sensors

MDPI

Article

Feature-Selection and Mutual-Clustering Approaches to Improve DoS Detection and Maintain WSNs' Lifetime

Rami Ahmad [1], Raniyah Wazirali [2,*], Qusay Bsoul [3], Tarik Abu-Ain [2] and Waleed Abu-Ain [4]

[1] The School of Information Technology, Sebha University, Sebha 71, Libya; r_a_sh2001@yahoo.com
[2] College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia; t.aboain@seu.edu.sa
[3] Faculty of Science and Information Technology, Irbid National University, Irbid 21110, Jordan; q.bsoul@inu.edu.jo
[4] College of Community, Taibah University, Badr 46354, Saudi Arabia; wabuain@taibahu.edu.sa
* Correspondence: r.wazirali@seu.edu.sa

Abstract: Wireless Sensor Networks (WSNs) continue to face two major challenges: energy and security. As a consequence, one of the WSN-related security tasks is to protect them from Denial of Service (DoS) and Distributed DoS (DDoS) attacks. Machine learning-based systems are the only viable option for these types of attacks, as traditional packet deep scan systems depend on open field inspection in transport layer security packets and the open field encryption trend. Moreover, network data traffic will become more complex due to increases in the amount of data transmitted between WSN nodes as a result of increasing usage in the future. Therefore, there is a need to use feature selection techniques with machine learning in order to determine which data in the DoS detection process are most important. This paper examined techniques for improving DoS anomalies detection along with power reservation in WSNs to balance them. A new clustering technique was introduced, called the CH_Rotations algorithm, to improve anomaly detection efficiency over a WSN's lifetime. Furthermore, the use of feature selection techniques with machine learning algorithms in examining WSN node traffic and the effect of these techniques on the lifetime of WSNs was evaluated. The evaluation results showed that the Water Cycle (WC) feature selection displayed the best average performance accuracy of 2%, 5%, 3%, and 3% greater than Particle Swarm Optimization (PSO), Simulated Annealing (SA), Harmony Search (HS), and Genetic Algorithm (GA), respectively. Moreover, the WC with Decision Tree (DT) classifier showed 100% accuracy with only one feature. In addition, the CH_Rotations algorithm improved network lifetime by 30% compared to the standard LEACH protocol. Network lifetime using the WC + DT technique was reduced by 5% compared to other WC + DT-free scenarios.

Keywords: IDS; machine learning; DoS; WSN security; feature selection; LEACH

Citation: Ahmad, R.; Wazirali, R.; Bsoul, Q.; Abu-Ain, T.; Abu-Ain, W. Feature-Selection and Mutual-Clustering Approaches to Improve DoS Detection and Maintain WSNs' Lifetime. *Sensors* **2021**, *21*, 4821. https://doi.org/10.3390/s21144821

Academic Editors: José L. Hernández Ramos, Georgios Kambourakis, Erol Gelenbe and Gianmarco Baldini

Received: 23 June 2021
Accepted: 9 July 2021
Published: 15 July 2021

1. Introduction

Wireless network technology is at the heart of the growth of the Internet of Things (IoT). This is because wireless networks are critical for transmitting interactive data from devices to humans, as well as between devices [1]. These devices are part of automation and control systems, embedded systems, Wireless Sensor Networks (WSNs), and other systems that exchange data in a variety of environments without requiring human intervention. Most applications that use these devices are made up of perception, network, and application layers [2]. The application and network layers are mostly executed in high-powered devices, while the perception layer is mostly executed in low-powered devices to keep them running as long as possible, particularly when using systems with limited battery life. Since perception devices depend on public wireless networks, the perception layer is considered one of the most sensitive topics in need of attention, particularly to protection against attack [3]. Accreditation within this layer contributes to many issues in WSN

architecture. One of these issues applies to security, privacy, and availability within the perception layer [4]. Attackers can listen in on radio transmissions, send fake messages over communication channels, and alter received data packets [5,6]. Moreover, they can use compromised WSN nodes with similar hardware resources to legitimate network nodes [7]. In addition, an attacker might be capable of stopping WSN node services by using various attacks such as sinkhole, wormhole, hello flood, Sybil, and Denial of Service (DoS) [8].

DoS and Distributed DoS (DDoS) attacks are one of the most common and dangerous threats to WSN security, as they occur when several compromised WSN nodes are infected by malicious WSN nodes at the same time under the control of a single attacker by overwhelming the target WSN nodes with bogus requests. This depletes their resources and forces them to refuse services to legitimate WSN nodes [9]. Therefore, in this paper, we will focus on slowing down DoS and DDoS attacks in WSNs with minimum power consumption and good detection accuracy.

Due to the inability to prevent or completely stop such types of attacks, Intrusion Detection Systems (IDS) are used to discover suspicious or abnormal activities and alert the WSN nodes [10]. Signature-based and anomaly-based intrusion detection are the two types of intrusion detection. In an anomaly pattern, the system must regularly track network access and compare ongoing WSN operations to normal traffic patterns [10,11]. However, the anomaly detection technique needs to examine and analyze each transmission packet in detail. Thus, it consumes energy and CPU, which are weaknesses of WSN nodes. As a result, a variety of techniques have been used to boost DoS detection efficiency using both active and passive methods. Supervised machine learning algorithms are one such technique used to predict and classify DoS and DDoS attacks [12–15]. Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Deep-Learning (DL) classifier, and Naive Bayes (NB) are common algorithms for this purpose [16]. The authors in [1,2,9,12,14,17,18] used various deep learning mechanisms, and their results in terms of detection precision, mean squared error, and sensitivity were satisfactory, but none of them addressed the impact of their proposals on WSNs, such as node power consumption and network lifetime. The problem with these techniques is the amount of data they require for the training and testing process, and WSN nodes' inability to manage such a huge number of dimension records. Therefore, another technology can be used in conjunction with machine learning classifiers is a feature selection algorithm. This is used to determine which data features are most important in the IDS process [19], and helps in understanding data, minimizing computation requirements, and improving prediction performance. Examples of feature selection technologies are Water Cycle (WC) [20], Particle Swarm Optimization (PSO) [21], Simulated Annealing (SA) [22], Harmony Search (HS) [19], and Genetic Algorithm (GA) [23].

Since the main objective of this paper is to find the best solution to protect the perception layer from DoS attacks while taking into account the capabilities of these devices, we will analyze the effects of various machine learning algorithms along with different feature selection techniques in order to balance their performance accuracy for DoS detection with their consumption energy. As some studies, such as [24,25], have shown that machine learning techniques (logistic regression, SVM, and DT) are more appropriate for real-world deployment of wireless devices than a deep learning mechanism, machine learning techniques are favored. That is due to the need for a significant amount of training data for deep learning algorithms in order to provide high-accuracy classification performance. On the other hand, the IDS technique in WSNs must be compatible with their protocols. In WSNs there are different routing protocols that are used to transmit data packets to the Access Point (AP), such as Low Power and Lossy Networks (RPL) in 6LoWPAN [26], Adhoc On Demand Distance Vector (AODV) routing in ZigBee [27], and Low Energy Aware Cluster Hierarchy (LEACH) [28]. LEACH is one of the most widely used hierarchical routing protocols due to its limited power consumption, and is used in this study due to the fact that the WSN-Data Set [29] on which we rely for analysis was collected according to this protocol. It aims to boost energy efficiency by using a rotation-based Cluster Head

(CH) selection process with a random number. However, the reliance on randomness in rotation-based CH selection is considered a weakness in the LEACH protocol, and various studies [30–34] have improved its performance. It remains possible to improve the protocol further in terms of increasing its efficiency in selecting the appropriate CH in each loop, and increasing the network lifetime. By summarizing previous studies and applications, we see that there has been an increase in the use of WSNs recently, that these devices play multiple roles, and the biggest challenges for WSNs remain security and energy. To overcome these challenges, several solutions that discuss data security performance problems have been proposed [10–13,15,35,36] that do not consider the impact on wireless power consumption. Moreover, other studies have discussed the concept of energy conservation from a data security-independent perspective [30–34]. Therefore, in this study, we will create a complete picture that combines both elements (security and energy), by determining how best to raise the level of security in WSNs and protect them from DoS attacks with minimal energy consumption. We modified the cluster protocol to increase network performance efficiency while saving energy and increasing DoS detection. Moreover, in certain situations, WSN nodes must be used in sensitive areas and we cannot power them up regularly; thus, we sought to save as much energy as possible while preserving security. In this work, we will contribute to improving DoS detection and reduce power consumption in WSNs. The following are the main contributions of this work:

1. We modified the LEACH protocol to reduce randomness in determining the CH nodes by adding other factors such as node residual power, distance between nodes, and distance to AP in order to increase efficiency and extend the lifetime of the WSN.
2. We analyzed the effect of feature selection techniques along with machine learning algorithms on the accuracy of DoS detection.
3. We studied the effect of this modified LEACH protocol on the best-performing technique in terms of network lifetime. This is in contrast to related studies that have improved the accuracy of DoS detection over WSN without analyzing its actual effect on sensors.

The rest of our paper is set out as follows. The second section covers related work in WSNs, DoS attacks, feature selection techniques, and machine learning algorithms. The methodology, environmental development, cluster management, feature selection machine learning test, and decision-making are all covered in Section 3. Data collection and arrangement are covered in Section 4. Section 5 delves into the implementation and assessment of complexity analysis in feature selection and machine learning techniques, as well as the lifetime of WSNs. In Section 6, the paper's conclusions and directions for future work are presented.

2. Background and Related Works

In this section we will provide a literature review and technical background concerning WSNs, clustering efficiency and intrusion detection in these networks, and machine learning algorithms in intrusion detection.

2.1. LEACH Protocol Energy Efficiency in WSNs

A WSN is a radio access spectrum that uses a frequency of 2.4 GHz which is designed to work in low-power, low-range, and low-processor circumstances. Every WSN has different WSN nodes that communicate with each other and their Access Point (AP). Data are transferred from WSN node to AP through different routing protocols. One of these protocols, LEACH, is used to reduce the power consumption of the WSN nodes, which have limited power capacity. The key concept of this protocol is to spread the energy load of the entire network equally to each WSN node by selecting WSN CH nodes at random in each loop. Due to the CH-shifting technique used in the LEACH protocol, all WSN nodes are believed to have a similar survival time [37]. The CH selection process in LEACH takes place in two phases in each loop: CH establishment and steady-state [38]. In the CH establishment phase, each WSN node generates a random value between 0 and 1, and then

starts computing the threshold formula *Thrd(n).* Subsequently, in each WSN node if the random chosen value is less than *Thrd(n),* it becomes a WSN CH node and it will send request messages to neighboring ordinary WSN nodes. The formula *Thrd (n)* is illustrated in (1)

$$Thrd(n) = \begin{cases} \frac{1}{1-p\left(r \bmod \frac{1}{p}\right)} & n \in G \\ 0 & otherwise \end{cases} \quad (1)$$

where n is the WSN node, p is the probability that the n becomes CH, r is the current loop number, and G represents the group of WSN nodes that have not joined in the previous CH selection loops $(1/p)$.

We conclude from this that a WSN node that is chosen as CH for the r loop will not participate in the loops following r. Thus, all WSN nodes have the same chance to be CH nodes. In the steady-state phase, the ordinary WSN nodes in each CH transmit the data to their cluster WSN node by using the Time-Division Multiple Access (TDMA) schedule. The LEACH protocol architecture is illustrated in Figure 1.

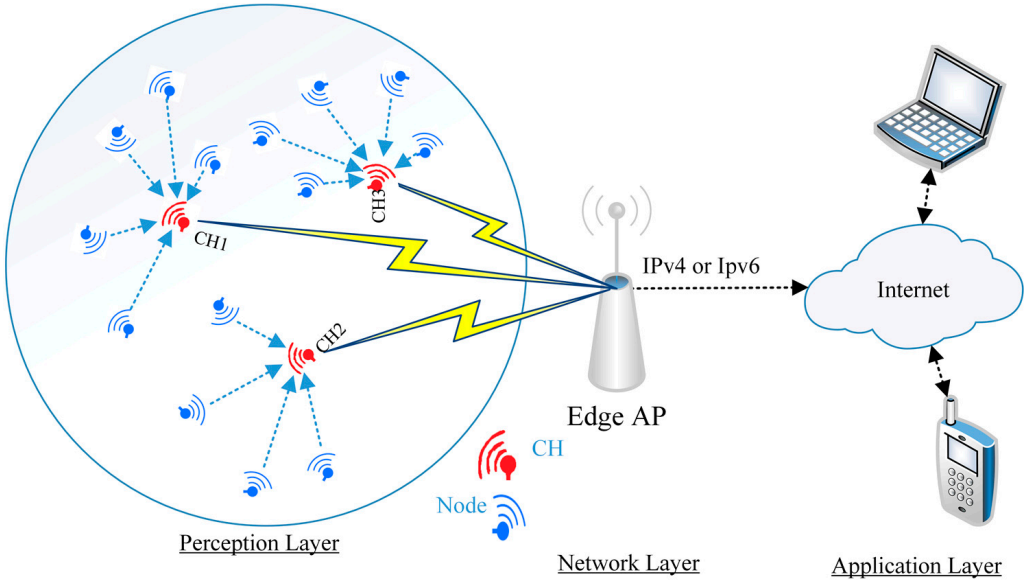

Figure 1. The architecture of the LEACH protocol in WSNs.

As shown in Figure 1, the WSNs are responsible for drawing the network topology and routing table in the perception layer using different protocols [39] as explained earlier. Each WSN node associates with the set of its neighboring peer WSN nodes, after which the WSN node starts collecting data from different locations and forwarding the data to the network layer (edge-AP). Therefore, much research has been done to improve the LEACH protocol in the perception layer and thus reduce the WSN nodes' consumption power. The authors in [32] altered the LEACH protocol through using a number of WSN node links and factoring in the cost of the path for choosing CH nodes rather than a random selection technique. In [33], the authors studied the effect of shortening the distance between ordinary WSN nodes and CH nodes from one side and the effect of shortening the distance between CH nodes and AP from the other side on network lifetime. The result showed a reduction in power consumption. The authors in [40] optimized a new routing protocol through employing three complementary steps in each CH selection loop. First, it determined the optimal CH numbers used in WSNs, then it created a Voronoi diagram in between neighboring WSN nodes to determine CH nodes. Finally, it used an ant colony algorithm to optimize the multi-hop routing protocol. However, through using all of these steps in every setting, the CH loop exhausted the network power and CPU. In addition, [31] created an algorithm called a standardized experimental bat has been proposed to improve CH selection in each loop. This algorithm worked on the basis of a

search balance between local and global WSN nodes in a WSN mobile node environment. To examine the feasibility of the use of Global Positioning System (GPS), the authors in [30] divided the area of the network belonging to the coverage area of the AP into different zones. In each CH determination loop, one WSN node was identified as a CH node in each zone, and the role was rotated between WSN nodes that occurred in the same zone. However, this technology does not deal with a large number of WSN nodes [41]. Therefore, the LEACH protocol remains amenable to improvement through increased efficiency in selecting the appropriate CH in each loop, and that will be one of the contributions of this paper.

A User Datagram Protocol (UDP) is used as the data transmission protocol in WSNs to reduce the packet's clustering complexity and reduce CPU overhead. Moreover, to secure data transmission over UDP, the Datagram Transport Layer Security (DTLS) protocol is used on top of UDP [42]. However, these WSN nodes are designed to operate in various untrusted surroundings, which are not periodically monitored. This makes WSN nodes vulnerable to various security attacks, especially if they are related to important and sensitive data [43]. Moreover, with regard to WSN nodes' operational boundaries in CPU power and energy [44], it is sometimes difficult to provide a charger for them in these conditions.

2.2. DoS in WSNs

As mentioned previously, the main objective of the DoS or DDoS attack is to affect the network's availability by disrupting services and network performance. Therefore, the effect of this type of attack varies depending on the network layer stack [5]. Since wireless sensor networks have stacks of five network layers, each layer is vulnerable to different types of attacks [10,45]. The attached Table 1 shows each layer and type of DDoS attack that can be represented.

Table 1. Taxonomy of WSN DDoS attacks.

WSN Layers	Description
Perception layer	Jamming Tempering Scheduling
Perception MAC Layer	Collision Exhaustion
Network or Routing Layer	Blackhole Grayhole Hello
Transport Layer	Flooding
Application Layer	Overwhelming nodes Path-Based DoS

Blackhole and grayhole DDoS attacks affect the routing protocol in layer three by declaring the attacker node itself as the cluster head, and we will discuss the cluster head functionality in detail in the clustering management subsection. By comparison, a flooding attack affects WSN availability by sending a large number of advertising messages to cluster heads. In scheduling attacks, the perception and MAC layers' activity is targeted by changing the broadcast channel schedule to a unicast channel schedule. This change leads to packet collision and later data loss [29].

Several researchers have attempted to mitigate DDoS or DoS in WSNs. The Message Authentication System (MAS) algorithm was used by the authors in [7] to locate and delete DoS attacks. The proposal divides WSN nodes into different clusters, with each cluster head using the MAS method to separate legitimate traffic from phishing scams. The authors in [20,46] improved the k-means clustering scheme for finding DDoS and misdirection attacks. To detect attacks in a home WSN, the authors of [47] used user-behaviors learning analysis. In [18], the authors used Restricted Boltzmann Machine-based Clustered IDS

(RBC-IDS), a deep learning-based technique for tracking sensitive infrastructure utilizing three hidden layers for potential intruders. Authors in [48] used a genetic algorithm combined with a Multi-Layer Perceptron algorithm to improve detection methods. In [49], the authors presented a novel swarm optimization algorithm for clustering WSN nodes and then used a VSM classifier to detect DoS attacks in each cluster. However, rather than anti-attack initiatives, the focus of this study will be on analyzing DoS attack detection efficiency. In this work we will look at four types of DoS attacks, namely Blackhole, Grayhole, Flooding, and Scheduling.

2.3. Feature Selection and Machine Learning in Intrusion Detection Approach

Feature subset selection is a common problem in network detection [23] due to the high dimensionality of features of the sensor. Therefore, the development of new approaches to handle feature selection is still an active area of research, particularly for feature identification. The purpose of feature selection is to improve performance in areas such as accuracy, data visualization and simplification for model selection, and dimensionality reduction to remove noise and irrelevant features [50]. The selection of the features and distribution of the data have a high impact on the performance of classifier algorithms. These tend to obtain local minima rather than the global minimum. The obtained results are often very good, especially when the initial features are fairly far apart. This is because the algorithm can usually distinguish the main category or class in a given data set. Moreover, the classifier algorithm's main process and the quality of feature identification are both affected by the initial feature selection. Thus, the initial features can enhance the quality of the results [51]. There are two methods of feature selection: filter and wrapper. The filter method ignores feature dependencies [50,51]. The wrapper method uses optimization in feature selection and thus can provide a good initial choice of feature and perform better as features are refined and the best feature selection is found [50].

During the past two decades, a large number of classical optimization techniques have been developed to improve function, including the bat algorithm, biogeography-based optimization, bacterial research optimization, modified Lagrange approach, synthetic physics improvement, artificial plant improvement algorithm, generative models [52], PSO, GA, and cuckoo search. Nature-inspired heuristic algorithms have been used extensively, and more recently, the water cycle (WC) method [53], which is influenced by the behavior of rivers and seas in nature, has been used. The authors in [54] employed ant colony optimization as a feature selection method for classifiers and used the absolute value [55] as the similarity to optimize between features. The efficiency and effectiveness of the ant colony optimization were better than those of other feature selection methods used as classifiers. Abualigah et al. (2016) [56] employed a GA for feature selection and used the mean absolute difference as the similarity between features. They compared their proposed method with those of other feature selection methods and showed that their proposed method increased feature detection performance. Subsequently, Abualigah et al. [57] employed Particle Swarm Optimization (PSO) for feature selection. Their results showed that their proposed feature selection method outperformed other feature selection methods such as genetic and harmony search algorithms.

Additionally, feature selection techniques can be combined with machine learning algorithms to enhance the accuracy of results. Machine learning is a method that improves or learns from an interpretation or experience without requiring manual configuration. It can be divided into supervised and unsupervised learning. Classification and regression are two types of supervised learning. Statistical (SVM and Bayesian), logic-based (DT), instance-based (KNN), perceptron-based (deep learning (Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM), Convolutional Neural Networks (CNN)) and Multi-Layer Perceptron (MLP) learning are the different types of classification [58]. The primary goal of this learning methodology is to develop a model that defines the relationships and dependencies between input features and predicted objective outcomes [16]. As a result, supervised learning can be used to solve real world problems in WSNs, including fault and

anomaly detection. Table 2 shows the different types of detection methods and machine learning strategies used in attack detection for WSNs.

Table 2. Taxonomy of different machine learning algorithms in detection attacks.

Category	Reference	Technique	Dataset	Accuracy	Goals	Limitation
Statistical-based	[15]	Game theory and an autoregressive model	Live format simulator (Matlab)	81%	Reduce detection power consumption in the intrusion detection process	Accuracy is low
MLP	[23]	Sequential feature selection with MLP algorithm	NSL-KDD	99.7%	Reduce DDoS attacks	The proposal is not considered a WSN restriction
Statistical-based	[20]	K-medoid clustering technique	Live format simulator (NS-2)	-	Attacks detection	Accuracy unknown
Deep Learning	[58]	Deep Neural Network	WSN-DS	99%	Improve intrusion detection in IoT networks	The proposal's consumption makes it unsuitable for WSNs
Deep Learning	[13]	Deep Learning-based Defense Mechanism	Live format network forward packet	90%	Improve DDoS detection in WSNs	No energy consumption tested on WSNs
Statistical-based	[59]	Binary Logistic Regression (BLR)	Live format simulator (Monitoring Tool)	96–100%	Improve DoS detection in WSNs	Data features are few and do not cover the majority of common attacks
Deep Learning	[58]	Deep Neural Network	WSN-DS	99%	Improve intrusion detection in IoT networks	The proposal's consumption makes it unsuitable for WSNs

In this paper we will analyze WSNs' traffic based on different feature selection techniques combined with machine learning classifications to see how they perform in detecting DoS attacks.

3. Proposed Methodology

The main objective in this paper is to enhance the detection of anomalies in DoS with the lowest possible power consumption. First, we improve network performance through the use of the cluster protocol. Next, we combine lightweight feature selection with machine learning technologies to minimize DoS detection power cost and raise detection accuracy. Therefore, the proposal environment consists of three processes. The first is to aggregate WSN nodes into multiple clusters, each cluster having a CH in the WSN nodes environment. The LEACH standard protocol is updated with additional parameters to improve its performance, and the protocol generated from the modifications denoted CH_Rotations. In the following process, feature selection and machine learning testing approaches are used to analyze traffic in each CH node to distinguish between normal and abnormal incoming packets. The feature selection technique is used to detect the most important data features and also to reduce mathematical operations during each packet inspection at each WSN CH node. In the last process, CH makes a decision based on the results of the second process. The structure of the general proposal model is illustrated in Figure 2, and each of the processes will be discussed in the following subsections.

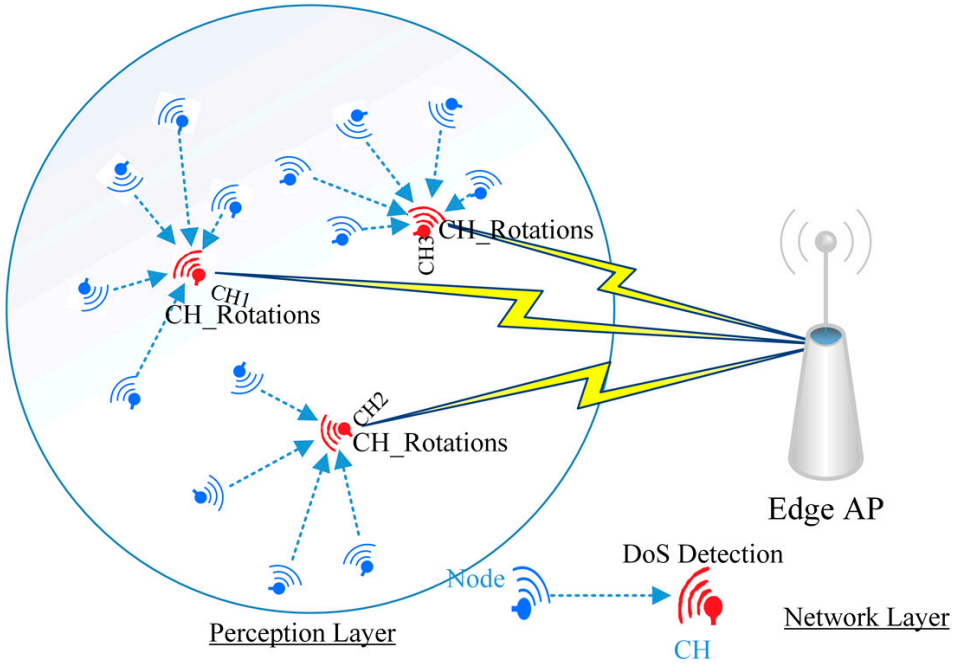

Figure 2. The structure of the general proposal model.

3.1. Clustering Management

The LEACH protocol increases power efficiency in WSNs by implementing CH_Rotations selection in each loop. However, the standard LEACH has various limitations regarding random CH selection, CH location relative to AP consideration, CH location relative to its ordinary nodes location, the number of ordinary nodes in each CH, and the number of CHs in each loop [38]. Therefore, we update the LEACH protocol to enhance its performance, as the distance between WSN nodes themselves and between them and the AP has a strong effect on power consumption during the communication process. The average distance between neighbour WSN nodes (β), the closest distance (d) to the AP, and the WSN node energy (E) are considered along with the LEACH parameters in the establishment phase to select the most suitable CH. Moreover, a new CH node can be selected in advance for the next loop, unlike in the standard protocol. The steady-state will be the same as the standard LEACH protocol. These alteration processes are depicted as follows:

1. Our work's computing radio energy (E) model is based on [60] for each WSN node. The primary source of power usage is correspondence between WSN nodes, so power consumption is proportional to the distance between the transmitter and the receiver. The transmitter consumes power to operate the electrical transmission circuit and amplification, while the receiver consumes power to operate the cellular electronics of this embodiment. The power law of the distance between the transmitter and receiver can be used to shape the propagation of electromagnetic waves. As a result, the transmitter circuit uses $P_{TX\text{-}elec}$ in proportion to the message transmitting size (q-bit) in terms of distance (d). The transmitter uses $P_{TX\text{-}amp}$ to amplify the signal in order to produce a reasonable signal-to-noise ratio. The radio model's cumulative power spent to transmit q-bit over distance d(P_{TX}) is proposed to be (2):

$$P_{TX}(q,d) = P_{TX-elec}(q) * P_{TX-amp}(q,d) \quad (2)$$

where $P_{TX\text{-}amp}$ is equivalent to either β_{fsm} in a free space model or to β_{trm} in two-ray ground propagation models depending on distance between transmitter and receiver. Furthermore, a WSN node is in charge of transmitting data messages to other WSN nodes. As a consequence, WSN nodes will accept messages from other

WSN nodes. Equation (3) can be used to measure the power needed to receive the q-bit message E_{RX}:

$$E_{RX}(q) = E_{RX-elec}(q) = E_{elec} \times q \quad (3)$$

where E_{elec} is the power absorbed by the transmitter and receiver per bit in nJ/bit, $E_{RX\text{-}elec}$ is the power dissipated by the receiver during q-bits reception, and q is a bit-message.

2. The threshold energy (E_T). A node with an energy value smaller than the threshold value will be removed from the CH selection process. Equation (4) is used to measure each node's threshold power (n):

$$E_T(n) = N_{adjacents} * q * E_{elec} + N_{adjacents} * q * [E_{elec} + \beta_{fsm}] * d^2 \quad (4)$$

where the node energy threshold is represented by $E_T(n)$, the number of neighbouring nodes is represented by $N_{adjacents}$, the free space model is β_{fsm}, and the interval between transmitter and receiver nodes is represented by d. Furthermore, each node (n) sends *data_message* with its ID and energy level to all adjacent WSN nodes, which is used to measure and update the adjacent WSN nodes table for each loop.

3. In each cluster, the mean distance (β) between candidate CH and its neighboring WSN nodes is important. If it is smaller, then the chances of selecting that nominated CH are higher. Equation (5) is used to calculate the value of β:

$$\beta(n) = \frac{\sum_{j=1,\, j\neq n}^{\Omega} d_{nj}}{\Omega} \quad (5)$$

where Ω is the number of WSN nodes located in each cluster. Moreover, Equation (6) can be used to work out the distance threshold for each WSN node:

$$d_T = \left(\sqrt{\frac{\beta_{fsm}}{\beta_{trm}}} \right) \quad (6)$$

4. The shortest path to AP (d_A) must be calculated. The Received Signal Strength Indicator (RSSI) between WSN nodes and the AP can be used to measure the d_A value. The AP can also use GPS to calculate the positions of all WSN nodes [61]. The ideal CH is evaluated using these parameters, depending on the highest weight value measured using (7):

$$d_T = \left(\omega = \frac{\Omega * 2E * \beta}{d_A} \right) \quad (7)$$

According to (7), the ideal value of CH has the highest energy, is located near the AP location, and is located at the middle of the other WSN nodes in each cluster. Moreover, we tried to give the residual energy parameter a higher score than the rest of the parameters due to its importance. Therefore, in this case, the consumption of communication energy between the WSN nodes and their CH, as well as between the CH and the AP, will be the lowest.

In this study, we use the AP to pick the CHs from the nodes deployed in the AP area. This collection is allocated to the numerous zoning regions covering the AP, after which CH_Rotations will be self-organized. The CH then sends the *adv_CH* message, which broadcasts its ID within its own domain. Algorithm 1 has a detailed discussion of CH_Rotations in the new LEACH protocol in each zone area.

The algorithm always starts by counting the loop and then in each cluster will search for the best CH to cluster group through computing ω for each WSN node.

Algorithm 1 CH_Rotations

1. Set E_T
2. **for** each loop **do**
3. **for** *each* $n \in (1, N)$ **do**
4. *Find E*
5. **if** $(E \leq E_T)$ **then**
6. *Calcuate d*
7. *Calcuate* d_A
8. *Calucate* β
9. *Compute* ω
10. **end if//**line 5
11. **end for/**/line 2
12. *A new CH is selected based on the highest* ω
13. *A new CH sends Adv-CH messages*
14. *The N nodes send Join_Req to new CH*
15. **end for**

3.2. Water Cycle Detection Approach

One of the essential aims of this study is to introduce the Water Cycle (WC) algorithm as a feature selection technology to identify the least number of attributes and achieve better accuracy in machine learning algorithms, as well as to reduce predictable features and evaluate the recommended extraction over the benchmark dataset along with the actual dataset. Similar to meta-heuristic algorithms, the recommended approach begins with an initial population named raindrops in which the best individual is chosen as a sea. Subsequently, several good raindrops "features" are selected as a river whereas the other raindrops are considered streams that flow into the sea as well as rivers. Therefore, on the basis of its volume flow, water is taken from riverbeds. Moreover, the amount of water in streams entering the sea or rivers varies from one stream to another. The rivers that flow into the sea are the hilliest sites [53,62].

The WC method utilizes several representations to code the entire F of the feature in a vector of length m, where m represents the number of features. Each portion of such a vector contains a label indicating whether the features are dropped or chosen. Figure 3 depicts an example of how solutions can be represented. In this case, 6 features (3, 4, 5, 6, and 8) are chosen while the others (1, 2, 7, 9, and 10) are dropped.

Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Feature 8	Feature 9	Feature 10
0	0	1	1	1	1	0	1	0	0

Figure 3. Representation of selected features.

3.2.1. Initial Features Development

The values of the dataset features are considered as an array. In the optimization terminologies of the practical swarm and the genetic algorithm, the array is called "Particle Position" or "Chromosome" respectively. Therefore, in the recommended approach, the label is "Raindrop Features" for an individual feature. In the M_{var} dimensional feature selection problem, the raindrop represents an array of $1 \times M_{var}$. This array is illustrated as follows:

$$Feature\ of\ Raindrop = [x_1, x_2, x_3, \ldots, x_M] \tag{8}$$

At the beginning of the feature selection, a candidate is represented by a $M_{pop} \times M_{var}$ raindrop size array (i.e., raindrop features). Thus, the random matrix x is provided as (columns and rows that make up the design variable quantity plus feature selection quantity):

$$Feature\ Raindrops = \begin{bmatrix} Raindrop_1 \\ Raindrop_2 \\ Raindrop_3 \\ \vdots \\ Raindrop_{M_{pop}} \end{bmatrix} * \begin{bmatrix} x_1^1 x_2^1 x_3^1 & \cdots & x_{M_{var}}^1 \\ \vdots & \ddots & \vdots \\ x_1^{M_{pop}} x_2^{M_{pop}} x_3^{M_{pop}} & \cdots & x_{M_{var}}^{M_{pop}} \end{bmatrix} \tag{9}$$

Each value of the decision variable $(x_1,\ x_2,\ x_3 \ldots\ x_{M_{var}})$ can be described as the following numbers (0 or 1), where M_{vars} and M_{pop} are the number of design variables as well as the number of raindrops (preliminary feature selection), respectively. Further M_{pop} raindrops are generated, thus the raindrop cost is achieved by evaluating the cost function (Cost) as follows:

$$Cost_i = f\left(x_1^i,\ x_2^i, \ldots x_{M_{var}}^i\right),\ i = 1, 2, 3, \ldots., M_{pop} \tag{10}$$

3.2.2. Cost of Solutions

All possible solutions are evaluated according to the fitness selection procedure along with classifier algorithms, namely KNN [63], DT [64], SVM [65], DL [21], and NB [59], to obtain the highest performance accuracy among the classification algorithms and features selected for each solution. For the purpose of maintaining an adequate balance between all the selected features in each of the minimum solutions and providing maximum accuracy for feature selection, the fitness function, i.e., objective function in (11) is used in the WC technique to evaluate solutions in M_{pop}:

$$fitness = \Phi\gamma_R\ (D) + \partial\ \frac{|R|}{|M|} \tag{11}$$

where $\gamma_R\ (D)$ is the rating of classification error for a given classifier; |R| is the total items in the selected subset, |M| is the total number of features in the dataset, ∂ and Φ are two parameters that represent the importance of the classification quality and subset length, $\partial \in [0, 1]$ and $\Phi = (1 - \alpha)$ [50,66].

Many M_{sr} are selected from among the best individuals (minimum values) of the sea and rivers. The raindrop of the lowest value represents the sea. As a matter of fact, M_{sr} represents the total quantity of rivers (i.e., the user-defined parameters) plus the individual sea as shown in (12). The remaining preliminary features (raindrops from the streams flowing to the rivers or directly to the sea) are calculated on the basis of (13).

$$M_{sr} = Number\ of\ Rivers + (Sea = 1) \tag{12}$$

$$M_{Raindrops} = M_{pop} - M_{sr} \tag{13}$$

To facilitate the allocation of raindrops for sea and rivers in terms of the flow density, this Equation is used:

$$M_{sn} = round\left\{\left|\frac{Cost_m}{\sum_{i=1}^{M_{sr}} Cost_i}\right| \times M_{Raindrops}\right\},\ m = 1,\ 2, \ldots,\ M_{sr} \tag{14}$$

where M_{Sr} represents the quantity of stream flowing into a given sea or river [53].

3.2.3. Stream Flow to Rivers or Sea

The streams of raindrops are generated and communicate with each other to generate new rivers. In fact, a number of streams might flow directly into the sea. All streams, as well as rivers, end up in the sea (best-chosen features). To illustrate, a stream moves towards a river which lies along a line linking them using a randomly defined distance which can be illustrated as follows:

$$X\epsilon\ (0, C \times di), C > 1 \tag{15}$$

C represents a value between 1 and 2 (closer to 2). The best value for C might be selected as 2. The existing distance between the river and stream is described as di. The X value in (15) matches a number that is randomly distributed between 0 and $(C \times di)$. When the C value is greater than 1, it enables streams to run towards the rivers. This idea might

also be utilized on the rivers that reach the sea. Therefore, the new location for rivers and streams might be represented as:

$$X_{Stream}^{i+1} = X_{Stream}^{i} + rand \times C \times \left(X_{River}^{i} - X_{Stream}^{i} \right) \tag{16}$$

$$X_{River}^{i+1} = X_{River}^{i} + rand \times C \times \left(X_{Sea}^{i} - X_{River}^{i} \right) \tag{17}$$

The *rand* represents a uniform number that is randomly assigned between the values of 0 and 1. Moreover, if the precision provided by the stream works better than the river connecting it, the position of the stream and the river are swapped (i.e., the stream becomes a river and vice versa). This exchange can also occur for the sea and rivers [53].

3.2.4. Evaporation Condition

Evaporation is one of the most important factors preventing the algorithm from rapid convergence (immature convergence) [53]. Generally, water evaporates from lakes and rivers while trees absorb and then release water via photosynthesis. The evaporated water rises to the atmosphere to form clouds, which in turn condense into rain under colder conditions, releasing water to the ground. Accordingly, the rain generates new streams that reach the rivers and these rivers also flow to the sea [67]. In the proposed approach, the evaporation process allows seawater to evaporate as streams/rivers flow into the sea. The following pseudo-code (18) shows how to determine if a river extends into the sea or not:

$$\begin{gathered} If \\ \left(\left| X_{Sea}^{i} - X_{River}^{i} \right| < di_{max} \right),\ i = 1,\ 2,\ 3, \ldots, M_{sr} - 1 \\ Evaporation\ and\ raining\ process \\ End \end{gathered} \tag{18}$$

where di_{max} represents a tiny number (near 0). If the area between a sea and river is smaller than a di_{max}, this indicates that the river joined the sea. In this case, the process of evaporation is used, and as a result of widespread evaporation, precipitation (rain) begins. A large di_{max} reduces the search while the small value stimulates the search intensity near the sea. As such, di_{max} controls the search intensity near the sea (best possible solution). Therefore, the di_{max} value decreases flexibly as shown below:

$$di_{max}^{i+1} = di_{max}^{i} - \frac{di_{max}^{i}}{maxiteration} \tag{19}$$

3.2.5. Raining Process

After the evaporation process is achieved, the precipitation process takes place. In the process of rain, new raindrops form streams at different locations (working equally with the mutation factor of the genetic algorithm). For the purpose of determining the positions of the newly produced streams, this Equation is used:

$$X_{Stream}^{new} = LB + rand \times (UB - LB) \tag{20}$$

where UB and LB are the upper and lower limits defined by the problem investigated, respectively.

Moreover, the best recently formed raindrop is considered a river flowing into the sea. The remaining new raindrops are thought to generate some new streams that flow into rivers or directly into the sea. In order to enhance the convergence rate, as well as the computational performance of the algorithm for the specific problems, Equation (19) is only used for flows that take place directly to the sea. Equation (21) aims to encourage the establishment of watercourses heading directly to the sea and to promote searches near the sea (optimal features) in the potential area of specific problems [67]:

$$X_{Stream}^{\text{new}} = X_{sea} + \sqrt{\mu} \times rand\ (1,\ M_{var}) \tag{21}$$

where μ is the parameter indicating the extent to which the near-sea area is surveyed, and *Rand* is the naturally distributed random number. A greater value of μ increases the potential to exit the region, while a smaller value of μ causes the algorithm to explore in a smaller near-sea area. There is a suitable value for μ at 0.1. The term $\sqrt{\mu}$ in (21) mathematically describes the standard deviation. Hence, μ defines the concept of variance. Depending on such concepts, individual raindrops created with variance μ are assigned the best choice of features achieved (sea) [53].

3.2.6. Convergence Criteria

The water cycle stops when a feature is selected, which occurs when the standard-fit does not change at a predetermined value $\varepsilon = di_{max}$ after several iterations or achieving the largest number of generations.

The WC is illustrated along with the classifier algorithms for a WSN-DS dataset in Algorithm 2.

Algorithm 2 WC Feature Selection method

Set Mpop, Msr, dimax, Max_Iteration.
Execute Equations (11) and (12)//to determine the number of streams flowing into rivers and sea
k = Features (WSN-DS)
Mpop (1:k) = random initial population
Set the classifier algorithm (DT, KNN, SVM, DL, NB)
Calculate the fitness function for Mpop (1:k) using Equation (10)
Sort fitness values in descending order
F_sea = best fitness for initial populations
F_river = best next fitness after the F_sea
F_stream = Execute Equation (13)//to determine the number of streams flow to their corresponding rivers and sea, which is considered the best fitness after F_river
t = 0
while *(t < Max_Iterration)* ***do***
  for *i = 1: Mpop do*
    new_stream = Execute Equations (15) and (16)//to find new stream flows
    F_new_stream = Execute Equation (10) for new_stream
    if *F_new_stream < F_river* ***then***
      River = new_stream
    if *F_new_stream < F_sea* ***then***
      Sea = new_stream
  end if//line 16
  end if//line 18
  new_river = Execute Equation (17)//to find new river flows
  F_new_river = Execute Equation (10) for new_river
  if F_new_river < F_sea then
    Sea = new_river
  end if//line 24
  end for//line 13
  for *i = 1: Msr* ***do***
  if *(distance (Sea and River) < dimax) or (rand < 0.1)* ***then***
    New_stream = Execute Equation (18)
  end if//line 29
  end for//line 28
t = t + 1
end while
print the performance accuracy and number of selected features

The WC can be integrated with various machine learning classification algorithms such as DT, SVM, KNN, DL classifier, and Naive Bayes (NB). The integrated scheme that has the best performance metrics will be determined to be a WC approach in the CH node in the WSN simulation. Furthermore, the WC technique is benchmarked with various feature selection techniques such as PSO, SA, HS, and GA on the same classifier algorithms and same dataset. Based on [62], WC has been demonstrated to be a highly efficient statistical algorithm compared to many other feature selection techniques and has demonstrated superiority over them. This is in addition to the accuracy of the algorithm in terms of the

number of evaluation functions for each problem. It has also been empirically proven that WC can offer competitive solutions compared to most metaheuristics. Therefore, since the WSN traffic data is statistical, it is expected to give the best results.

3.3. Decision Making

At this stage of our proposal system, the CH monitors the number of duplicate suspicious packets; if there is a confirmation of duplication, the CH node will cut off the connection with the suspicious WSN node, add WSN node information to its blacklist, and send a broadcast message to the all neighbor CHs and the AP informing them about the suspicious WSN node. This process is illustrated in Figure 4.

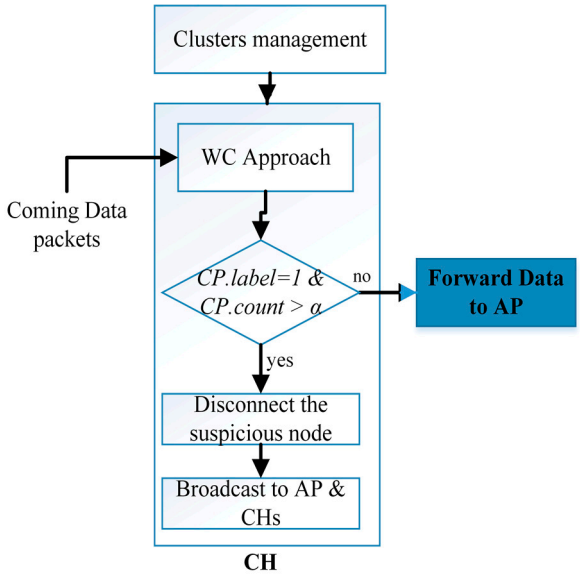

Figure 4. Decision making process in each CH.

Based on Figure 4, each data Packet that Comes (CP) will be scanned; if it is labelled as 1, CH will count it as a suspicious packet and then count the number of times it repeats over a specified period of time. Subsequently, if the *CP.count* exceeds α, the CH will execute the broadcast command and disconnect the connection, where α is the maximum number of occurrences of suspicious packets. This process gives an opportunity to reduce the false-negative rate in our proposal by confirming the attack. As for the values of time and α, they are subordinate to the policy of the organization or company, which will determine the number of repeated DoS or DDoS packets in a period of time to be considered malicious attacks or not.

4. Data Collection

In this work, we tested our proposed method using a WSN traffic dataset associated with DoS and DDoS attacks. There are different types of network traffic datasets such as CICDDoS2019 [8], BoT-IoT [68], and WSN-DS [29]. The CICDDoS2019 and BoT-IoT datasets were collected from various device types (servers, sensors, routers, and switches) that were originally classified as IoT networks and not WSNs, whereas the WSN-DS was collected from WSNs by using the LEACH protocol. The Label feature in WSN-DS data categorizes different types of DoS attacks (Blackhole, Grayhole, Flooding, and Scheduling). Moreover, it contains 18 features with approximately 325,000 records. The features disclosed are: WSN node ID, occurrence time, whether the WSN node is CH, identity of the CH for WSN nodes that are not CH, distance between WSN node and CH, advertisement messages sent from CH to WSN nodes, advertisement messages received from CHs, join request messages sent from WSN nodes to CH, join request messages received by CH from WSN nodes, TDMA advertisement messages sent to WSN nodes from CH, TDMA advertisement

messages received by CHs from WSN nodes, rank, number of data packets sent from WSN nodes to CH, number of data packets received from CH, number of data packets sent to the AP, distance between CH and AP, send code, and label. These features and their sequence are illustrated in Table 3.

Table 3. WSN-DS Features Sequence.

WSN-DS Feature	Sequence
Time	1
Is_CH	2
Who-CH	3
Distance to CH	4
ADV_S	5
ADV_R	6
Join_S	7
Join_R	8
SCH_S	9
SCH_R	10
Rank	11
Data_S	12
Data_R	13
Data_sent_to_AP	14
Dist_CH_to_AP	15
Send_code	16
Expanded_energy	17
Label	18

As with our goals to reduce node power consumption and move the defense decision to edge AP, we converted these four types of DoS attacks into one class labelled "1"; the label "0" indicates normal traffic. The amount of attack data is 88.6% less than that of normal data. Furthermore, the attack data is distributed as 33% Blackhole, 11% Flooding, 35% Grayhole, and 21% TDMA.

5. Implementation and Evaluation

In this section, we first discuss the analysis of the WC approach using the WSN-DS traffic dataset. The simulation environment and experimental performance of the WC approach are discussed, and then an analysis of the results for this approach is presented. After that, another environment was applied based on the output of the first analysis to show the impact of this approach on the lifetime of the WSN.

5.1. Complexity Analysis for the Water Cycle Detection Approach

In this work, the accuracy performance metric was utilized to analyse the WC detection approach using the WSN-SD Dataset in different classification categories. The aims were to find the highest number of True Positives (*TP*) and True Negatives (*TN*) and the lowest number of False Negatives (*FN*) and False Positives (*FP*). The number of *TN* indicates that valid traffic is recognized, while *TP* is the likelihood of irregular traffic being recognized. The number of *FN* illustrates the likelihood of attack flows being identified as normal flows, while *FP* represents the likelihood of normal flows identified as attack flows. The False Positive Rate (FPR) indicates the wrongly defined attack ratio and the inverse False Negative Rate (FNR) represents the cumulative sum of incorrect forecasts. Accuracy is the percentage of accurate model prediction for all types of predictions produced. The following Equation represents the accuracy performance metric used in this analysis:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (22)$$

5.1.1. Water Cycle Parameter Settings

We investigated the evolution of WC solutions under several parameter settings. These parameters were chosen for the WC scheme because the WSN-DS dataset differs from other datasets, and we needed to map the best results according to the fitness function mentioned earlier in (11). The di_{max} was set as 1E03. The most important parameters are M_{sr} and M_{pop}. To clarify, M_{sr} represents the total number of rivers (i.e., parameters defined by the user) plus the individual sea, while M_{pop} represents the number of raindrops (i.e., the preliminary population of features). Consequently, this section highlights the impact of changes in individual parameters. Table 4 examines three contrasting scenarios (M_{pop} = 2, 4, and 8).

Table 4. Some scenarios of parameters for the water cycle as feature selection.

Scenarios	*Mpop*	*Msr*
1	2	3
2	2	5
3	2	9
4	4	3
5	4	5
6	4	9
7	8	3
8	8	5
9	8	9

Experimental research showed that the specific relationship between M_{sr}, M_{pop} and the number of features provides the best results. In each scenario examined, the highest number of iterations was set at 100 for each run. The best result was selected based on the fitness function value. The best scenario was the seventh one, which had M_{pop} = 8 and M_{sr} = 3. Regarding other benchmark feature selection methods, we set their parameters as shown in Table 5. The rest of their parameters remained the defaults. In addition, for all methods, we defined a population size of eight and a maximum iteration limit of 100.

Table 5. Feature selection parameters settings.

Technique	Parameter	Value
GA		
	Mutation rate	0.5
PSO		
	Number of selection	3
	Constant-1	2
	Constant-2	2
HS		
	HRCR	0.7
	PAR max	0.8
	PAR min	0.2
SA		
	Initial Temp	0.2
	Temp reduction rate	0.87

5.1.2. Evaluation of the Water Cycle Approach

WC was used with the WSN-DS dataset to detect the most important features. Moreover, five classifier algorithms were evaluated in this work along with the WC technique. The standard cross-validation was used, which requires the training and validation sets to crossover in successive rounds. All this means that every data point can have a chance to be validated. A sub-category of cross-validation was *k*-fold data cross-validation. After using this cross-validation, the data were split into *k* segments of training and testing,

which were either equal or assumed to be approximately equal in size or folds. As such, k iterations were performed to practice and train results alongside validation in such a way that a different data fold was maintained within each iteration for validation, while the $(k - 1)$ iterations were utilized for basic learning. In the context of data extraction, text mining, and machine learning, (k = 10) 10-fold cross-validated appears to be the most widely used and widespread value for the data [62]. As such, we first used 90% training and 10% testing, then 80% training and 20% testing and so on until reaching 10% training and 90% testing, after which we took the average.

At this point, it must be emphasized that the results shown in the rest of this section are based on average scores for more than 20 algorithms (combining feature selection methods and classifier algorithms). In order to facilitate the computation of the average in the results of the implementation of the algorithm, we repeated the execution of the algorithms about 100 times for each execution.

To begin with, we reviewed the accuracy performance of the classifier algorithms, namely DT, SVM, KNN, DL, and NB, on the WSN-DS dataset, along with the total number of dataset features. Table 6 shows the accuracy results for these classifier algorithms.

Table 6. The accuracy results of five classifier algorithms for WSN-DS dataset.

Classifier Algorithms	Accuracy	#Features
DT	99.5922	18
KNN	98.512259	18
NB	56.87	18
SVM	98.817526	18
DL	97.6987	18

According to the values retrieved from the Table, the DT classifier displayed the highest accuracy result with a rate of 99.6%. The SVM was 98.8%, KNN was 98.5%, DL was 97.7%, and NB was 56.9%. We note that the NB showed the worst performance in terms of accuracy. One of the reasons for DT's good results is that the type of data collected from network traffic in most of its features are numerical statistical values [68]. Therefore, statistical and logical machine learning techniques provide good results with less training time.

In order to illustrate the effect of feature selection techniques on the accuracy of detection using the WSN-DS dataset, we integrated machine learning classifier algorithms along with feature selection techniques as mentioned previously to detect the most important features of the WSN-DS dataset. The WC technique was benchmarked with various feature selection techniques such as PSO [21], SA [22], HS [19], and GA [23] using the same classifier algorithms and dataset. The output of these operations is shown in Table 7. The Table also shows the accuracy performance of each integrated technique, number of WSN-DS features they used, and sequence of the features.

Based on the results of Table 7, we found that the best accuracy performance was achieved with the WC feature selection technique. It gave accuracy results of approximately 100% when using the DT and DL classifier algorithms. Moreover, if we take into account the average accuracy performance for each feature selection technique, WC continued to have the best accuracy performance, followed by POS, HS, GA, and SA in that order. With regard to the classifier algorithms used with the WC feature selection technique, we found that the DT and DL algorithms were equally the best in terms of performance accuracy and number of identified WSN-DS features. The accuracy performance metric was 100% for both and the number of WSN-DS features identified was 1. The WSN-DS feature sequence was 17 for both classifiers. The accuracy performance of the remaining classifier algorithms was distributed as SVM 99.04%, KNN 98.9%, and NB 81.98%. The number of selected WSN-DS features was also distributed between 12 and 15.

Table 7. Results of five classifier algorithms using various feature selection techniques.

Techniques	Accuracy	#Features after Selection	Feature Sequence
WC + DT	100	1	17
WC + SVM	99.0356	15	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 17, 18
WC + KNN	98.92145	16	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18
WC + NB	80.98	12	1, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 18
WC + DL	100	1	17
POS + DT	99.4278	10	1, 5, 6, 7, 9, 10, 11, 13, 16, 17
PSO + SVM	98.9156	15	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 18
POS + KNN	98.6	14	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18
POS + NB	77.1	13	1, 2, 4, 5, 6, 7, 8, 9, 10, 13, 15, 16, 18
POS + DL	97.6891	8	1, 4, 5, 6, 8, 10, 15, 16
SA + DT	99.3471	7	2, 3, 5, 7, 10, 15, 17
SA + SVM	98.267	9	1, 2, 4, 6, 7, 9, 10, 15, 16
SA + KNN	98.599	15	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 16, 18
SA + NB	58.9	14	2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 18
SA + DL	97.6913	7	7, 8, 10, 11, 13, 15, 16
HS + DT	99.3594	8	5, 7, 8, 10, 12, 14, 15, 17
HS + SVM	98.183	10	5, 6, 7, 8, 9, 10, 12, 13, 14, 16
HS + KNN	98.527	13	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 16, 18
HS + NB	78.6	13	1, 3, 4, 5, 6, 8, 10, 11, 12, 13, 14, 15, 18
HS + DL	89.9296	10	1, 4, 5, 8, 10, 11, 13, 14, 15, 18
GA + DT	99.5794	10	1, 3, 4, 5, 6, 7, 8, 10, 11, 17
GA + SVM	98.1789	13	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 15, 16, 18
GA + KNN	98.714	12	1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15
GA + NB	68.92	16	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18
GA + DL	97.6993	11	1, 4, 5, 6, 8, 11, 12, 13, 15, 16, 18

Due to the similarity of performance accuracy and number of selected features between the WC + DT and WC + DL, we resorted to using Friedman and Iman–Davenport statistical tests [69] in order to compare the two types. The Friedman and Iman–Davenport tests are designed to demonstrate whether there is a statistical difference between classes (crossover operators) [70]. Table 8 shows the average ranking of the WC with machine learning algorithms according to Friedman's test (the lower the value, the higher the rank). The last two rows in Table 8 refer to the p-value of the Friedman and Iman–Davenport statistical tests [69]. The results tabulated in Table 8 show that the WC+DT had the lowest value, so it was rated first.

Table 8. Average ranking of Friedman test for WC with machine learning techniques.

Techniques	Average Ranking
WCA + NB	9.71
WCA + KNN	6.68
WCA + SVM	6.51
WCA + DL	5.99
WC + DT	5.94
Friedman test (*p*-value)	0.00
Iman–Davenport (*p*-value)	0.00

We chose the WC + DT technique to implement in a WSN simulation. We selected it based on the highest performance in accuracy and Friedman's test ranking. To implement this technique in WSN simulations, we needed to determine the WC+DT output model from the training and testing processes. Therefore, the best value for maximum depth was set as 10 based on the accuracy performance metric. The dependent variable (Label feature) of the WSN-DS dataset had two values (0 "Normal" or 1 "Attack"), and thus the portion of the WC + DT output model from the WSN-DS dataset represented the relationship between the independent variables and dependent variables as illustrated in Figure 5.

```
| | | |--- Expanded_Energy > 2.26
| | | | |--- Expanded_Energy <= 3.02
| | | | | |--- Expanded_Energy <= 2.31
| | | | | | |--- Expanded_Energy <= 2.27
| | | | | | | |--- class: 0
| | | | | | |--- Expanded_Energy > 2.27
| | | | | | | |--- Expanded_Energy <= 2.31
| | | | | | | | |--- Expanded_Energy <= 2.27
| | | | | | | | | |--- class: 1
| | | | | | | | |--- Expanded_Energy > 2.27
| | | | | | | | | |--- Expanded_Energy <= 2.30
| | | | | | | | | | |--- class: 0
| | | | | | | | | |--- Expanded_Energy > 2.30
| | | | | | | | | | |--- class: 0
| | | | | | | |--- Expanded_Energy > 2.31
| | | | | | | | |--- class: 1
| | | | | |--- Expanded_Energy > 2.31
| | | | | | |--- Expanded_Energy <= 2.43
```

Figure 5. A sample of the WC+DT output model for the WSN-DS dataset.

5.2. Complexity Analysis for the Lifetime of WSNs in CH_Rotations and WC + DT Approaches

5.2.1. Simulation Environment

Contiki operating system with Cooja simulator were used to simulate WSN architecture [71]. The modified LEACH protocol was used as a clustering management subsection to manage and control the WSN node's hardware and software. The simulation was run on a machine with a 1.8 GHz Intel Core i5 processor, 6 MB cache, and 8 GB RAM. The default parameters used in the architecture of the wireless network are plotted in Table 9, and parameter values in the table are taken from the values in [72].

Table 9. Simulation parameters used.

Parameter	Value
WSN node size	60 m × 120 m
ER location	X = 30, Y = 90
Number of CHs	Changeable
Number of WSN nodes	100
Simulation time	500
Message size	6400 bits
Control message size	200 bits
Initial energy (Joule)	1
Two-ray ground propagation models	0.0013 PJ/bit/m4
Free space model	10 PJ/bit/m2
Power consumed by transmitter	50 nJ/bit
Transition power	20 nJ/bit
Power consumed by receiver	50 nJ/bit
Distance threshold	87 m

In the simulation, the WSN nodes were initially spread across dimensions of 500 × 500 terrain associated with nine CHs at initial values distributed in different subregions within the AP coverage region. The initial energy of the WSN node and the number of used WSN nodes were chosen based on the proposed simulated need, which we will discuss later in detail. Moreover, each ordinary node sent 64 packets per second to its CH node and each packet size was 1000 bits. If the node was CH, the received packets were forwarded to AP.

5.2.2. Experimental Metrics and Results

In this step, the CH_Rotations algorithm was first analysed to show its effect on WSN lifetime. Next, we analysed the impact of the WC + DT technique incorporating the CH_Rotations algorithm on WSN lifetime. The lifetime of the network was calculated when the power of some WSN nodes reached 0.

With regard to the evaluation of the "CH_Rotations" proposal scheme, we set the initial energy of the WSN node to 1 joule and the simulation time to 200 s to allow some of the WSN nodes' energy to reach 0. The number of nodes was increased by 100 each time and the number of CHs was set to 9 CHs for both schemes (CH_Rotations and LEACH). Analysis of the effect of WSN node number to network lifetime for both schemes is depicted in Figure 6.

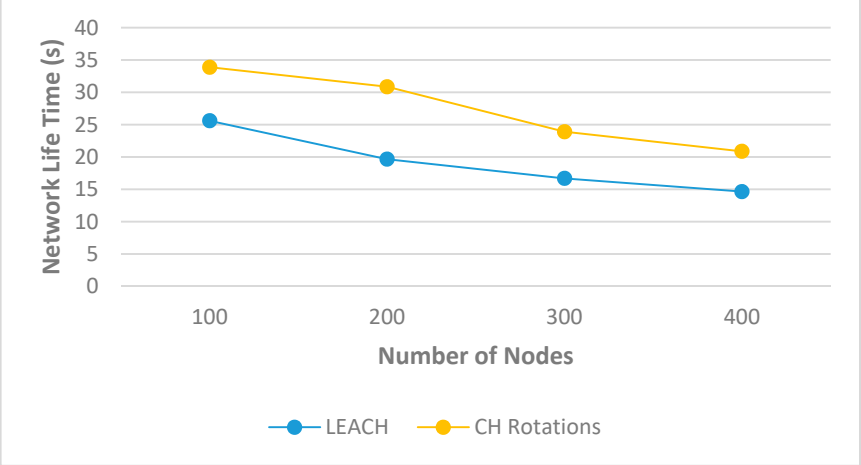

Figure 6. Analysis of the effect of WSN nodes number on the network lifetime.

As illustrated in Figure 6, an increase in the number of WSN nodes resulted in reduced network lifetime when using either technique. The reason behind this decrease is the effect of over-connecting ordinary WSN nodes to the CHs. However, the CH_Rotations algorithm showed an improvement in network lifetime compared to the LEACH technique, because the process of selecting CH was done mathematically based on several factors and not randomly as in LEACH. In addition, based on the effect of distance and received and transmitted signal strength between the WSN nodes, an increase in distance increases and decrease in signal strength correlates to an increase in energy consumption and a decrease in network transmission rate. Thus, selecting CH positions close to neighbouring WSN nodes and to the AP provides good communication and conserves the network lifetime. Finally, the result showed that CH_Rotations improved network lifetime by 24%, 36%, 30%, and 29% compared to the LEACH technique when using 100, 200, 300, and 400 WSN nodes, respectively.

For DoS detection evaluation using the WC + DT technique, we simulated it with the CH_Rotations algorithm and ran it to see the effect of monitoring and packet inspection in each CH on the network lifetime. The WC + DT output model was distributed to all WSN nodes, and when a WSN node became a CH, it started monitoring the consumption energy, then found expanded energy and calculated y (Label feature) for each packet. For each positive y (Label = 1), the CH created a counter table for the WSN node that sent suspicious packets, and calculated from one to α; if the counter reached this value in a period of time (t), the CH blocked this node and sent a broadcast message to the AP for this case. Moreover, these counter tables were forwarded between WSN nodes so that all of them were aware of these numbers. The idea of a counter table is important and is meant to reduce the FNR in WSNs.

In the WC+DT technique simulation, the WSN nodes were initially spread randomly as seen in Figure 7. The initial energy of the WSN node was set to 1.5 Joules and the

simulation time to 500 s to allow some of the WSN nodes' energy in the first scenario to reach zero.

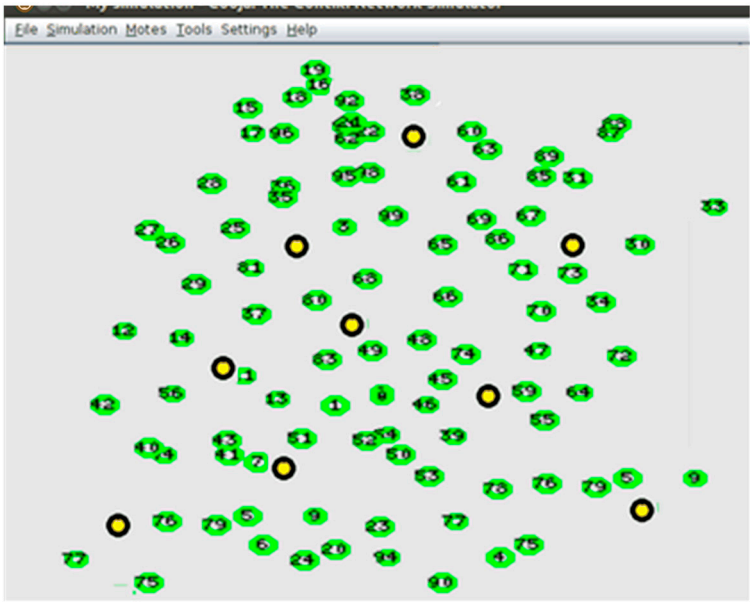

Figure 7. Simulation WSN nodes.

The monitoring of nodes' packet activity occurred in a constant time interval, and the statistical calculation for each DoS attack during these time intervals (t) was the same data features calculation of [29]. Regarding the WC+DT detection model, we supposed that the energy consumption per each packet inspection would be 0.001 J, and depending on this energy consumption value, the analysis effect of the WC+DT DoS detection on the WSN lifetime is illustrated in Figure 8. As illustrated in Figure 8, the increments of initial power in WSN nodes increased the lifetime of WSNs in both scenarios. This result is due to the positive relationship between WSN node initial power and time intervals. Moreover, from the same Figure 8, we can observe that the variation in the network lifetime between two scenarios increased with increases in the initial power of the WSN nodes.

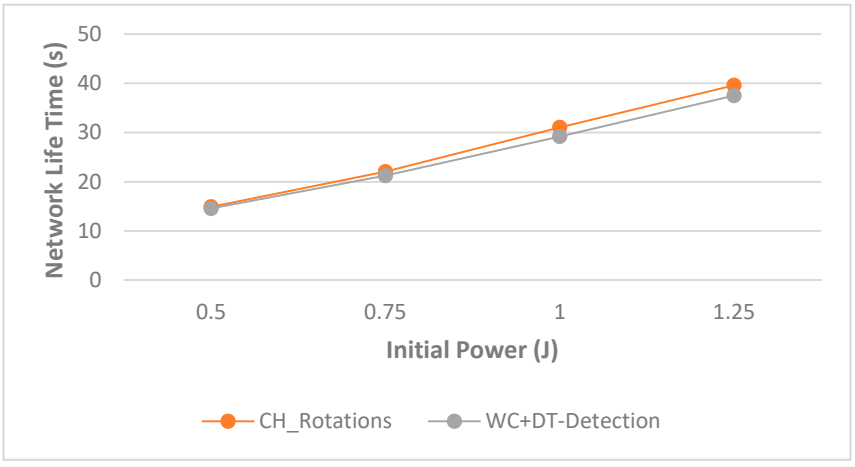

Figure 8. Analysis of the effect of WC+DT technique on WSN lifetime.

This variation increased from 2% to 6% when the WSN node initial power increased from 0.5 to 1.25 J. The reason for the increase in this variance was due to the increase in the rate of packets received by the CH nodes, which in turn led to an increase in the rate of inspection and verification messages. This in turn led to an increase in the rate of power

consumption within the CH nodes, and thus the result was a decrease in the network lifetime. The results show that the WC+DT detection algorithm decreased the network lifetime by 2%, 4%, 6%, and 6% compared to the WC+DT-free scenario for WSN node initial power of 0.5, 0.75, 1, and 1.25 J, respectively.

6. Conclusions and Future Work

Network traffic is becoming more complex due to the increase in the amount of data transferred between WSN nodes resulting from increased usage. It is important to reduce power consumption and improve data protection in these networks, especially in order to prevent DoS attacks. In this paper, we modified the LEACH clustering protocol to improve its performance by adding various factors such as WSN node residual power, distance between WSN nodes, and the distance between the candidate CHs and the AP. Moreover, we analysed the performance of various feature selection techniques along with different machine learning algorithms to improve DoS detection in the WSN-DS dataset. The feature selection techniques used were WC, SPO, HS, and GA, and with each feature selection technique different machine learning algorithms such as DT, DL, KNN, NB, and SVM were used. Performance accuracy metrics were used to evaluate each algorithm. The best technique for feature selection was WC as its average performance accuracy was 2%, 5%, 3%, and 3% higher than that of PSO, SA, HS, and GA, respectively. The best machine learning algorithm results when used with WC were displayed by the DT and DL algorithms, which had the highest accuracy of 100% and the lowest number of features (Expanded Energy). The rest of machine learning algorithms achieved accuracy performances of SVM 99%, KNN 99%, and NB 81% with different numbers of WSN_DS features distributed between 12 and 15. The Friedman and Iman–Davenport statistical tests were used to select which of the two highest-performing machine learning algorithms (DT or DL) was most appropriate. The WC+DT had the lowest score of 5.449, hence WC + DT was selected as the best DoS detection technique.

Furthermore, Cooja simulator software was also used to obtain WSN lifetime. The simulation environment was managed by either CH_Rotations, a modified LEACH protocol, or the LEACH standard protocol. CH_Rotations improved the WSN lifetime by 30% compared to the standard LEACH routing protocol. The WC + DT technique consumed 5% of the total WSN lifetime compared to the WC + DT-free scenario. In future, we plan to collect a new WSN dataset from the 6LoWPAN protocol and add new features such as packet size per stream, dropped packets per stream, flow change ratio, and packet change ratio.

Author Contributions: All authors contributed to this manuscript. Conceptualization, R.A., R.W., T.A.-A. and W.A.-A.; investigation, R.A., Q.B., R.W. and T.A.-A.; data duration, R.A., R.W. and W.A.-A.; writing—original draft, R.A., T.A.-A. and Q.B.; visualization, R.A.; supervision, R.A.; writing—review & editing, R.A., R.W. and Q.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al-Emran, M.; Malik, S.I.; Al-Kabi, M.N. A Survey of Internet of Things (IoT) in Education: Opportunities and Challenges. In *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 197–209, ISBN 9783030245139.
2. Zhang, G.; Kou, L.; Zhang, L.; Liu, C.; Da, Q.; Sun, J. A New Digital Watermarking Method for Data Integrity Protection in the Perception Layer of IoT. *Secur. Commun. Netw.* **2017**, *2017*, 1–12. [CrossRef]
3. Yi, L.; Tong, X.; Wang, Z.; Zhang, M.; Zhu, H.; Liu, J. A Novel Block Encryption Algorithm Based on Chaotic S-Box for Wireless Sensor Network. *IEEE Access* **2019**, *7*, 53079–53090. [CrossRef]

4. Butun, I.; Morgera, S.D.; Sankar, R. A Survey of Intrusion Detection Systems in Wireless Sensor Networks. *IEEE Commun. Surv. Tutorials* **2014**, *16*, 266–282. [CrossRef]
5. Glissa, G.; Meddeb, A. 6LoWPAN Multi-Layered Security Protocol Based on IEEE 802.15.4 Security Features. In Proceedings of the 2017 13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017, Valencia, Spain, 26–30 June 2017; pp. 264–269.
6. Lee, C.-C. Security and Privacy in Wireless Sensor Networks: Advances and Challenges. *Sensors* **2020**, *20*, 744. [CrossRef]
7. Abidoye, A.P.; Obagbuwa, I.C. DDoS attacks in WSNs: Detection and Countermeasures. *IET Wirel. Sens. Syst.* **2018**, *8*, 52–59. [CrossRef]
8. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–8.
9. Vishwakarma, R.; Jain, A.K. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommun. Syst.* **2020**, *73*, 3–25. [CrossRef]
10. Khan, K.; Mehmood, A.; Khan, S.; Khan, M.A.; Iqbal, Z.; Mashwani, W.K. A survey on intrusion detection and prevention in wireless ad-hoc networks. *J. Syst. Archit.* **2020**, *105*, 101701. [CrossRef]
11. Kaur, T.; Saluja, K.K.; Sharma, A.K. DDOS attack in WSN: A survey. In Proceedings of the 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 23–25 December 2016.
12. Cauteruccio, F.; Cinelli, L.; Corradini, E.; Terracina, G.; Ursino, D.; Virgili, L.; Savaglio, C.; Liotta, A.; Fortino, G. A framework for anomaly detection and classification in Multiple IoT scenarios. *Futur. Gener. Comput. Syst.* **2021**, *114*, 322–335. [CrossRef]
13. Premkumar, M.; Sundararajan, T.V.P. DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks. *Microprocess. Microsyst.* **2020**, *79*, 103278. [CrossRef]
14. Wu, D.; Jiang, Z.; Xie, X.; Wei, X.; Yu, W.; Li, R. LSTM Learning with Bayesian and Gaussian Processing for Anomaly Detection in Industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *16*, 5244–5253. [CrossRef]
15. Han, L.; Zhou, M.; Jia, W.; Dalil, Z.; Xu, X. Intrusion detection model of wireless sensor networks based on game theory and an autoregressive model. *Inf. Sci.* **2019**, *476*, 491–504. [CrossRef]
16. Praveen Kumar, D.; Amgoth, T.; Annavarapu, C.S.R. Machine learning algorithms for wireless sensor networks: A survey. *Inf. Fusion* **2019**, *49*, 1–25. [CrossRef]
17. Cheng, J.; Zhou, J.; Liu, Q.; Tang, X.; Guo, Y. A DDoS detection method for socially aware networking based on forecasting fusion feature sequence. *Comput. J.* **2018**, *61*, 959–970. [CrossRef]
18. Otoum, S.; Kantarci, B.; Mouftah, H.T. On the Feasibility of Deep Learning in Sensor Network Intrusion Detection. *IEEE Netw. Lett.* **2019**, *1*, 68–71. [CrossRef]
19. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [CrossRef]
20. Ahmad, B.; Jian, W.; Ali, Z.A.; Tanvir, S.; Khan, M.S.A. Hybrid Anomaly Detection by Using Clustering for Wireless Sensor Network. *Wirel. Pers. Commun.* **2019**, *106*, 1841–1853. [CrossRef]
21. Lu, X.; Han, D.; Duan, L.; Tian, Q. Intrusion detection of wireless sensor networks based on IPSO algorithm and BP neural network. *Int. J. Comput. Sci. Eng.* **2020**, *22*, 221–232. [CrossRef]
22. Lin, S.W.; Ying, K.C.; Lee, C.Y.; Lee, Z.J. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. *Appl. Soft Comput. J.* **2012**, *12*, 3285–3290. [CrossRef]
23. Wang, M.; Lu, Y.; Qin, J. A dynamic MLP-based DDoS attack detection method using feature selection and feedback. *Comput. Secur.* **2020**, *88*, 101645. [CrossRef]
24. Bikmukhamedov, R.F.; Nadeev, A.F. Lightweight Machine Learning Classifiers of IoT Traffic Flows. In Proceedings of the 2019 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), Yaroslavl, Russia, 1–3 July 2019; pp. 1–5.
25. Depari, A.; Ferrari, P.; Flammini, A.; Rinaldi, S.; Sisinni, E. Lightweight Machine Learning-Based Approach for Supervision of Fitness Workout. In Proceedings of the 2019 IEEE Sensors Applications Symposium (SAS), Sophia Antipolis, France, 11–13 March 2019.
26. Bouaziz, M.; Rachedi, A. A survey on mobility management protocols in Wireless Sensor Networks based on 6LoWPAN technology. *Comput. Commun.* **2016**, *74*, 3–15. [CrossRef]
27. Chakeres, I.D.; Belding-Royer, E.M. AODV Routing Protocol Implementation Design. In Proceedings of the 24th International Conference on Distributed Computing Systems Workshops, Tokyo, Japan, 23–24 March 2004; pp. 698–703.
28. Tyagi, S.; Kumar, N. A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 623–645. [CrossRef]
29. Almomani, I.; Al-Kasasbeh, B.; Al-Akhras, M. WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks. *J. Sens.* **2016**, *2016*, 4731953. [CrossRef]
30. Gupta, V.; Doja, M.N. H-LEACH: Modified and efficient LEACH protocol for hybrid clustering scenario in wireless sensor networks. *Adv. Intell. Syst. Comput.* **2018**, *638*, 399–408.
31. Cai, X.; Geng, S.; Wu, D.; Wang, L.; Wu, Q. A unified heuristic bat algorithm to optimize the LEACH protocol. *Concurr. Comput.* **2020**, *32*, 1–9. [CrossRef]

32. Al-Baz, A.; El-Sayed, A. A new algorithm for cluster head selection in LEACH protocol for wireless sensor networks. *Int. J. Commun. Syst.* **2018**, *31*, 1–13. [CrossRef]
33. Abu Salem, A.O.; Shudifat, N. Enhanced LEACH protocol for increasing a lifetime of WSNs. *Pers. Ubiquitous Comput.* **2019**, *23*, 901–907. [CrossRef]
34. Cui, Z.; Cao, Y.; Cai, X.; Cai, J.; Chen, J. Optimal LEACH protocol with modified bat algorithm for big data sensing systems in Internet of Things. *J. Parallel Distrib. Comput.* **2019**, *132*, 217–229. [CrossRef]
35. Hasan, M.; Islam, M.M.; Zarif, M.I.I.; Hashem, M.M.A. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet Things* **2019**, *7*, 100059. [CrossRef]
36. Islam, M.N.U.; Fahmin, A.; Hossain, M.S.; Atiquzzaman, M. Denial-of-Service Attacks on Wireless Sensor Network and Defense Techniques. *Wirel. Pers. Commun.* **2020**, *116*, 1993–2021. [CrossRef]
37. Behera, T.M.; Samal, U.C.; Mohapatra, S.K. Energy-efficient modified LEACH protocol for IoT application. *IET Wirel. Sens. Syst.* **2018**, *8*, 223–228. [CrossRef]
38. Singh, S.K.; Kumar, P.; Singh, J.P. A Survey on Successors of LEACH Protocol. *IEEE Access* **2017**, *5*, 4298–4328. [CrossRef]
39. Kumar, V.; Tiwari, S. Routing in IPv6 over low-power wireless personal area networks (6LoWPAN): A survey. *J. Comput. Networks Commun.* **2012**, *2012*, 316839. [CrossRef]
40. Liang, H.; Yang, S.; Li, L.; Gao, J. Research on routing optimization of WSNs based on improved LEACH protocol. *Eurasip J. Wirel. Commun. Netw.* **2019**, *2019*, 1–12. [CrossRef]
41. Monser, M.E.; Chikha, H.B.; Attia, R. Prolonging the lifetime of large-scale wireless sensor networks using distributed cooperative transmissions. *IET Wirel. Sens. Syst.* **2018**, *8*, 229–236. [CrossRef]
42. Kumar, P.M.; Gandhi, U.D. Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application. *J. Supercomput.* **2020**, *76*, 3963–3983. [CrossRef]
43. Zhang, X.; Heys, H.M.; Li, C. Energy efficiency of encryption schemes applied to wireless sensor networks. *Secur. Commun. Networks* **2012**, *5*, 789–808. [CrossRef]
44. Yang, Y.; Wu, L.; Yin, G.; Li, L.; Zhao, H. A Survey on Security and Privacy Issues in Internet-of-Things. *IEEE Internet Things J.* **2017**, *4*, 1250–1258. [CrossRef]
45. Can, O.; Sahingoz, O.K. A Survey of Intrusion Detection Systems in Wireless Sensor Networks. In Proceedings of the 2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), Istanbul, Turkey, 27–29 May 2015.
46. Vangipuram, R.; Gunupudi, R.K.; Puligadda, V.K.; Vinjamuri, J. A machine learning approach for imputation and anomaly detection in IoT environment. *Expert Syst.* **2020**, *37*, 1–16. [CrossRef]
47. Yamauchi, M.; Ohsita, Y.; Murata, M.; Ueda, K.; Kato, Y. Anomaly Detection in Smart Home Operation from User Behaviors and Home Conditions. *IEEE Trans. Consum. Electron.* **2020**, *66*, 183–192. [CrossRef]
48. Singh, K.J.; De, T. MLP-GA based algorithm to detect application layer DDoS attack. *J. Inf. Secur. Appl.* **2017**, *36*, 145–153. [CrossRef]
49. Borkar, G.M.; Patil, L.H.; Dalgade, D.; Hutke, A. A novel clustering approach and adaptive SVM classifier for intrusion detection in WSN: A data mining concept. *Sustain. Comput. Inform. Syst.* **2019**, *23*, 120–135. [CrossRef]
50. Alweshah, M.; Alkhalaileh, S.; Albashish, D.; Mafarja, M.; Bsoul, Q.; Dorgham, O. A hybrid mine blast algorithm for feature selection problems. *Soft Comput.* **2021**, *25*, 517–534. [CrossRef]
51. Mafarja, M.M.; Mirjalili, S. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing* **2017**, *260*, 302–312. [CrossRef]
52. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A. Variational data generative model for intrusion detection. *Knowl. Inf. Syst.* **2019**, *60*, 569–590. [CrossRef]
53. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **2012**, *110–111*, 151–166. [CrossRef]
54. Tabakhi, S.; Moradi, P.; Akhlaghian, F. An unsupervised feature selection algorithm based on ant colony optimization. *Eng. Appl. Artif. Intell.* **2014**, *32*, 112–123. [CrossRef]
55. Bharti, K.K.; Singh, P.K. A three-stage unsupervised dimension reduction method for text clustering. *J. Comput. Sci.* **2014**, *5*, 156–169. [CrossRef]
56. Abualigah, L.M.; Khader, A.T.; Al-Betar, M.A. Unsupervised Feature Selection Technique Based on Genetic Algorithm for Improving the Text Clustering. In Proceedings of the 2016 7th International Conference on Computer Science and Information Technology (CSIT), Amman, Jordan, 13–14 July 2016.
57. Abualigah, L.M.; Khader, A.T.; Hanandeh, E.S. A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *J. Comput. Sci.* **2018**, *25*, 456–466. [CrossRef]
58. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
59. Ioannou, C.; Vassiliou, V. An Intrusion Detection System for Constrained WSN and IoT Nodes Based on Binary Logistic Regression. In *21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*; ACM: New York, NY, USA, 2018; pp. 259–263.
60. Darabkh, K.A.; El-Yabroudi, M.Z.; El-Mousa, A.H. BPA-CRP: A balanced power-aware clustering and routing protocol for wireless sensor networks. *Ad Hoc Netw.* **2019**, *82*, 155–171. [CrossRef]

61. Darabkh, K.A.; Al-Maaitah, N.J.; Jafar, I.F.; Khalifeh, A.F. EA-CRP: A Novel Energy-aware Clustering and Routing Protocol in Wireless Sensor Networks. *Comput. Electr. Eng.* **2018**, *72*, 702–718. [CrossRef]
62. Al-Rawashdeh, G.; Mamat, R.; Hafhizah Binti Abd Rahim, N. Hybrid Water Cycle Optimization Algorithm with Simulated Annealing for Spam E-mail Detection. *IEEE Access* **2019**, *7*, 143721–143734. [CrossRef]
63. Ali, N.; Neagu, D.; Trundle, P. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. *SN Appl. Sci.* **2019**, *1*, 1559. [CrossRef]
64. Coppolino, L.; DAntonio, S.; Garofalo, A.; Romano, L. Applying Data Mining Techniques to Intrusion Detection in Wireless Sensor Networks. In Proceedings of the 2013 Eighth International Conference on P2P Parallel, Grid, Cloud and Internet Computing, Compiegne, France, 28–30 October 2013; pp. 247–254.
65. Qu, H.; Lei, L.; Tang, X.; Wang, P. A Lightweight Intrusion Detection Method Based on Fuzzy Clustering Algorithm for Wireless Sensor Networks. *Adv. Fuzzy Syst.* **2018**, *2018*, 4071851. [CrossRef]
66. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary ant lion approaches for feature selection. *Neurocomputing* **2016**, *213*, 54–65. [CrossRef]
67. Chen, C.; Wang, P.; Dong, H.; Wang, X. Hierarchical Learning Water Cycle Algorithm. *Appl. Soft Comput.* **2020**, *86*, 105935. [CrossRef]
68. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset. *Future Gener. Comput. Syst.* **2018**, *100*, 779–796. [CrossRef]
69. Luengo, J.; García, S.; Herrera, F. A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests. *Expert Syst. Appl.* **2009**, *36*, 7798–7808. [CrossRef]
70. Picek, S.; Golub, M.; Jakobovic, D. Evaluation of crossover operator performance in genetic algorithms with binary representation. *Lect. Notes Comput. Sci.* **2011**, *6840 LNBI*, 223–230.
71. Zikria, Y.B.; Afzal, M.K.; Ishmanov, F.; Kim, S.W.; Yu, H. A survey on routing protocols supported by the Contiki Internet of things operating system. *Futur. Gener. Comput. Syst.* **2018**, *82*, 200–219. [CrossRef]
72. Khashan, O.A.; Ahmad, R.; Khafajah, N.M. An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks. *Ad Hoc Netw.* **2021**, *115*, 102448. [CrossRef]