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Open Access
IJEEE University of Basrah College of Engineering

Machine Learning-Based Detection and Prevention of DoS and DDoS Attacks in SDWSN

Eng. Ahmad Al Ebrahim*, Prof. Dr. Abdelrazak Badawieh
Department of Electronics and Communications Engineering, Faculty of Mechanical and Electrical Engineering, Damascus University, Syria

Correspondance
*Ahmad Al Ebrahim
Department of Electronics and Communications Engineering, Faculty of Mechanical and Electrical Engineering, Damascus University, Syria
Email: ahmad.alebrahim@damascusuniversity.edu.sy

Abstract
Software Defined Wireless Sensor Networks (SDWSN) has emerged as a contemporary model to achieve dynamic and secure control in the realm of Internet of Things (IoT) applications. By leveraging the benefits of Software Defined Networks (SDN), SDWSN enables ease of management and configuration of wireless networks, thereby overcoming the challenges associated with traditional Wireless Sensor Networks (WSN). However, SDWSN networks are susceptible to emerging network intrusion and threats, particularly Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks, which can significantly impact the network's performance and cause operational losses. This study proposes a machine learning based algorithm for detecting and preventing DoS and DDoS attacks in SDWSN networks. The proposed algorithm uses various features to distinguish between benign traffic and malicious traffic generated by attacks. The results demonstrate that the proposed algorithm can effectively detect and prevent DoS attacks, significantly contributing to the security of SDWSN networks.

Keywords
SDWSN, SDN, WSN, IoT, Machine Learning, Denial of Service (DoS) Attacks, Distributed Denial of Service (DDoS) Attacks, Security, Detection.

I. Introduction

The incorporation of Software-Defined Networking (SDN) models into Wireless Sensor Networks (WSNs) has yielded significant advantages for diverse applications, including Network Functions Virtualization (NFV), data centres, enterprise networks, and Internet of Things (IoT) applications [1]. This convergence, resulting in Software-Defined Wireless Sensor Networks (SDWSNs), has garnered substantial interest from researchers, developers, and manufacturers [1]. These networks find application in various environments encompassing university campuses, data centres, smart industries, military bases, and other security-sensitive locations. Given the potential sensitivity of data exchanged between nodes and network control units, securing these networks remains an active research area. In practice, cryptographic techniques such as symmetric, asymmetric, and hybrid encryption are employed to safeguard against intrusions within SDWSNs, while Intrusion Detection Systems (IDS) serve as the primary line of defence for intrusion detection [2]. The SDWSN model demonstrably overcomes inherent limitations commonly associated with traditional WSNs, including restricted computational resources, limited data storage capacities, constrained transmission bandwidth, and inflexibility in security implementation. Moreover, the centralized control architecture of SDWSNs facilitates the application of more robust security measures for network resources. However, despite these advancements, the SDWSN network model is not immune to new security challenges [3]. One distinctive feature of the centralization of SDWSNs is the radical transformation of network architecture by separating network intelligence and logic from routing devices [4]. The open nature of Application

This is an open-access article under the terms of the Creative Commons Attribution License, which permits use, distribution, and reproduction in any medium, provided the original work is properly cited.
©2026 The Authors.
Published by Iraqi Journal for Electrical and Electronic Engineering | College of Engineering, University of Basrah.

Programming Interfaces (APIs) between the infrastructure, application, and control planes creates exploitable entry points for attacks on the network. These attacks can target sensor node clusters and control units through Denial of Service (DoS) tactics, aiming to deplete their resource capacities [5]. Routing devices themselves can also be targeted by attackers to manipulate network traffic. This manipulation can take the form of dropping, slowing down, or tampering with data packets. Consequently, attackers can exploit the lack of robust detection mechanisms to inject fake traffic into the network. This fake traffic can then be used to overwhelm both control plane elements, such as network controllers, and data plane elements, including switches, routers, and sensor clusters, etc., in the absence of a detection mechanism [6]. In fact, the proliferation of smart devices and advancements in wireless technologies have spurred the development of SDWSN. These networks are finding widespread application in various Internet of Things (IoT) domains [7]. However, this growing adoption also amplifies the vulnerability of SDWSNs to security breaches perpetrated through Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. These attacks overwhelm the network with illegitimate traffic, hindering authorized user access and disrupting critical network services like data transmission, control, and monitoring [7]. The remainder of this paper is organized to provide a comprehensive overview of the proposed solution for DoS/DDoS attack detection and prevention in SDWSNs.Section II. reviews existing research on various aspects of SDWSNs, and identifies relevant prior work and establishes the context for the proposed approach. In Section III. describes the typical structure of SDWSNs and explores the security threats associated with these networks. Section IV. presents the core contribution of the paper: a novel algorithm for detecting and preventing DoS and DDoS attacks in SDWSNs. The section details the algorithm's functionality and includes relevant flowcharts for improved clarity. Section V. is devoted to presents the findings of the research. This section is further divided into several subsections, which cover the emulation process and parameters, performance metrics, as well as the obtained results and discussion. Finally, Section VI. summarizes the paper's main contributions, highlighting the proposed algorithm's effectiveness in mitigating DoS/DDoS attacks within SDWSNs. Additionally, it suggests potential directions for future research in this domain.

II. Related Studies

A growing body of research has explored the application of machine learning for intrusion detection systems (IDS) in wireless networks. This approach leverages the capabilities of machine learning algorithms to identify and classify malicious activities within wireless network traffic. [8] sheds light on WSN restrictions, weaknesses, and security threats with a focus on DoS attacks. Accordingly, this study proposes a lightweight Machine Learning (ML) detection approach based on a Decision Tree (DT) algorithm with the Gini feature selection method to detect DoS attacks in WSNs. An enhanced version of the WSN-DS dataset, developed by the author, was used to train and test the proposed approach. The proposed approach has shown good performance by achieving a remarkable accuracy rate of 99.5% with while maintaining minimal computational overhead compared to random forest (RF), extreme gradient boosting (XGBoost), and k-nearest neighbor (KNN) classifiers. It only takes 9.7%, 13%, and 2% of the processing time required by FR, XGBoost, and KNN respectively, which indicates that proposed approach significantly outperforms these classifiers in terms of processing time. While it is acknowledged that RF achieved a slightly higher accuracy, the proposed DT approach offers a substantial advantage in processing time, requiring only 9.7% of the time taken by RF. This efficiency is a crucial factor when considering the inherent resource constraints of WSNs. Building upon recent advancements in non-parametric real-time change point detection, [9] proposes a DoS attack detection algorithm specifically designed for software-defined resource constrained wireless networks. Notably, the lightweight design allows the algorithm to operate efficiently on individual sensor nodes with limited resources. Experimental evaluation demonstrates that the algorithm achieves high detection rates and attacker identification probabilities exceeding equal or over 0.93, comparable to centralized detection methods. Two implementation approaches were examined: running the detector on every node and deploying it in clusters. Both approaches yielded high detection rates. Additionally, when running the detector on each node, the algorithm successfully identified attackers without generating additional network traffic. However, a trade-off exists between the two implementations. The clustering approach requires less memory on each node, but generates more network traffic compared to running the detector on every node. While [10] proposed a machine learning multi-layer DDoS detection system to prevent DDoS attacks in IoT gateway by extract features of four types of DDoS attacks, including sensor data flood, ICMP flood, SYN flood, and UDP flood and make these features to be numerical. Then launch DDoS attacks from eight smart poles as a real IoT scenario and show that multi-layer DDoS detection system can distinguish normal packets and DDoS attack packets from IoT devices accurately. The proposed system was able to detect DDoS attacks with high accuracy. The F1-score is over 97% using the DT algorithm. Upon identifying anomalous packets, the system transmits the IP and MAC addresses of the malicious devices to the SDN controller. This controller then utilizes these addresses to populate blacklists and configure

SDN switch rules accordingly. Consequently, the blacklisted devices are promptly blocked by the SDN switches. This multi-layered approach not only effectively detects DDoS attacks but also actively mitigates them by blocking malicious devices. Whereas [11] demonstrates the use of ML algorithms for traffic monitoring within Network Intrusion Detection Systems (NIDS) deployed in SDN controllers. Different classical and advanced tree-based machine learning techniques, Decision Tree (DT), Random Forest (RF) and XGBoost are chosen, and suing six different evaluation metrics to demonstrate attack detection. The NSL-KDD dataset is used for training and testing the proposed methods. Focusing on a subset of five features from the NSL-KDD dataset (out of a total of 41), the proposed approach tackles multi-class classification. This involves not only detecting the presence of an attack but also classifying the specific attack type. The proposed method achieves a high accuracy of 95.5% in this task. This research demonstrates the potential of machine learning for real-time anomaly detection and protection of SDN platforms from cyberattacks. Whereas [12] a method was proposed that combines statistical features such as Speed of IP Sources, Speed of Flow Entries, Ratio of Pair-Flow Entries, with Support Vector Machines (SVM) ML algorithm for efficient DDoS attack detection and mitigation in SDN. The implementation utilized the Ryu controller and Mininet network simulator with the OpenFlow SDN protocol. The authors reported an accuracy of 99.26% and a 100% detection rate for DDoS attacks using their SVM-based approach within an SDN environment, with no false positives identified. However, a potential limitation of this method is its vulnerability to attacks originating from trusted IP sources. Since the SVM relies on historical traffic patterns, it may not be able to effectively predict malicious traffic originating from previously trusted IP addresses. [13] proposed a lightweight and highly effective approach for detecting DDoS attacks in SDWSNs using change point analysis. By demonstrate the performance of the change point detector in SDWSNs of varying sizes (36 and 100 nodes) and attack intensities (attacker numbers ranging from 5% to 20% of total nodes). They monitored anomalies in two key metrics: data packet delivery rate and control packet overhead. The results demonstrated that their approach achieved near-perfect detection rates (close to 100%) as the attack intensity increased. Additionally, the method exhibited the capability to infer the type of attack being launched. In a comparative study, [14] explored the use of machine learning algorithms for IDS system in SDWSN. This investigation evaluated the performance of three algorithms: DT, SVM, and Logistic Regression (LR). The results show that SVM model is the most effective algorithm followed by DT in terms of detection rate of both the normal and anomaly instances. On the basis of efficiency, DT produced 77.43% and 22.57% accuracy for correct and incorrect classification respectively as well as minimum time for classification in both training and testing. Based on these findings, the authors suggest that the DT algorithm offers a more efficient and effective solution for real-time intrusion detection in SDWSNs, enabling the network to maintain a constant state of alertness and proactive response. Similarly, [2] compared the performance of three Artificial Intelligence (AI) approaches for IDS in SDWSNs: Decision Trees (DT), Deep Learning (DL), and Naïve Bayes (NB). The evaluation focused on identifying the most suitable approach for SDWSN applications, considering both performance and resource constraints. The study found that NB was the most suitable for SDWSN applications due to its lower energy consumption requirements. The performance of three anomaly detectors (DT, NB and deep ANN) were recorded and the results show that the DT-based anomaly detector should be used as the default anomaly detector in SDWSNs unless a specific requirement such as an extremely low memory size available on the controller demands an alternate anomaly detector. Since the accuracy of deep ANN increases as the dataset gets bigger and since more attacks are identified and recorded every day in datasets, then we expect that soon deep ANNs could be the default anomaly detectors for SDWSNs requiring a very high level of security. However, DT showed the best overall performance in detecting abnormal cases, especially for delay-sensitive applications. While [15] proposed a novel Low-Rate Denial of Service (LDoS) attack detection method termed HCN based on Hilbert–Huang Transform (HHT) and Convolutional Neural Network (CNN). The proposed method involved the extraction of the features of multiple one-dimensional data sources from the network, and these were then transformed into two-dimensional frequency spectrum features by compressed HHT. This transformation enables a more comprehensive characterization of network traffic dynamics. The resulting two-dimensional frequency spectrum features were input into a CNN for LDoS attack detection. To assess the detection efficacy of HCN, a range of LDoS attacks were simulated within a SDWSN testbed environment using Network Simulator-3 (NS-3). The experimental results demonstrate that HCN achieves a high level of accuracy while maintaining low false positive and false negative rates. This performance suggests that HCN is a promising approach for LDoS attack detection in resource-constrained SDWSN environments. Another study by [16] proposed a hybrid IDS for SDWSN which leverages the Salp Swarm Optimization (SSO) algorithm for feature selection and incorporates a Random Forest (RF) classifier for intrusion detection. The SSO algorithm optimizes the selection of relevant features, aiming to improve the detection efficiency of the RF classifier. The authors evaluate the proposed approach using the widely used NSL-KDD dataset to assess its reliability.

Therefore, the proposed hybrid IDS-SSO-RF classifier further analyzes these detected abnormal activities. The known and unknown attacks are also identified. Hybrid framework also shown by the experimental results can reliably detect anomaly behavior and obtains better results in terms in terms of delay, delivery ratio, drop overhead, energy consumption and throughput. Based on the abovementioned studies, this study aims to develop a machine learning (ML) based algorithm for detecting and preventing DoS and DDoS attacks in SDWSN networks to enhance their security. The key contributions of this study can be highlighted as follows.

- The primary contribution of this study lies in the development of a sophisticated machine learning based algorithm tailored specifically for the intricate domain of SDWSN. By leveraging a diverse array of features such as packet type, packet size, packet rate, and statistically computed attributes, our algorithm exhibits a unique capability to discern between legitimate network traffic and malicious activities propagated by DoS/DDoS attacks.
- Furthermore, our research introduces a novel approach to enhancing the security landscape of SDWSN networks by formulating a robust classifier that effectively differentiates benign traffic from malicious traffic. This pioneering methodology not only fortifies the network infrastructure against potential threats but also contributes significantly towards mitigating the adverse impacts of cyber-attacks on operational efficiency and performance metrics.
- Through a meticulously orchestrated Emulation using the Mininet-Wi-Fi platform in conjunction with the Ryu-Manager controller, we have demonstrated the efficacy and practical applicability of our proposed algorithm in real-world scenarios. By showcasing its adeptness in detecting and preventing DoS attacks, our research accentuates a pivotal milestone in fortifying the security paradigm of SDWSN networks.

In essence, the amalgamation of cutting-edge machine learning techniques with the exigencies of SDWSN security underscores the pioneering essence of our study, heralding a paradigm shift in combatting cyber threats endemic to modern network architectures.

III. Software Defined Wireless Sensor Network (SDWSN)

Traditional Wireless Sensor Networks (WSNs) lacking Software-Defined Networking (SDN) capabilities require specific mechanisms to discover their network infrastructure. These networks typically rely on periodic broadcasts initiated by individual nodes to identify their immediate neighbours. This approach, however, places a significant burden on the network in terms of message overhead and energy consumption. Once the network topology is established, routing decisions are made. However, each node must store routing tables within its limited memory capacity and independently calculate routing paths for other nodes, further straining its limited resources. In contrast, SDWSNs delegate many of these tasks to a control unit, which boasts superior computational resources, an independent power source, and a holistic view of the entire network. SDWSNs do not necessitate periodic broadcast messages for network topology discovery; routing decisions are orchestrated by the control unit, obviating the need for nodes to maintain routing information in their individual tables. Additionally, the control unit can adjust the transmission range for each node to reduce communication interference between nodes. These computationally intensive tasks are carried out by the control unit in SDN-based WSNs, thereby preserving the residual energy within the nodes [17]. Figure 1 presents an overview of the SDWSN model, where sensor nodes function solely for data acquisition and forwarding, delegating network intelligence to a dedicated SDN control unit. This control unit is responsible for computationally intensive tasks associated with network management, including traffic management, Quality of Service (QoS) management, resource allocation, network performance monitoring, and energy level estimation [17]. Within SDWSNs, the infrastructure, or data plane, is primarily comprised of sensing nodes, which are composed of hardware components (such as a power unit, sensing unit, and radio communication unit) and software components (such as flow tables, sensing elements, and processing capabilities). The sensing unit generates traffic, subsequently processed by the network's central processor. The network can perform data aggregation in the sensing nodes or data fusion in the sink nodes to process the data [3].

SDWSNs rely heavily on the control unit as the central entity for network intelligence. This unit plays a pivotal role in the network by creating flow rules and mapping programming interfaces. Traditionally, SDWSNs utilize an integrated control unit that connects to a dedicated sink node via a serial interface connection. This sink node then serves as the bridge for wireless communication with the distributed sensor nodes. However, some network architectures integrate the control unit directly into the base station, effectively combining these functionalities within a single device. Within the control unit, the intermediate plane manages crucial tasks such as assigning network functions and defining flow tables, further enhancing the programmatic capabilities of the SDWSN. The control unit within the control plane governs network management and supervises its operations, and keeps the network status up

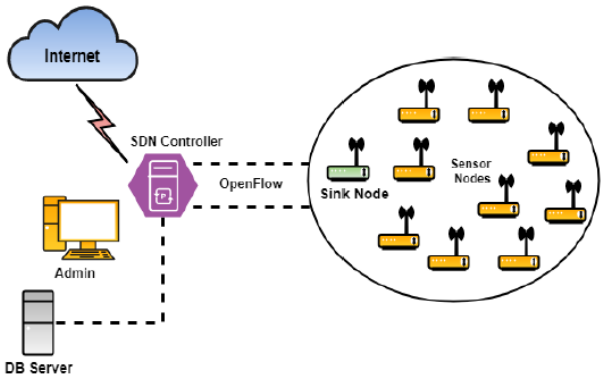

Fig. 1. Overview of SDWSN [17]

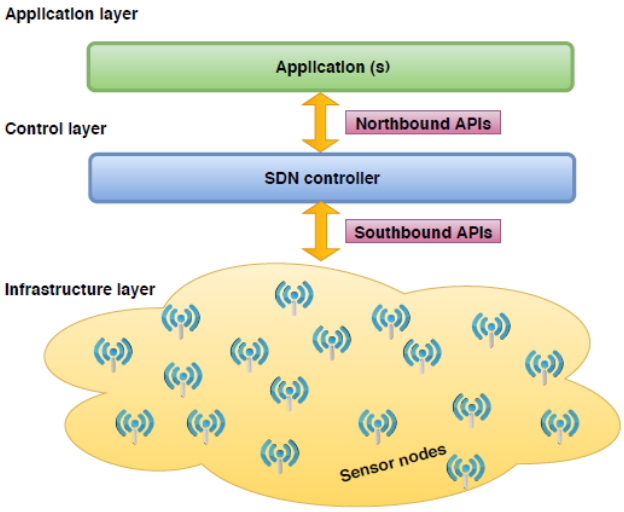

Fig. 2. Structure of SDWSN [18]

to date by constantly monitoring messages. These messages convey critical network information, including the energy levels of sensing nodes, distance from the base station, neighbor list, and connection status such as link quality and response time [3].

SDWSNs can be configured with two distinct control architectures [19]:

- Directly connected controller: a dedicated communication channel is required to establish direct control over the data traffic of all sensing nodes within the network.
- Indirectly connected controller: architecture utilizes a separate node to relay communication between the controller and the sensor nodes. In this case, the controller manages data traffic through the established backbone network.

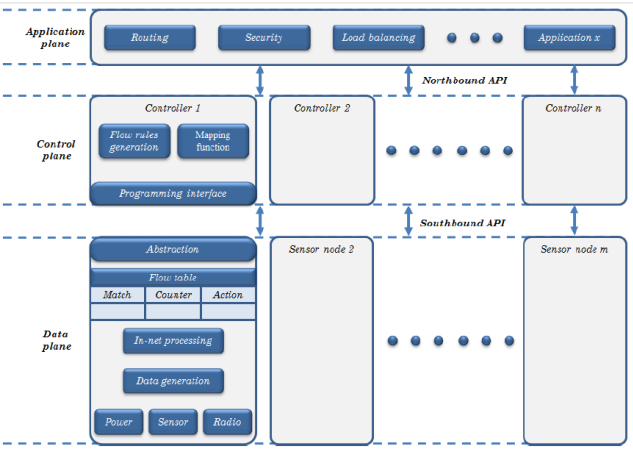

Fig. 3. Logical planes of SDWSNs [3]

A. *Security Threats in SDWSN*

The SDN paradigm presents a promising approach to bolstering the security of WSN networks, which can be seamlessly integrated into the security framework of SDWSN networks. Given that centralized control simplifies security management and alleviates resource constraints on individual nodes, offering significant advantages for secure network implementation [7]. Notably, the separation of the control plane from the data plane in SDN networks renders sensor nodes incapable of directly accepting control commands. This characteristic significantly reduces the vulnerability of sensor nodes to exploitation in malicious attacks, although they may still function as gateways for such attacks [3]. In order to identify security solutions for the SDWSN network that can be adapted from SDN and WSN networks, it is necessary to first identify the security threats present in both networks. The application plane of SDWSN architecture comprises the application layer in the WSN stack as well as the application plane in the SDN network model, and often under threat from security breaches such as authentication and authorization challenges [20], lack of trust mechanisms [4], as well as threats from network applications. Proposed solutions for such threats include implementing verification models or encryption solutions to secure northbound interface communications [7]. The control plane of the SDWSN architecture contains the transport and network layers in the WSN stack, as well as the control plane in the SDN network model. Encryption solutions could be employed in WSN networks to secure the control unit against transport layer and network layer attacks. Furthermore, researchers have proposed various models that incorporate dedicated control units to enhance the system's resilience against malicious applications. These control units also contribute to improved scalability, ultimately mitigating the impact of DoS attacks [20]. The infrastructure plane (or data plane)

of the SDWSN architecture contains the data link layer and physical layer in the WSN stack and the data plane in the SDN network model. A significant advantage of the SDN approach in WSNs) is its ability to mitigate many vulnerabilities residing within the data link and physical layers. This stems from the inherent limitation of sensor nodes in WSNs. Unlike traditional networks, sensor nodes can only process and respond to control commands. This restricted functionality impedes their potential use as malicious actors within the network [4]. While the aforementioned solutions may offer some improvements to the security posture of SDWSN networks, they present limitations. These approaches often rely on external security mechanisms that focus on protecting network infrastructure elements like control units, routers, and switches. To achieve robust security for SDWSN networks, security considerations must be intrinsically embedded within the network architecture itself, ensuring compatibility with other design constraints [3, 4]. A summary of security threats associated with each plane of SDN and WSN networks can be found in [7].

IV. Proposed Algorithm for Detection and Prevention of DoS and DDoS Attacks in SDWSN

The proposed algorithm employs a diverse set of features for training a classifier that can distinguish benign traffic from malicious DoS/DDoS attack traffic. These features include packet type, size, rate, source and destination, along with statistically derived features. Figure 4 illustrates the flowchart data collection approach used for both training and comparison purposes. The data collection process involves the following steps:

- *Input Traffic Flow*: Network traffic is generated by emulating an SDWSN within a simulation environment, where nodes exchange data with the controller via access points.
- *Traffic Collection*: Network traffic data is collected and stored on the controller.
- *Parameter Matching and Labelling*: Relevant parameters are extracted from the collected data and assigned labels based on their classification as Benign Traffic or malicious data generated by a DoS attack or DDoS attack (DDoS Traffic).
- *Feature Engineering*: The data undergoes a cleaning process to remove irrelevant information which negatively affects the model training results (e.g., additional columns that may be altered by the attack method or attacker node) and duplicate data that could bias the model.
- *Normalization*: Each data point in the dataset is normalized, scaling all features to a common range (typically 0 to 1) to prevent the classification model from favoring features with larger scales. Following feature engineering, the prepared dataset is then used to train the Decision Tree (DT) algorithm.

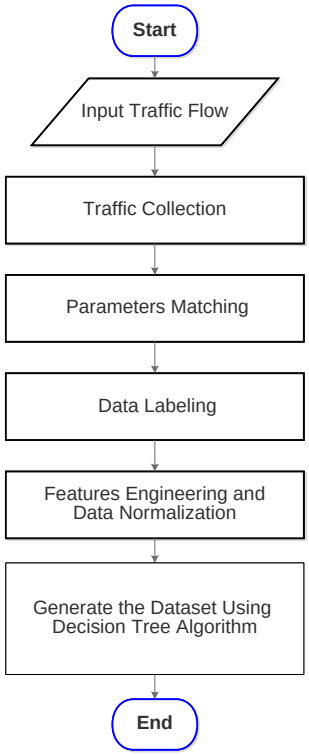

Fig. 4. Flowchart for the data collection process used in the proposed algorithm.

A DT algorithm is employed in the proposed scheme with the following feature set tp_src, tp_dst, icmpv4_ck, tcp_ck, udp_ck, icmp_code, icmp_type, flow_duration_sec, packet_count, byte_count, packet_count_per_second, byte_count_per_second, sfe, ssip, rfip, type in the following ordered steps:

1. *Data Collection and Feature Engineering (Preparation)*: This initial phase involves gathering network traffic data from the SDWSN. The features mentioned above are then extracted and prepared for subsequent analysis.
2. *Decision Tree Training (Learning)*: A portion of the collected data is manually labelled as either ”Benign Traffic” or ”DDoS Traffic.” This labelled data is then

used to train the Decision Tree model, enabling it to learn the key characteristics that differentiate normal network behaviour from DoS/DDoS attack patterns.

3. *Detection and Mitigation (Action)*: Real-time network traffic data is continuously fed into the trained Decision Tree model. Based on the model's predictions, the traffic is classified as either "Benign" or "DDoS" Upon detecting a DoS/DDoS attack, appropriate mitigation strategies can be implemented.

Where:

- tp_src: denotes the sending port.
- tp_dst: denotes the receiving port.
- icmpv4_ck: verifies the ICMP protocol.
- tcp_ck: verifies the TCP protocol.
- udp_ck: verifies the UDP protocol.
- icmp_code: specific code of ICMP protocol.
- icmp_type: identifies the ICMP protocol type.
- flow_duration_sec: data flow rate per second.
- packet_count: represents the cumulative number of sent packets.
- byte_count: denotes the number of bytes sent.
- packet_count_per_second: represents the number of packets sent per second.
- byte_count_per_second: represents the number of bytes sent per second.
- sfe: measures the Speed of Flow Entries, which represents total number of flow entries to the OpenFlow switch in network N during a specific period T.
- ssip: measures the Speed of IP Sources, which is the total number of logical addresses (IPs) received in the network during a specific time period.
- rfip: represents the Ratio of Pair-Flow Entries, which is the total number of flow entries received by the OpenFlow Switch.
- type: indicates the presence of a DoS/DDoS attack, where benign flows are denoted by (0) and attack flows are denoted by (1).

Figure 5 illustrates the proposed algorithm for detecting and preventing DoS and DDoS attacks in SDWSN. The algorithm introduces new data traffic flows into the WSN. Data is exchanged between sensor nodes, the controller, and potential attacker stations through access points and OpenFlow switches. The algorithm then collects and stores this flow data in the controller. Subsequently, feature engineering is performed to extract relevant characteristics from the data. Finally, the data is normalized to a range of 0 to 1 to facilitate training and comparison during attack detection. Notably, the dataset specific to the newly introduced traffic is stored in real-time within a dedicated file named "Predict_Flow" This file serves as a crucial component for predicting and subsequently detecting DoS attacks. The proposed algorithm employs a pre-trained model, "X_Flow_Predict", to detect and mitigate DoS attacks. This model operates by analyzing the (type) field within each data record stored in the "Predict_Flow" file. The model then infers a corresponding "Y_Flow_Predict" value, which includes a (type) field. If the (type) field value is equal to one, it signifies the presence of a DoS attack. In response, the algorithm identifies the attacker's physical address (MAC address) and adds it to a "Block List". Additionally, it modifies the Action field within the OpenFlow Switch to drop all incoming packets originating from this blocked address, preventing them from reaching the controller. Conversely, if the (type) field value is zero, the algorithm maintains the default forwarding mode for the Action field in the OpenFlow Switch, allowing legitimate packets from this address to reach the controller.

The proposed algorithm guarantees that DoS attacks do not reach the controller by dropping all incoming packets and ensuring the uninterrupted operation of the controller and network.

V. Results and Discussion

To evaluate the effectiveness of the proposed algorithm, an emulation environment was established. This environment employed Mininet-WiFi to simulate the SDWSN and an open-source Ryu controller for network management. The practical application aimed to assess the algorithm's real-time detection capabilities and its impact on network performance. The following section details the emulation setup, including the environmental parameters utilized. The algorithm's performance was evaluated using established metrics: Packet Delivery Ratio (PDR), Packet Loss Ratio (PLR), End-to-End Delay, Average Power Consumption, and Round-Trip Time (RTT) analysis. Additionally, an analysis of performance metrics specific to machine learning algorithms was conducted.

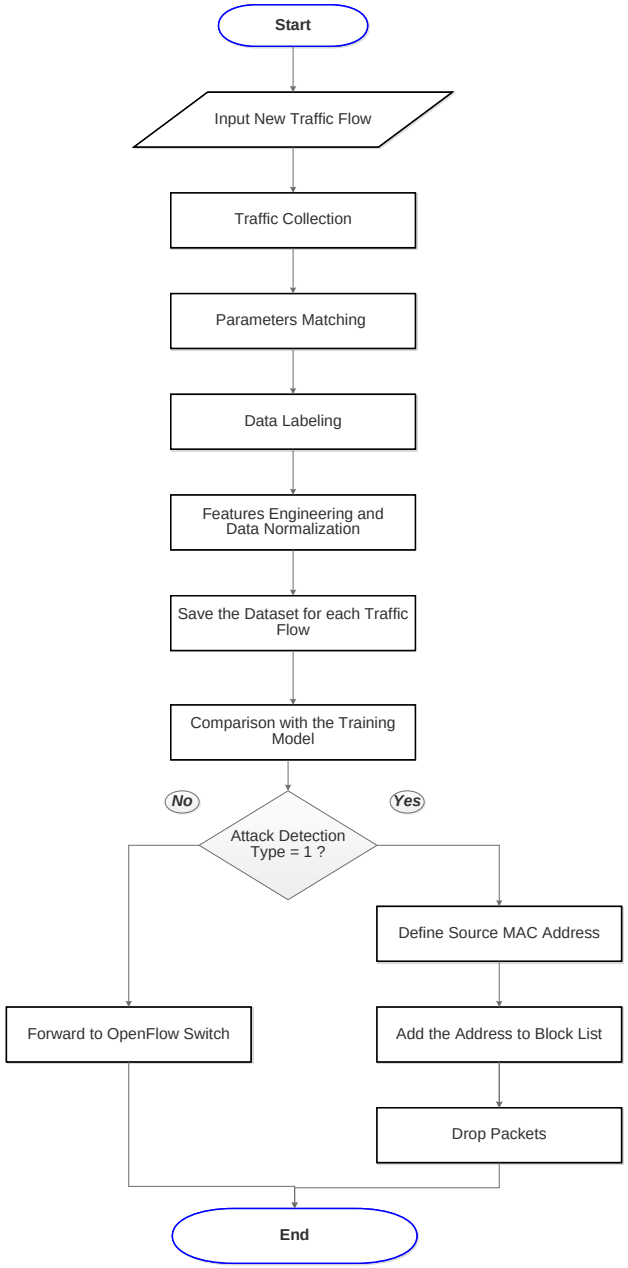

Fig. 5. Flowchart for the data collection process used in the proposed algorithm.

A. Emulation Process

This study presents a benchmarking framework designed to evaluate the performance of a novel algorithm for detecting DoS and DDoS attacks within a SDWSN network. The framework simulates a real-world deployment scenario within a 500m x 500m area populated with 24 sensor nodes (Sens1 to Sens24) arranged in a fixed node distribution pattern. A Ryu-Manager controller oversees network operations, while access points (AP1 and AP2) facilitate communication between the sensor nodes and the controller. An OpenFlow Switch is employed to manage data exchange between the controller, attack stations, and access points. To simulate the attacks, four attack stations (Sta1 to Sta4) were introduced. Data generated by this simulated attack traffic was collected and stored in a designated file named "Predict_Flow" for the purpose of training and evaluating the proposed detection algorithm. Data exchange operations were conducted as follows. Sensing nodes (Sens1 to Sens12) exchanged data with the first access point (AP1) using the ping tool, while sensing nodes (Sens13 to Sens24) exchanged data with the second access point (AP2) also using the ping tool. Attack stations 1 (Sta1) and 2 (Sta2) targeted the first access point (AP1) using the hping3 tool. Attack stations 3 (Sta3) and 4 (Sta4) targeted the second access point (AP2) also using the hping3 tool. Finally, AP1 and AP2 were directly connected to the OpenFlow Switch, which was then connected to the controller.

B. Emulation Parameters

This section details the fundamental parameters employed in the proposed work environment across various simulation scenarios. These parameters, presented in TABLES I, II, and III, encompass general environmental parameters, wireless channel and antenna Parameters, and Energy Parameters sequentially. Furthermore, to facilitate simulation and data encoding, a fixed distribution scheme for sensing nodes within the work environment is established, as illustrated in Fig. 6. Each sensing node is assigned a sequential numerical identifier, ranging from Sens1 to Sens24. Attacker stations are labelled Sta1 to Sta4, and access points are designated AP1 and AP2. This consistent naming convention streamlines the simulation process.

C. Performance Metrics

This study utilizes several key measures for evaluation and comparison, focusing on network performance and the performance of the employed machine learning algorithm.

1. *Network Performance Metrics:* The specific network performance metrics employed in this study are Packet Delivery Ratio (PDR), Packet Loss Ratio (PLR), Average End to End Delay (E2EAD), and Average Energy

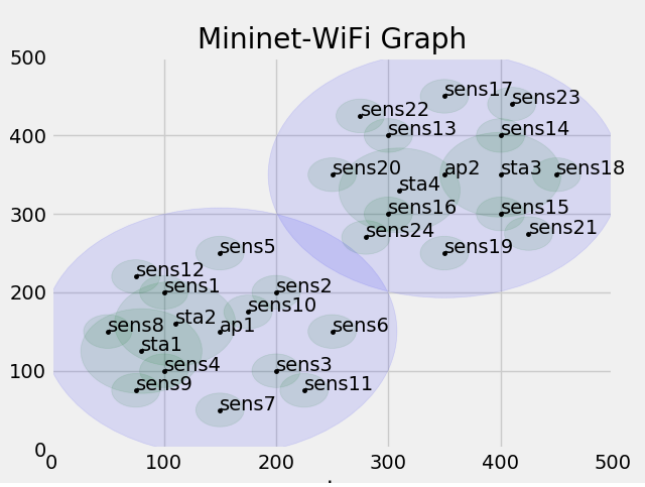

Fig. 6. Mininet-WiFi Graph.

TABLE I. GENERAL ENVIRONMENTAL PARAMETERS

Parameter	Value
Field size	500X500 [m2]
Controller type	Ryu-Manager Controller
Number of controllers	1
Number of sensor nodes	24
Number of access points	2
Number of attack stations	4
Nodes distribution	Fixed
Switching protocol	OpenFlow1.3
Traffic types	ICMP–TCP–UDP

TABLE II. WIRELESS CHANNEL AND ANTENNA PARAMETERS

Parameter	Value
Channel type	Wireless
Network interface type	Wireless Phy
Wireless protocol	802.11
Wireless protocol mode	N
Frequency	2.4 [GHz]
Communication channel	1 and 6
Antenna gain for sensor node	3 [dBi]
Antenna gain for access points	9 [dBi]
Antenna gain for attack stations	6 [dBi]
Range for sensor nodes	$\approx$ 25 [m]
Range for access points	$\approx$ 158 [m]
Range for attack stations	$\approx$ 54 [m]
Radio propagation model	Log-Distance Path Loss Model

TABLE III. ENERGY PARAMETERS

Parameter	Value
Initial voltage of sensor node	10 [Volt]
Initial voltage of access points	10 [Volt]
Initial voltage of attack stations	10 [Volt]
Transmit power for sensor nodes	3 [dBm]
Transmit power for access points	30 [dBm]
Transmit power for attack stations	15 [dBm]
Idle mode power factor	0.273
Transmission power factor	0.380
Receiving power factor	0.313
Sleep power factor	0.033

Consumption. These metrics provide valuable insights into network functionality and efficiency.

- Packet Delivery Ratio (PDR): is a crucial performance metric for the network, representing the percentage of packets delivered by the source node to the destination node, as shown in (1) [16]. A high PDR indicates better performance.

$$PDR[\%] = \frac{TRP}{TTP} \times 100 \quad (1)$$

  Where, PDR (Packet Delivery Ratio) represents the percentage of packets delivered, TRP (Total Received Packets) represents the total number of received packets, and TTP (Total Transmitted Packets) represents the total number of transmitted packets.

- Packet Loss Ratio (PLR): represents the percentage of packets failed to reach their destination or recipient, as shown in (2) [16]. A lower PLR indicates better performance.

$$PLR[\%] = \frac{TLP}{TTP} \times 100 \quad (2)$$

  Where, PLR (Packet Loss Ratio) represents the percentage of packet loss, and TLP (Total Lost Packets) represents the total number of lost packets.

- Average End to End Delay (E2EAD): indicates the average delay between the sent time of packet and the received time at the destination, as shown in (3) [16]. The delay is measured in milliseconds and provides an average delay for all packets created by the entire network.

$$E2EAD[ms] = \frac{TRPT - TSPT}{TRPN} \quad (3)$$

Where, E2EAD (End to End Average Delay) represents the average delay from end to end, TRPT (Total Received Packet Time) represents the total time taken to receive packets, TSPT (Total Sent Packets Time) represents the total time taken to send packets, and TRPN (Total Received Packets Number) represents the total number of received packets.

- Energy Consumption: represents the amount of energy consumed by a node or station used in emulation. It is calculated using the Mininet-WiFi emulator (included in the energy.py library), as shown in (4), and is measured in mill joules. Better network performance is achieved when less energy is consumed during the emulation period.

$$EC[mJ] = T \times V \times I \times F \tag{4}$$

Where, EC represents the amount of energy consumption, T represents the time of operation, V represents the voltage, I represents the current flowing through the node, and F is a power factor represents depending on the state of the node as mentioned in TABLE III.

- Round-Trip Time (RTT): is a metric used to calculate the time taken for a packet to travel to and from the controller, measuring the standard deviation or delay across all packets recorded during the evaluation period. RTT metrics help calculate error rate and packet loss rate [21].

2. *Machine Learning Algorithms Performance Metrics:* While the Confusion Matrix serves as a powerful tool for evaluating the performance of classification-based machine learning algorithms, it presents limitations for comparing the efficacy of Intrusion Detection Systems. To address this, alternative performance metrics have been developed that leverage the data contained within the Confusion Matrix. These metrics, often expressed as numerical values or percentages within a 0 to 1 range [2]. Here some performance metrics used to evaluate machine learning algorithms:

- Classification Rate (CR): defined as the ratio of correctly classified cases to the total number of cases, as shown in (5) [2].

$$CR = \frac{TP+TN}{TP+FP+TN+FN} \tag{5}$$

Where

  - TP (True Positive) represents the number of attempted breaches that were correctly detected.
  - FP (False Positive) represents the number of intrusion alerts while the system behavior was normal.
  - TN (True Negative) represents the probability that an actual negative will test negative.
  - FN (False Negative) represents the number of attacks on the system that the intrusion detection system could not detect.

- Precision (P): It is the proportion of predicted anomalies that are actually anomalies, as shown in (6) [2].

$$P = \frac{TP}{TP+FP} \tag{6}$$

- False Positive Rate (FPR): represents the number of cases of normal behavior that were considered an attempted intrusion to the total number of cases of normal behavior, as shown in (7) [22].

$$FPR = \frac{FP}{FP+TN} \tag{7}$$

- False Negative Rate (FNR): represents the number of cases of behavior resulting from an attack that were considered normal behavior to the total number of attack cases, as shown in equation (8) [22].

$$FNR = \frac{FN}{FN+TP} \tag{8}$$

D. Main Result

The impact of Denial of Service (DoS) attacks on network performance was evaluated by comparing network behaviour in both the absence and presence of attacks. This analysis utilized two key metrics: graphical representations of energy consumption characteristics and Round-Trip Time (RTT) analysis for packets transmitted from sensor nodes to access points. Figures 7 and 8 depict the energy consumption of the first access point (AP1) and second access point (AP2), respectively. These figures compare scenarios with and without DoS attacks, as well as scenarios with and without the proposed implemented algorithm. The results clearly demonstrate that the proposed algorithm significantly improves energy conservation and network longevity compared to scenarios where DoS attacks occur without the algorithm's intervention. Notably, energy consumption levels approach those observed during normal operation (no attacks) after the algorithm is implemented. TABLE IV. presents the average power consumption of sensing nodes under various conditions: no DoS attacks,

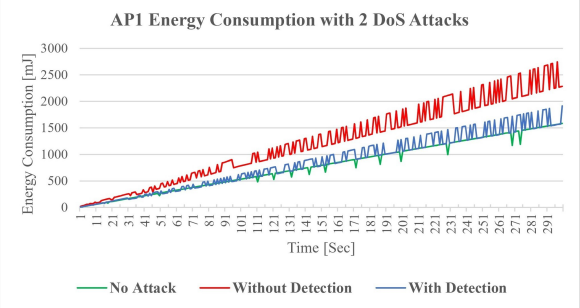

Fig. 7. The energy consumption of the first access point (AP1) under the proposed algorithm.

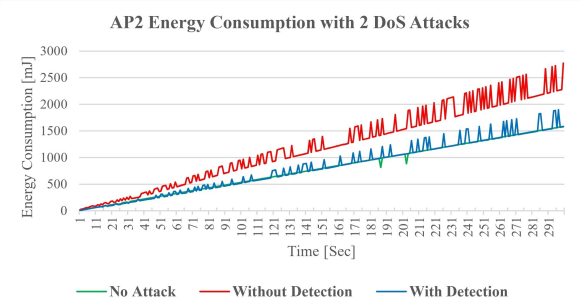

Fig. 8. The energy consumption of the second access point (AP2) under the proposed algorithm.

two DoS attacks per access point (without the proposed algorithm), and two DoS attacks per access point (with the proposed algorithm). The table clearly demonstrates that the proposed algorithm significantly reduces the average power consumption of sensing nodes. Furthermore, TABLE V. summarizes the impact of DoS attacks on RTT metric analysis for packets transmitted from sensing nodes to access points. It compares scenarios with no DoS attacks, two DoS attacks per access point without the proposed algorithm, and two DoS attacks per access point with the proposed algorithm.

TABLE IV. AVERAGE POWER CONSUMPTION FOR SENSING NODES

Scenario	Value [mJ]
No attacks	795.921
Two attacks at each access point without the proposed algorithm	1005.973
Two attacks at each access point with the proposed algorithm	924.706

TABLE V. SUMMARIZATION OF THE AVERAGE RESULTS

	No Attacks	Without Proposed Algorithm	With Proposed Algorithm
# Sent packets	791	800	800
# Received packets	9	12	582
# Lost packets	99%	788	218
PDR%	1%	1.45%	72.74%
PLR%	7.099	98.55%	27.26%
Average End-End Delay	791 [ms]	12016.059 [ms]	296.855 [ms]

E. Discussion

The proposed machine learning-based algorithm for DoS and DDoS attack detection and prevention in SDWSN exhibits a significant improvement for network performance and resource efficiency compared to scenarios without attack mitigation. Energy Consumption Reduction: As illustrated in Fig. 6 and Fig. 7, this algorithm aligns network behaviour closer to its natural operating state during non-attack periods. Furthermore, TABLE IV. demonstrates that the algorithm reduces the average energy consumption per sensing node within the WSN, thereby extending the network's overall lifespan. Enhanced Packet Delivery: The proposed algorithm demonstrably improves network performance under attack conditions. Packet delivery ratio (PDR) increases significantly from 1.45% to 72.74% with the algorithm deployed, as shown in TABLE V. This represents a substantial improvement compared to scenarios without the algorithm in place. Similarly, the Average End-to-End Delay (average packet delivery time from source node to access point/controller) is significantly reduced, dropping from 12,016 milliseconds to approximately 297 milliseconds with the algorithm. These improvements directly translate to more efficient network resource management and enhanced overall network performance, making it more suitable for diverse IoT applications that demand reliable data transmission. By providing accurate and timely data, the algorithm empowers prompt and effective decision-making. Real-Time Attack Detection and Preven-

TABLE VI. PERFORMANCE CHARACTERISTICS OF ML ALGORITHM

FNR	FPR	P	CR
0.000	0.000	1.000	1.000

tion: TABLE VI. summarizes the performance characteristics of the machine learning algorithms integrated within the proposed solution. These algorithms demonstrate the capability of predicting and stopping the attacks in real-time. During performance testing, the Detection Rate reached 100%, and the False Negative Rate remained at zero, resulting in an overall detection accuracy of 100%, and indicating perfect accuracy for the machine learning component of the algorithm.

F. Comparison with Relevant Studies

A comprehensive comparison with relevant literature is presented in TABLE VII, this table details the methodology employed for comparison, including the research concerns, algorithms utilized, datasets used, and achieved detection accuracy. Our work proposes a unique approach that addresses the challenges of attack detection, packet delivery, delay time, and energy consumption within a unified framework, setting it apart from previous studies. Notably, the proposed algorithm achieves superior detection accuracy when evaluated on the generated dataset.

G. Conclusion and Future Work

This paper proposes a novel machine learning-based algorithm for real-time detection and prevention of Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks in Software-Defined Wireless Sensor Networks (SDWSN). The algorithm leverages a diverse set of features including packet type, size, rate, source, destination, and statistically derived metrics to train a classifier. This classifier effectively distinguishes between legitimate network traffic (Benign Traffic) and malicious attack traffic (DDoS Traffic). Our proposed algorithm demonstrates significant improvements in network performance and energy efficiency. The classifier's accurate traffic classification translates to an enhanced Packet Delivery Ratio (PDR) and reduced End-to-End Delay. Furthermore, the algorithm's real-time detection and prevention capabilities have been validated with a 100% detection rate and zero false negatives. Future work should focus on evaluating the algorithm's performance with multiple datasets and exploring the use of more complex deep learning techniques in developing a routing protocol based on the proposed algorithm. These efforts hold promise for optimizing sensor node energy consumption and enhancing network resource management. Ultimately, these advancements will contribute to a more efficient and reliable network infrastructure, fostering the development of a wider range of robust IoT applications.

ACKNOWLEDGMENT

I offer my gratitude to my family for their support and patience. I also thank my colleagues in Department of Electronics and Communications Engineering, Faculty of Mechanical and Electrical Engineering, Damascus University, for their valuable comments and suggestions for improving the quality of my research in several ways.

CONFLICT OF INTEREST

The authors have no conflict of relevant interest to this article.

REFERENCES

[1] I. Mathebula, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, "Analysis of sdn-based security challenges and solution approaches for sdwsn usage," in *2019 IEEE 28th international symposium on industrial electronics (ISIE)*, pp. 1288–1293, IEEE, 2019.

[2] S. M. W. Umba, A. M. Abu-Mahfouz, T. Ramotsoela, and G. P. Hancke, "Comparative study of artificial intelligence based intrusion detection for software-defined wireless sensor networks," in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pp. 2220–2225, IEEE, 2019.

[3] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE access*, vol. 5, pp. 1872–1899, 2017.

[4] D. Kreutz, F. M. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pp. 55–60, 2013.

[5] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for sdn? implementation challenges for software-defined networks," *IEEE Communications magazine*, vol. 51, no. 7, pp. 36–43, 2013.

[6] Z. Yan and C. Prehofer, "Autonomic trust management for a component-based software system," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 810–823, 2010.

TABLE VII. COMPARISON OF RESULTS WITH RELEVANT STUDIES

Paper	Research Concerns				Algorithms	Dataset	Accuracy
	Attacks Detection	Packet Delivery	Time Delay	Energy Consumption			
[2]	✓		✓		NB-DT-Deep ANN	NSL-KDD	99.98%
[8]	✓	✓			DT-RF-XGBoost-KNN	WSN-DS	99.5%
[9]	✓				CP	Generated	93%
[10]	✓				DT	Smartcity	97.39%
[11]	✓		✓		DT-RF-XGBoost	NSL-KDD	95.5%
[12]	✓				SVM	Generated	99.26%
[13]	✓	✓	✓		CP	Generated	98%
[14]	✓				DT-SVM-LR	KDD-Cup99	77.43%
[15]	✓				HHT-CNN	MAWI	99.8%
[16]	✓	✓	✓	✓	RF-KNN	NSL-KDD	X
This Study	✓	✓	✓	✓	DT	Generated	100%

[7] S. W. Pritchard, G. P. Hancke, and A. M. Abu-Mahfouz, "Security in software-defined wireless sensor networks: Threats, challenges and potential solutions," in *2017 IEEE 15th international conference on industrial informatics (INDIN)*, pp. 168–173, IEEE, 2017.

[8] M. A. Elsadig, "Detection of denial-of-service attack in wireless sensor networks: A lightweight machine learning approach," *IEEE Access*, 2023.

[9] G. A. N. Segura, A. Chorti, and C. B. Margi, "Distributed dos attack detection in sdn: Tradeoffs in resource constrained wireless networks," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 131–135, IEEE, 2021.

[10] Y.-W. Chen, J.-P. Sheu, Y.-C. Kuo, and N. Van Cuong, "Design and implementation of iot ddos attacks detection system based on machine learning," in *2020 European Conference on Networks and Communications (EuCNC)*, pp. 122–127, IEEE, 2020.

[11] A. O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, p. 111, 2021.

[12] V. Kumar Singh, *DDOS attack detection and mitigation using statistical and machine learning methods in SDN*. PhD thesis, Dublin, National College of Ireland, 2020.

[13] G. A. N. Segura, S. Skaperas, A. Chorti, L. Mamatas, and C. B. Margi, "Denial of service attacks detection in software-defined wireless sensor networks," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–7, IEEE, 2020.

[14] A. T. Kgogo, *Intrusion detection system in software defined wireless sensor networks*. PhD thesis, North-West University (South Africa), 2019.

[15] Y. Liu, D. Sun, R. Zhang, and W. Li, "A method for detecting ldos attacks in sdwsn based on compressed hilbert–huang transform and convolutional neural networks," *Sensors*, vol. 23, no. 10, p. 4745, 2023.

[16] K. Indira and U. Sakthi, "A hybrid intrusion detection system for sdwsn using random forest (rf) machine learning approach," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, 2020.

[17] B. B. Letswamotse, R. Malekian, C.-Y. Chen, and K. M. Modieginyane, "Software defined wireless sensor networks (sdwsn): a review on efficient resources, applica-

tions and technologies," *Journal of Internet Technology*, vol. 19, no. 5, pp. 1303–1313, 2018.

[18] H. Mostafaei and M. Menth, "Software-defined wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 119, pp. 42–56, 2018.

[19] N. Ahmed, A. b. Ngadi, J. M. Sharif, S. Hussain, M. Uddin, M. S. Rathore, J. Iqbal, M. Abdelhaq, R. Alsaqour, S. S. Ullah, *et al.*, "Network threat detection using machine/deep learning in sdn-based platforms: a comprehensive analysis of state-of-the-art solutions, discussion, challenges, and future research direction," *Sensors*, vol. 22, no. 20, p. 7896, 2022.

[20] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2317–2346, 2015.

[21] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "Fragmentation-based distributed control system for software-defined wireless sensor networks," *IEEE transactions on industrial informatics*, vol. 15, no. 2, pp. 901–910, 2018.

[22] G. G. Gebremariam, J. Panda, and S. Indu, "Localization and detection of multiple attacks in wireless sensor networks using artificial neural network," *Wireless Communications and Mobile Computing*, vol. 2023, no. 1, p. 2744706, 2023.