International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 57, No. 6, December, 2024, pp. 1649-1657
Journal homepage: http://iieta.org/journals/jesa

Enhanced Extreme Learning Machine for Energy Efficient Vampire Attack Detection in Wireless Sensor Networks

Ibrahim Saud Khaleel[1*], Bilal Mishaal Mohammed[2], Ahmed Adnan[3]

[1] Anbar Vocational Education Department, General Directorate of Education Anbar, Ramadi 31001, Iraq
[2] College of Basic Education, University of Anbar, Ramadi 31001, Iraq
[3] College of Energy and Environmental Science, Al-Karkh University of Science, Baghdad 10081, Iraq

Corresponding Author Email: ibrahem.abomusab@uoanbar.edu.iq

Copyright: ©2024 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/jesa.570612

Received: 14 October 2024
Revised: 28 November 2024
Accepted: 6 December 2024
Available online: 31 December 2024

Keywords:
vampire attack, sleep scheduling, wireless sensor networks, clustering, machine learning, optimization method

ABSTRACT

A sensor node fulfils a specific function inside a wireless sensor network (WSN). WSNs are characterised by a lower tolerance for errors or failures, but they are also more essential to the success of businesses and the well-being of individuals because of the inherent risks involved. Consequently, the battery life of a node would diminish, rendering it non-operational, which is considered the most severe kind of denial of service assault. Vampire assaults, a kind of denial of service attack, may cause damage to a network, resulting in increased difficulty in detection and unnecessary energy consumption. This research proposes a new method for detecting and preventing vampire attacks by predicting energy consumption in the data path. The method uses the Extreme Learning Machine with Sleep Scheduling Algorithm (ELM_SSA), which has a fast learning speed and is well-suited for resource-limited environments such as WSNs. The sleep scheduling algorithm determines when the nodes should be active and when they can enter sleep mode to conserve energy. Nodes may be scheduled to wake up periodically to perform energy consumption measurements and collect data for anomaly detection.

1. INTRODUCTION

Wireless sensor networks (WSNs) are comprised of sensor nodes that are interconnected wirelessly, forming a subset of ad hoc networks. Infrastructure is not required for self-configuring networks [1, 2]. Sensor nodes can detect, relay, and share data with external networks [3]. WSNs were originally developed for military purposes, namely for detecting enemy movements. However, they have since been used in several civil applications as well [4]. Security risks targeting WSNs have been extensively researched because hostile acts intended to prevent military-related WSN applications from operating normally naturally pose a danger to them [5]. The majority of earlier research has concluded that many attacks have the same objective of immediately, sufficiently, or quickly preventing the network from operating [6]. Even if the source of these assaults may not be found right away, the disturbances they produce may highlight other attempts that are already in progress [7]. The network operator will thus be informed and take action to protect against the impact of these assaults [8, 9]. This kind of attack strategy may be less successful from the attacker's perspective since it targets military-related networks, whose users have most likely taken security precautions before deployment [10]. It is preferable to carry out subtle assaults that go undetected for a while. Certain attacks aim to weaken the network gradually over a considerable amount of time rather than causing an instant disruption to its availability [11]. Vampire assaults [12] are one example of this kind of attack, which aims to covertly drain the network's energy resources (typically nodes' batteries). Vampire attacks pose a significant threat to WSN systems operating in challenging situations, such as environmental monitoring or enemy identification. These assaults are particularly detrimental since the nodes of these applications are difficult to access, making battery replacement a challenging or even impossible task [13]. Vampire attacks may take advantage of the absence of authentication in control messages of conventional routing protocols specifically built for WSNs. As indicated earlier in the preceding paragraph, it might be challenging (and often impossible) to recharge the energy storage of sensor nodes in some applications of WSNs. Vampire assaults may expedite the malfunctioning of certain sensors, leading to interruptions in the network. Therefore, network operators need to discover anomalous indicators of vampire attacks and ascertain probable perpetrators.

A vampire assault may do the following:

- The route loop assault, also known as the carousel attack, involves the deliberate creation of routing loops where data packets are made to constantly travel over the same loop [14].
- Stretch assault: In this kind of assault, the attackers attempt to increase the length of conventional routes as much as possible and compel data to pass through several unnecessary nodes.

Consequently, the average distance covered by data may see a substantial increase, and the number of nodes unintentionally involved in data transmission also increases.

Cluster creation is a prominent strategy in WSN that aims to decrease energy consumption [15]. CHs lead sensor node clusters. The best cluster head is chosen using bio-inspired algorithms such ensemble Particle Swarm Optimisation (PSO) and Gravitational Search Algorithm (GSA) [16], Genetic Algorithm (GA) [17], and Multi-Objective Evolutionary Algorithm (MOEA) [18]. The hybrid PSO/GSA calculates the cost function by incorporating CH proximity and energy. The cost function distributes the next hop for each CH to evenly share the workload across cluster heads. The HB (hybrid mode) method involves the selection of a group of cluster heads from the present nodes in the network. These cluster heads then establish the clusters depending on their positions. GA describes a system for dynamically optimizing wireless sensor node clusters using self-organizing network clustering. GA and multi-objective Particle Swarm Optimization (PSO) help the MOEA choose an efficient CH. Fuzzy methods select the best WSN cluster head from sensor nodes [19]. During the election process, sensor nodes are evaluated for their eligibility to become CH based on their eligibility index. Hierarchical routeing techniques such as LEACH and PEGASIS encounter difficulties in the selection of cluster heads and routeing of sink nodes. GEAR, GAF, and SPAN, location-based routing protocols, also have high overhead, energy consumption, and scalability issues in WSNs. Furthermore, these protocols do not prioritize the most efficient CH for packet routing. Therefore, there is an increase in energy consumption and a decline in the stability of the WSN. WSNs operate under stringent constraints, including limited processing power, memory, and bandwidth. Traditional security measures are often computationally intensive and energy-hungry. There is a need for lightweight, energy-aware strategies specifically tailored to the unique architecture of WSNs. Mitigating vampire attacks ensures that nodes maintain their operational lifespan, maximizing the utility and efficiency of WSN deployments, particularly in environments where replacing or recharging batteries is difficult (e.g., remote monitoring or disaster zones). Based on the aforementioned investigations, it is evident that WSN continues to face the challenges outlined below.

- The vampire attack presents a significant security threat whereby a compromised node, dubbed the "vampire", maliciously drains energy from neighboring nodes, hastening their depletion and potentially causing network failure. This attack jeopardizes the integrity and longevity of the network, requiring robust detection and mitigation strategies to safeguard against energy depletion and ensure the reliable operation of WSNs.
- Energy consumption is a significant issue in WSN. The majority of the works focus on clustering as a means to decrease energy use. However, it does not include a method for the selection of the most efficient CH. This diminishes the efficiency of aggregating data.

The following goals are constructed using our Energy-Scheduling utilizing Extreme Learning Machine Sleep Scheduling Algorithm (ELM_SSA) to address these flaws in the existing WSN communication protocols:

- This paper proposes a novel approach that combines K-Medoids clustering with an improved version of the Sailfish Optimization (SO) algorithm. The SO algorithm, inspired by the hunting behavior of sailfish, is known for its efficiency in optimization tasks. The SO algorithm is enhanced by introducing adaptive mechanisms to improve its exploration and exploitation capabilities. Fitness functions depend on residual energy, hop count, destination node distance, and bandwidth.
- By adopting the Extreme Learning Machine (ELM), this study formulates the detection of vampire attacks as a binary classification problem, where the objective is to distinguish between normal and attack traffic based on network features.
- To minimize the impact of vampire attacks, the hierarchical sleep scheduling mechanism should be dynamic and adaptive. Nodes may adjust their sleep patterns based on perceived threats. Nodes suspected of being under attack could reduce their active periods, increase sleep duration, or even temporarily disconnect from the network to prevent further energy drain.

This article is organized as follows: Section 2 provides an overview of the most advanced research related to the effective management of energy WSN. Section 3 examines the proposed work, including detailed discussions of the algorithms. Section 4 presents the experimental findings and a comparative analysis. Finally, Section 5 provides a conclusion and future works.

2. RELATED WORKS

Although there is growing interest in machine learning (ML) in the WSN field, there is a lack of a thorough overview specifically addressing ML for energy-efficient methods in WSNs. This article provides a comprehensive review of the latest advancements in research on machine learning-based routing algorithms in WSNs, intending to promote their practical implementation. This study seeks to bridge the divide between ML and routing techniques in WSNs by providing a comprehensive and cutting-edge overview. Its purpose is to engage practitioners who are interested in advancing the growth of this discipline.

A Modified-PLGPa protocol is introduced in the study [20] to identify and remove attacks in order to enhance the longevity of the network. This technique mitigates the impact of vampire attacks by minimizing the energy depletion resulting from the transmission of unnecessary packets in the network routing. The study [21] addresses anomalous energy draining via memory-efficient data structures, such as Bloom filters, count-min (CM) drawings, and cellular automata (CA). A trust architecture based on CAs selects the cluster head node. The fair selection procedure is ensured by the CM sketch algorithm, which regulates the frequency of cluster head selection. On the other hand, Bloom filters are utilized to keep track of malicious nodes that are prohibited from engaging in communication or cluster head selection. The VAD-FCOPRAS technique [22] detects and mitigates vampire attacks to preserve sensor node energy and increase network longevity. The VAD-FCOPRAS technique outperforms baseline vampire detection methods with varying sensor node density. The PFSVT-MCDM [23] uses Pythagorean Fuzzy Sets to mitigate resource depletion attacks and improve network QoS. When compared to resource depletion assault thwarting tactics used for assessment, PFSVT-MCDM shows a 21.29% throughput improvement, 22.38% packet delivery fraction, 18.92% energy consumption reduction, and 21.84%

end-to-end latency reduction. The study [24] develops DERNNets to categorise vampire nodes in networks. Grey Wolf Optimisation (GWO) finds the optimal aggregation locations to optimise data flow and improve network node battery life. WKSH cluster ensemble [25] prevents attacks using weighted averaging-based K-means Spectral and Hierarchical Clustering. WKSH finds anomalies using graphs, weighted Euclidean distance, and average consensus. Researchers [26] found that fuzzy rules and sets minimise network vampire attacks. These approaches accurately assess sensor node unpredictability. This project will use a probabilistic fuzzy chain set, authentication-based routeing protocol, and hybrid clustering method to optimise network data. A fuzzy-based chain rule set and probability calculations battle the growing diversity of vampire assaults. The authentication routeing technology has improved network routeing security. The proposed technique, PFCS-ARP_HC, increased network energy efficiency. NS2 simulations showed a throughput of 98%, packet delivery ratio of 89%, energy usage of 67%, end-to-end latency of 46%, control overhead of 53%, and attack detection ratio of 87.9% for the suggested model. The paper [27] proposes Bayesian optimization-based Deep Learning (DL). Nevertheless, the developed optimized deep learning approach, albeit demonstrating encouraging outcomes in improving security, has obstacles such as reliance on data, computational intricacy, and the risk of overfitting. The CTRF uses the fire hawk optimizer to construct a cluster-based trustworthy routing approach [28]. This technique improves WSN network security by considering node energy restrictions.

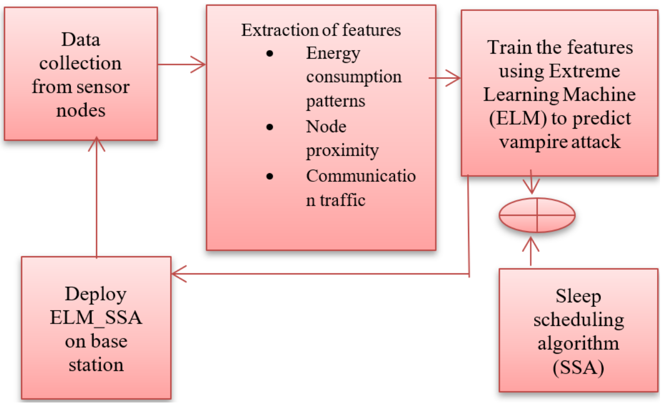
The system utilises the interaction between sensor nodes to create a trust mechanism that assigns weights to different nodes, known as a weighted trust mechanism (WTM). This trust method calculates weighted reception rate, redundancy rate, and energy state using exponential factors. This lets sensor nodes' trust ratings exponentially change depending on their antagonistic or friendly behavior. In the study [29], a virtual force-directed improved sand cat swarm optimisation algorithm (VF-ISCSO) is introduced. This approach uses artificial intelligence to enhance the performance and achieve two main objectives: increasing the coverage of sensor nodes and reducing coverage gaps. An innovative route clustering optimization method [30] considers grid size, orientation, velocity node density, and communication range to achieve its goal: intelligent HFFSCOA-based clustering. This HFFSCOA provides dependable and perfect paths connecting vehicular nodes for route clustering to build and evaluate optimum CHs in the network. The limitations are pointed as follows:

- Modified-PLGPa reduces unnecessary packet transmissions, the protocol still incurs additional computational overhead.
- VAD-FCOPRAS approach may not scale well to extremely large or highly dynamic networks, where real-time reconfiguration is required.
- PFSVT-MCDM has high computational overhead can render it unsuitable for real-time scenarios.
- DERNNets with GWO needs computational power makes these approaches less feasible for lightweight WSNs.

Some of the issues are recognized by examining the aforementioned literature. The optimization technique is predicated on the idea that any node may be a part of any instance as long as it seems to be related to traffic or forward instances. Moreover, vampire assaults may cause a depletion of energy, resulting in a reduction in network longevity and an escalation in average energy consumption. Additionally, it might result in energy waste through vampire attacks, which would shorten the network's lifespan and raise its average energy consumption. The creation of the suggested ELM_SSA approach, which will be addressed in the next section, is driven by the aforementioned limitations of the literature. ELM bypasses iterative processes for weight optimization by using randomly assigned hidden layer parameters and solving output weights in a single step using a closed-form solution.

3. PROPOSED MODEL

Sensors are deployed in the target environment and strategically placed to collect relevant data as shown in Figure 1. These sensors could be for measuring temperature, humidity, pressure, motion, light, sound, and so on. Each sensor node is configured with specific parameters, such as sampling frequency, data transmission protocols, and data format. This configuration ensures that the sensor nodes operate efficiently and transmit data accurately. From the collected data, some of the features, such as energy consumption patterns, node proximity, and communication traffic-oriented features, are extracted. Now, the features are trained using the Extreme Learning Machine (ELM) to predict vampire attacks with the assistance of the sleep scheduling algorithm for better energy efficiency. The sleep scheduling algorithm determines the optimal sleep-wake schedule for each sensor node based on factors, such as energy levels, data traffic patterns, and communication requirements. Now, ELM_SSA is deployed on the base station (BS) to enhance energy efficiency.



Data collection from sensor nodes
Extraction of features
Energy consumption patterns
Node proximity
Communication traffic
Train the features using Extreme Learning Machine (ELM) to predict vampire attack
Sleep scheduling algorithm (SSA)
Deploy ELM_SSA on base station

Figure 1. Block diagram to enhance energy efficiency using machine learning

3.1 Energy model

The energy required for packet transmission is denoted as En_t and is influenced by the length of the message segment I_m and the distance between the transmitter and receiver, d_t. Consequently

$$En_t(I_m, d_t) = I_m(E_{el} + E_{cm}) \tag{1}$$

$$En_t(I_m, d_t) = I_m\left(E_{el} + \varepsilon_{sf} d_l^2\right) if\ d_t <= d_l \tag{2}$$

$$En_t(I_m, d_t) = I_m\left(E_{el} + \varepsilon_{sf} d_t^2\right) if\, d_l < d_t < d_u \tag{3}$$

$$En_t(I_m, d_t) = I_m(E_{el} + \varepsilon_{md} d_t^4)\ if\, d_t \geq d_u \tag{4}$$

E_{cm} represents the amount of energy used in transmitting a single bit of information. The symbol ε_{sf} represents the free space method, whereas ε_{md} represents the multipath channel model. d_l and d_u is the lower and upper bounds of the distance d_t, whereas E_{el} denotes the energy use reference point. The energy used during packet receipt is mostly determined by the length of the message segment, denoted as I_m. Consequently,

$$E_{rec}(I_m) = I_m \times E_{el} \tag{5}$$

3.2 Analysis of vampire attack in the network

The network's nodes have the responsibility of both accurately transmitting data and modifying received data packets with new information before transferring them to the next node. Therefore, data transported between nodes must be verified as secure. A node loses trust after an aggressive or passive attack. Carousal attacks are instances when a malicious node creates a loop with a data packet, therefore obstructing its intended path to the BS or destination node. The node's aberrant conduct rapidly depletes its energy, resulting in a vampire assault. In the event that the BS does not receive the data packet within the designated timeframe, it will request the CH to assess the trust levels of all member nodes.

$$T^d(t) = \frac{p_{n1}(t)}{p_{n2}(t)} \tag{6}$$

where， $T^d(t)$ represents the direct trust calculated between the nodes $n1$ and $n2$. The variable $p_{n1}(t)$ represents the packets that have been received.

The total number of packets transmitted is denoted as $p_{n2}(t)$. The estimated trust of the neighboring nodes is indicated in Eq. (6).

$$T^t(t) = \frac{1}{k}\sum_{d=1}^{k} T^d(t) \tag{7}$$

$$T = \alpha T^d(t) + \beta T^t(t) \tag{8}$$

The values of α, β vary from 0 to 1, with the condition that $\alpha + \beta = 1$. The value has been evaluated from the Eqs. (7) and (8). The trust threshold ranges from 0.99 to 1. If the predicted value is below this level, the sensor is an MN because it loses more packets during transmission. To ensure efficient data transfer, the trust requirements are considered while selecting the route for sending packets. The BS will inform the CH of the MN's ID and position after calculating the trust degree during verification. The CHs have taken the action of blacklisting and isolating the rogue node, while also disseminating its ID to all of their members.

3.3 Clustering process

Every node in the network actively engages in the clustering process and autonomously determines whether it is a Head Node or a Member Node. Neighbors of a certain node are defined as all the nodes that fall inside its communication radius. The energy parameter of the nodes is used to form clusters, which are regularly updated. The CH selection technique adopts the K-Medoids algorithm with an upgraded version of Adaptive Sailfish Optimisation (IASFO) for better effectiveness. After clustering the sensor nodes, a CH is selected for each WSN cluster. The CH's main task is to gather and send cluster node data to the BS. This paper suggests WSN clustering using K-medoids. K-medoids group all sensors. K-medoids associates each cluster with one item. A medoid, the cluster's centre, has been identified. The group of K-medoids linked to a cluster node is the shortest distance between clusters because they find the best centre. Improves sensor node communication, decreases energy use, and locates cluster centres to eliminate packet delays. Efficiency and a defined number of convergence stages characterise K-medoids. The K-medoids clustering technique follows these steps.

Step 1: Randomly choose k points from the input data.

Step 2: Every individual data point is allocated to the cluster including the center point that is closest to it.

Step 3: Determine and add the distance between all cluster 'i' data points. The centroid of the i cluster is the point with the lowest calculated distance from all others.

Step 4: Continue iterating steps 1 and 3 until convergence is achieved, meaning that the central point stops moving.

Clustering classifies sensor nodes and selects CHs for each WSN group. Data from a cluster node is sent to the BS through the CH. The K-medoids technique is used to achieve cluster formation by identifying precise core clusters or centroids. This approach leads to reduced power consumption, minimal packet delay, and improved performance of sensor nodes. The procedure entails determining the approximate number of clusters and computing the initial CH node by

$$c = \sqrt{\frac{n}{2}} \tag{9}$$

where, the number of nodes is n. These methods calculate the beginning average point and central location (L) for all nodes.

$$L = \frac{\sum_{n=1}^{N} x_n}{n} \tag{10}$$

where, x_n represents the sensor's coordinate. The formula represents the average spacing among the SN and L as D.

$$D = \frac{\sum_{n=1}^{N} |x_n - L|}{n} \tag{11}$$

The SN distance from the center location L may be utilized to compute the centroid. Use Eq. (10) frequently to cluster until a CH is selected. Algorithm 1 shows the process of creating clusters by K-medoids using Sailfish Optimizer.

Algorithm-1 K-medoids clustering algorithm with enhanced Sailfish Optimizer

Network Initialization

Step 1: Initializing the WSN.

Step 2: Place BS at the coordinates (50, 180).

Step 3: Arrange the SNs in any random order.

Cluster formation with K-medoids and CH selection with ASFO

Step 4: N nodes split into many clusters.

Step 5: Each cluster consists of N nodes, and each node is connected to the closest CH.

Step 6: Choose the first random medoid in the cluster from N at random to become the initial CH.

Step 7: Every normal node to CH generates 3D coordinates (x, y, and z).

```
Step 8: Calculating K-means distance is conducted by the CH.
Step 9: The cluster node is centered, and the new CH is chosen using the ASFO method.
    Initialize the population
    While (stopping criterion not met) do
    For each sailfish in population do
    Evaluate the fitness based on objective function
    Update the personal best position and fitness
    End for
Step 10: Repeat step 7 to 9 until the node in the absolute center is found.
End
```

3.4 Extreme Learning Machine

Anomaly-based intrusion detection at the BS uses the ELM. All packet behaviors are established via anomaly detection. Implementing ML algorithms requires more energy, which sensor nodes cannot handle. ML is applied at the BS. ELM is a feedforward neural network with one or more hidden nodes. Hidden nodes' input weights and biases may be given arbitrarily, but the output layer-hidden layer weights can be determined analytically. Figure 2 shows that ELM learns the output weights of the hidden nodes in one step, making it faster than BPN.

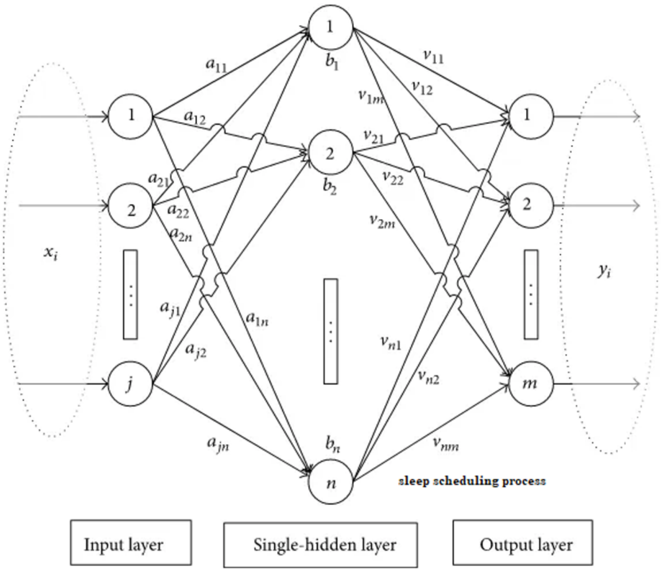


Figure 2. Architecture of ELM-SSA

The following is a method for a single hidden layer ELM. The output function of the i^{th} hidden node received a

$$h_i(x) = G(a_i, b_i, x) \tag{12}$$

The i -th hidden node parameters are a_i and b_i. The ELM output function is

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) \tag{13}$$

Here, L is the number of hidden nodes, and β_i is the output weight of hidden node. The hidden output mapping is

$$h(x) = [G(h_i(x), \ldots . h_L(x)] \tag{14}$$

Similarly, the output matrix H of the hidden layer is given as

$$H = \begin{bmatrix} h(x_1) \\ \ldots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} G(a_1, b_1, x_1) & \ldots & G(a_L, b_L, x_L) \\ \ldots & \ldots & \ldots \\ G(a_1, b_1, x_N) & \ldots & G(a_L, b_L, x_N) \end{bmatrix} \tag{15}$$

where, N is training samples. The target matrix T for the training data is provided as

$$T = \begin{bmatrix} t_1 \\ \ldots \\ t_N \end{bmatrix} \tag{16}$$

The ELM algorithm employs a simple form

$$Y' = W_2 \sigma(W_1 x) \tag{17}$$

The hidden layer's input weights are represented by W_1 and its output weights by W_2. The symbol σ denotes an activation function. The algorithm operates in the following manner.

- The input weights of the hidden layer, W_1, are allocated random values.
- W_2 is calculated by least-squares fitting a matrix of response variables Y using the pseudoinverse.

Algorithm-2 Extreme learning machine for sink hole attack detection

```
Step 1: Initialize the ELM parameters.
    Input_size=features extracted from network traffic (Energy consumption patterns, Node proximity, Communication traffic)
    Hidden layer_size=number of neurons in the hidden layer
    Output size=2 (normal or sinkhole attack)
Step 2: Generate random input weights and biases.
    Input_weights=random_matrix(input_size, hidden_layer_size)
    Hidden_biases=random_vector(hidden_layer_size)
Step 3: Define activation function.
    Def activation_function (x);
    Return sigmoid(x)
Step 4: Generate random outputs.
    Output_weights=random_matrix (hidden_layer_size, output_size)
Step 5: Train ELM.
    Compute the hidden layer output
    Calculate the output eright
Step 6: Test the ELM.
    Output=dot_product(hidden_layer_output, output_weights)
Step 7: Detect sinkhole attack.
    Def detect_sinkhole(input_data, output_eweights, threshold);
    Output=test_ELM(imput_data, output_weights)
    If output>threshold
    Return "sinkhole attack detected:
Else
Return "normal behavior".
```

This section provides a detailed description of the proposed Hierarchical Sleep Scheduling Mechanism (HSSM). The candidate-CHs disseminate their participation information to neighboring nodes within the competition radius using the appropriate transmission power. The comp_MSG message includes the candidate-CH ID and residual energy. If they get that message, additional candidate-CHs will add its ID to the neighbor candidate-CH database. The competition radius from candidate-CHs may vary, causing the following. If s_i's competition radius is bigger than candidate-CH, s_j and s_j may receive the $comp\_MSG$ message from s_i . Due to $s_j{}'$ s transmission range, the $comp\_MSG$ message cannot be delivered to s_i, hence s_i will not know about candidate-CH s_j. Any candidate-CH must estimate the sender's distance after receiving the $comp\_MSG$ message to learn about nearby rivals. Reissue a $comp\_MSG$ message to the sender if the distance exceeds their competition's radius. Thus, it may let senders get comprehensive neighbor candidate-CH information and update the database. Candidate-CH compares its residual energy to the sender's after receiving $comp\_MSG$. If its energy is less than the sender's, it quits the competition, becomes a member node, and broadcasts quit_MSG. Otherwise, it will wait for the $comp\_MSG$ from other competitors until the conclusion of the CH selection. If it does not withdraw from the competition, it will broadcast a sus_MSG message to all nodes in its transmission range to announce election to the CH and change the state fag. If a candidate-CH gets *quit_MSG*, the node checks the final state. Already a CH or member, it drops this notice. The sender will be updated to the member state in the neighbor candidate-CH table if the node is still in the candidate-CH state, and it will wait for messages from other neighbor candidate-CHs to determine its ultimate state. Members submit data to CH through TDMA. The CH will first divide data collecting time into numerous time slots and assign each interval to a member node to schedule data aggregation. The CH then sets its transmission range and sends *sched_MSG* to all member nodes for scheduling based on distance. Once the *sched_MSG* arrives, the member nodes record the data transmission time interval. In non-transmission slots, they may switch off the wireless communication device to conserve energy. While receiving data from all member nodes, the CH restores them and uses fuzzy clustering to identify redundant nodes.

When alerted, nodes in the redundant set $\{R_1, R_2, \ldots . R_t\}$ will adjust their status information and be scheduled for dormancy in the following cycle. Later, the CH would aggregate data and transmit it singly to the BS. Final round: BS gets all CH communications and ends round. These nodes will become members and clean their neighbor CHs tables after each round.

4. PERFORMANCE ANALYSIS

The Simulation model was constructed by MATLAB to analyses deployment real-time of network sensors in several facets of expanding the number of nodes Within their operational range, the detachable sensor nodes can communicate with the base station or other nodes. It is therefore possible to employ the nodes that are situated between the base station and other nodes as retranslates.

The simulation configuration of the Extreme Learning Machine with Sleep scheduling algorithm (ELM_SSA) has 100 mobile nodes that are randomly dispersed throughout a terrain area of 1000×1000 square meters. The pause time and simulation duration used to validate ELM_SSA's performance is 20 and 300 seconds, respectively. With a constant bit rate data source and a 2 Mbps channel capacity, the 802.11 MAC protocol is used in the ELM_SSA simulation environment. The simulation study also makes use of source and destination pairs with 20 and 50 mobile nodes, respectively.

The proposed ELM_SSA has been simulated in NS-2. The simulation factors employed for the accomplishment of the recommended ELM_SSA and the benchmarked Modified-PLGPa [20], VAD-FCOPRAS [22], and PFSVT-MCDM [23] approaches are analyzed in Table 1.

Table 1. Parameter settings

Parameter	Range
Number of Hidden Neurons (L)	10-200
Activation Function	ReLU
Regularization Factor (C)	0.1–10
Salp Velocity (v)	0.5–2.0
Exploration and Exploitation Balance (α)	0.7
Convergence Rate (Cr)	0.2

- **Throughput**

Data rate is the flow of data over a communication connection. Within a WSN setting, throughput is a critical metric that is evaluated while nodes are in motion without any concurrent traffic.

$$\text{Throughput (bits/sec)} = \sum \frac{(\text{No of successful pkts})*(\text{avg pkt size})}{\text{Total Time sent in delivering that amount of data}}$$

Figure 3 depicts the throughput analysis of existing Modified-PLGPa, VAD-FCOPRAS, and PFSVT-MCDM with the proposed ELM_SSA. When compared, existing methods achieve 11.45%,18.21%, and 21.29% of throughput but the proposed ELM_SSA method obtains 56.23% of throughput, which is 44.78%, 38.02%, and 34.94% better than the existing methods, as shown in Table 2.

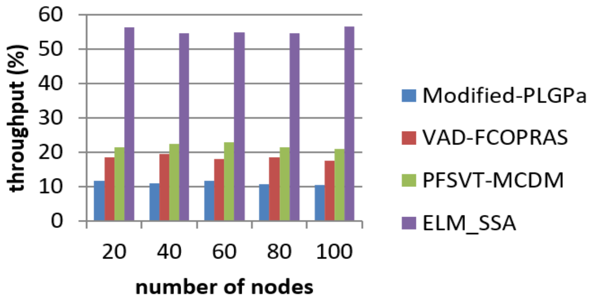


Figure 3. Calculation of throughput

Table 2. Analysis of throughput

Number of Nodes	Modified-PLGPa	VAD-FCOPRAS	PFSVT-MCDM	ELM_SSA
20	11.76	18.45	21.56	56.45
40	10.98	19.45	22.56	54.67
60	11.78	17.98	22.86	54.87
80	10.64	18.56	21.56	54.76
100	10.56	17.645	20.98	56.56

- **Packet delivery ratio (PDR)**

The success rate of packet transmission from the origin node to the target node in the network.

$$PDR = \frac{number\ of\ packet\ received\ succesfully}{Total\ number\ of\ packets\ forwarded}$$

Figure 4 depicts the PDR comparison of existing Modified-PLGPa, VAD-FCOPRAS, and PFSVT-MCDM with the proposed ELM_SSA. When compared, existing methods achieve 67.34%, 34.56%, and 22.38% of PDR but the proposed ELM_SSA method obtains 78.34% of PDR, which is 11%, 43.78%, and 55.96% better than the existing methods, as shown in Table 3.

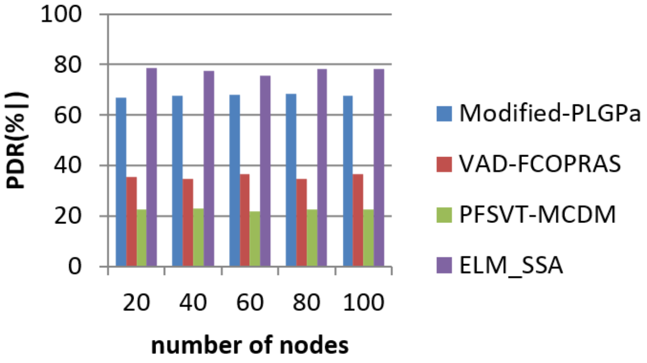


Figure 4. Comparison of PDR

Table 3. Performance of PDR

Number of Nodes	Modified-PLGPa	VAD-FCOPRAS	PFSVT-MCDM	ELM_SSA
20	66.78	35.45	22.56	78.45
40	67.43	34.65	22.76	77.45
60	67.89	36.34	21.56	75.67
80	68.453	34.65	22.56	78.23
100	67.43	36.56	22.65	78.23

- **Energy consumption**

This is quantified as the total energy of all hops and is calculated as

$$Energy = \frac{1}{p}\sum_{n}^{p} E_n$$

In multihop routing, p represents the number of hops and E_n represents the energy of the n^{th} hop.

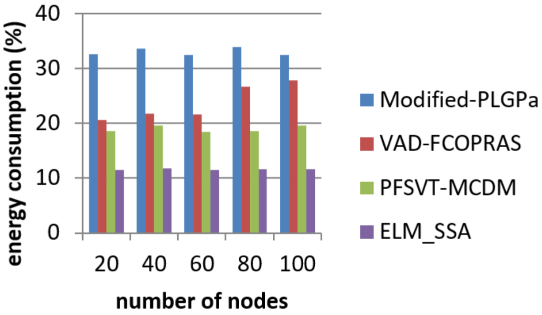


Figure 5. Calculation of energy consumption

Table 4. Performance of energy consumption

Number of Nodes	Modified-PLGPa	VAD-FCOPRAS	PFSVT-MCDM	ELM_SSA
20	32.56	20.645	18.65	11.54
40	33.56	21.76	19.54	11.78
60	32.45	21.67	18.43	11.54
80	33.867	26.67	18.54	11.67
100	32.45	27.87	19.54	11.57

Figure 5 depicts the energy consumption analysis among existing Modified-PLGPa, VAD-FCOPRAS, and PFSVT-MCDM with the proposed ELM_SSA. When compared, existing methods achieve 32.56%, 20.19%, and 18.92% of energy consumption but the proposed ELM_SSA method obtains 11.45% of energy consumption, which is 21.11%, 8.74%, and 7.86% better than the existing methods, as shown in Table 4.

- **End-to-end delay**

The ratio of the total count of hops (p) necessary for routing to the total quantity of nodes (n) is

$$Delay = \frac{p}{tn}$$

Figure 6 illustrates the end-to-end delay of existing Modified-PLGPa, VAD-FCOPRAS, and PFSVT-MCDM with the proposed ELM_SSA. When compared, existing methods achieve 23.45%, 17.94%, and 21.84% of end-to-end delay but the proposed ELM_SSA technique obtains 11.56% of end-to-end delay, which is 21.11%, 8.76%, and 10.67% better than the existing methods, as shown in Table 5.

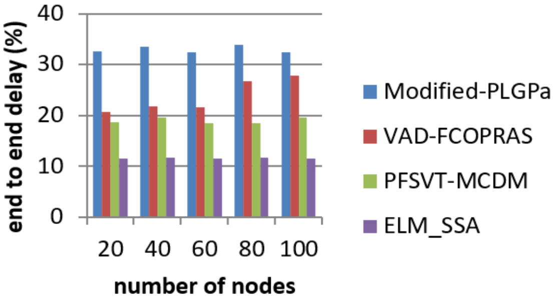


Figure 6. Calculation of end-to-end delay

Table 5. Performance of end-to-end delay

Number of Nodes	Modified-PLGPa	VAD-FCOPRAS	PFSVT-MCDM	ELM_SSA
20	23.45	17.53	21.67	11.54
40	24.54	16.67	21.56	10.54
60	23.65	16.87	20.54	11.64
80	25.56	17.43	21.67	10.65
100	23.54	17.89	21.78	10.765

Vampire node detection rate: The detection rate of vampire nodes is the proportion of network vampire nodes accurately recognized by a detection algorithm. To formulate this, it is necessary to define some terms.

$$detection\ rate = \frac{N_d}{N_v} \times 100\%$$

where, N_v is the total quantity of vampire nodes, and N_d is quantity of vampire nodes detected by the algorithm.

Figure 7 depicts the vampire node detection rate comparison of existing Modified-PLGPa, VAD-FCOPRAS, and PFSVT-MCDM with the proposed ELM_SSA. When compared, existing methods achieve 35.45%, 19.31%, and 18.45% of vampire node detection rates, while the proposed ELM_SSA method obtains 72.34% of vampire node detection rate, which is 45.65%, 56.87%, and 45.78% better than the existing methods, as shown in Table 6.

Table 7 shows all comparison analysis values between

Modified-PLGPa, VAD-FCOPRAS, PFSVT-MCDM, ELM_SSA for all metrics.

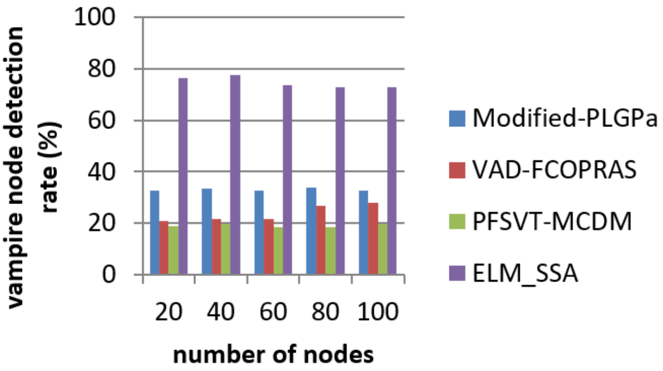


Figure 7. Comparison of vampire node detection rate

Table 6. Analysis of vampire detection rate

Number of Nodes	Modified-PLGPa	VAD-FCOPRAS	PFSVT-MCDM	ELM_SSA
20	35.45	19.56	18.54	76.34
40	34.54	18.98	18.765	77.45
60	36.45	19.453	18.78	73.45
80	35.65	19.76	18.54	72.67
100	36.654	19.54	18.69	72.75

Table 7. Overall comparative analysis

Parameters	Modified-PLGPa	VAD-FCOPRAS	PFSVT-MCDM	ELM_SSA
Throughput (%)	11.45	18.21	21.29	56.23
Vampire node detection rate (%)	35.45	19.31	18.45	72.34
Energy consumption (%)	32.56	20.19	18.92	11.45
End-to-end delay (%)	23.45	17.94	21.84	11.56
Packet delivery ratio (%)	67.34	34.56	22.38	78.34

5. CONCLUSION

This study presented a distinctive design approach to reduce vampire attacks and enhance network performance by using an energy-efficient routing mechanism. The Data was optimised using hybrid clustering, and the vampire attack was neutralised with extreme machine learning. Data optimization might lead to enhanced performance. The proposed model was evaluated using NS2 via simulation. The experimental findings indicated that the model achieved a throughput of 56.23%, vampire node detection rate of 72.34%, energy consumption of 11.45%, end-to-end delay of 11.56%, and packet delivery ratio of 78.34%. The ELM_SSA method is trained on known attack scenarios. While ELM has good generalization capabilities, it may still struggle with detecting novel or unknown attacks that were not part of the training set. Future research will focus on implementing blockchain technology in WSNs that can enhance security and trustworthiness while enabling transparent and tamper-proof logging of node behavior. This could investigate how blockchain can be utilized for decentralized vampire node detection and mitigation.

REFERENCES

[1] Tanenbaum, A.S. (2002). Network protocols. ACM Computing Surveys (CSUR), 13(4): 453-489. https://doi.org/10.1145/356859.356864

[2] Awoyemi, B.S., Alfa, A.S., Maharaj, B.T. (2019). Network restoration in wireless sensor networks for next-generation applications. IEEE Sensors Journal, 19(18): 8352-8363. https://doi.org/10.1109/JSEN.2019.2917998

[3] Awad, A.S., Khalaf, M., Alsaadi, M. (2024). Deep learning-enhanced cluster head optimization for intrusion detection in wireless sensor networks. Ingénierie des Systèmes d'Information, 29(2): 609. https://doi.org/10.18280/isi.290222

[4] Yang, K. (2014). Wireless Sensor Networks. Design and Applications. Springer, London, UK. https://doi.org/10.1007/978-1-4471-5505-8

[5] Liu, J., Huang, K., Yao, X. (2018). Common-innovation subspace pursuit for distributed compressed sensing in wireless sensor networks. IEEE Sensors Journal, 19(3): 1091-1103. https://doi.org/10.1109/JSEN.2018.2881056

[6] Mabrouki, J., Azrour, M., Dhiba, D., Farhaoui, Y., El Hajjaji, S. (2021). IoT-based data logger for weather monitoring using Arduino-based wireless sensor networks with remote graphical application and alerts. Big Data Mining and Analytics, 4(1): 25-32. https://doi.org/10.26599/BDMA.2020.9020018

[7] Zhou, L., Haas, Z.J. (1999). Securing ad hoc networks. IEEE Network, 13(6): 24-30. https://doi.org/10.1109/65.806983

[8] O'Mahony, G.D., Curran, J.T., Harris, P.J., Murphy, C.C. (2020). Interference and intrusion in wireless sensor networks. IEEE Aerospace and Electronic Systems Magazine, 35(2): 4-16. https://doi.org/10.1109/MAES.2020.2970262

[9] Xie, H., Yan, Z., Yao, Z., Atiquzzaman, M. (2018). Data collection for security measurement in wireless sensor networks: A survey. IEEE Internet of Things Journal, 6(2): 2205-2224. https://doi.org/10.1109/JIOT.2018.2883403

[10] Butun, I., Österberg, P., Song, H. (2019). Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. IEEE Communications Surveys & Tutorials, 22(1): 616-644. https://doi.org/10.1109/COMST.2019.2953364

[11] Yao, S., Li, Z.W., Guan, J.F., Liu, Y. (2019). Stochastic cost minimization mechanism based on identifier network for IoT security. IEEE Internet of Things Journal, 7(5): 3923-3934. https://doi.org/10.1109/JIOT.2019.2961839

[12] Abdalzaher, M.S., Muta, O. (2020). A game-theoretic approach for enhancing security and data trustworthiness in IoT applications. IEEE Internet of Things Journal, 7(11): 11250-11261. https://doi.org/10.1109/JIOT.2020.2996671

[13] Vasserman, E.Y., Hopper, N. (2011). Vampire attacks: Draining life from wireless ad hoc sensor networks. IEEE Transactions on Mobile Computing, 12(2): 318-

332. https://doi.org/10.1109/TMC.2011.274

[14] Kannan, G., Indragandhi, K., Jan, M. (2024). An anomaly detection system for vampire attacks crisis in wireless sensor networks. Computer Networks and Communications, 2(1): 101-110. https://doi.org/10.37256/cnc.2120244226

[15] Karthick, P.T., Palanisamy, C. (2019). Optimized cluster head selection using krill herd algorithm for wireless sensor network. Automatika: Časopis za Automatiku, Mjerenje, Elektroniku, Računarstvo i Komunikacije, 60(3): 340-348. https://doi.org/10.1080/00051144.2019.1637174

[16] Morsy, N.A., AbdelHay, E.H., Kishk, S.S. (2018). Proposed energy efficient algorithm for clustering and routing in WSN. Wireless Personal Communications, 103: 2575-2598. https://doi.org/10.1007/s11277-018-5948-2

[17] Awad, A.S., Hasan, E.H., Obaid, M.A. (2023). HITR-ECG: Human identification and classification simulation system using multichannel ECG signals: Biometric systems era. In International Conference on Innovative Computing and Communication, Delhi, India, pp. 171-181. https://doi.org/10.1007/978-981-99-3315-0_14

[18] Elhabyan, R., Shi, W., St-Hilaire, M. (2018). A Pareto optimization-based approach to clustering and routing in Wireless Sensor Networks. Journal of Network and Computer Applications, 114: 57-69. https://doi.org/10.1016/j.jnca.2018.04.005

[19] Hamzah, A., Shurman, M., Al-Jarrah, O., Taqieddin, E. (2019). Energy-efficient fuzzy-logic-based clustering technique for hierarchical routing protocols in wireless sensor networks. Sensors, 19(3): 561. https://doi.org/10.3390/s19030561

[20] Mohammed, B.M., Alsaadi, M., Khalaf, M., Awad, A.S. (2024). Game theory-based multi-hop routing protocol with metaheuristic optimization-based clustering process in WSN for precision agriculture. Journal Européen des Systèmes Automatisés, 57(3): 653-662. https://doi.org/10.18280/jesa.570302

[21] Bhatti, D.S., Saleem, S., Ali, Z., Park, T.J., Suh, B., Kamran, A., Buchanan, W.J., Kim, K.I. (2024). Design and evaluation of memory efficient data structure scheme for energy drainage attacks in wireless sensor networks. IEEE Access, 12: 41499-41516. https://doi.org/10.1109/ACCESS.2024.3377144

[22] Madhavi, S., Udhaya Sankar, S.M., Praveen, R., Jagadish Kumar, N. (2023). A fuzzy COPRAS-based decision-making framework for mitigating the impact of vampire sensor nodes in wireless sensor nodes (WSNs). International Journal of Information Technology, 15(4): 1859-1870. https://doi.org/10.1007/s41870-023-01219-5

[23] Madhavi, S., Santhosh, N.C., Rajkumar, S., Praveen, R. (2023). Pythagorean Fuzzy Sets-based VIKOR and TOPSIS-based multi-criteria decision-making model for mitigating resource deletion attacks in WSNs. Journal of Intelligent & Fuzzy Systems, 44(6): 9441-9459. https://doi.org/10.3233/JIFS-224141

[24] Venkatesh, A., Asha, S. (2023). DERNNet: Dual encoding recurrent neural network based secure optimal routing in WSN. Computer Systems Science & Engineering, 45(2): 1375-1392. http://doi.org/10.32604/csse.2023.030944

[25] Thomas, D., Shankaran, R., Orgun, M.A., Mukhopadhyay, S.C. (2021). Sec 2: A secure and energy efficient barrier coverage scheduling for WSN-based IoT applications. IEEE Transactions on Green Communications and Networking, 5(2): 622-634. https://doi.org/10.1109/TGCN.2021.3067606

[26] Alkwai, L.M., Mohammed Aledaily, A.N., Almansour, S., Alotaibi, S.D., Yadav, K., Lingamuthu, V. (2022). Vampire attack mitigation and network performance improvement using probabilistic fuzzy chain set with authentication routing protocol and hybrid clustering-based optimization in wireless sensor network. Mathematical Problems in Engineering, 2022(1): 4948190. https://doi.org/10.1155/2022/4948190

[27] Shakya, V., Choudhary, J., Singh, D.P. (2024). IRADA: Integrated reinforcement learning and deep learning algorithm for attack detection in wireless sensor networks. Multimedia Tools and Applications, 83: 71559-71578. https://doi.org/10.1007/s11042-024-18289-7

[28] Obaid, M.A., Awad, A.S., Khaleel, I.S. (2022). An optimal cluster head selection with trusted path routing and classification of intrusion in WSN Employing CHLNNet. Ingenierie des Systemes d'Information, 27(5): 685-693. https://doi.org/10.18280/isi.270501

[29] Li, Y., Zhao, L., Wang, Y., Wen, Q. (2024). Improved sand cat swarm optimization algorithm for enhancing coverage of wireless sensor networks. Measurement, 233: 114649. https://doi.org/10.1016/j.measurement.2024.114649

[30] Meera, V.K., Balasubramanian, C. (2024). A hybrid fennec fox and sand cat optimization algorithm for clustering scheme in VANETs. Sustainable Computing: Informatics and Systems, 42: 100983. https://doi.org/10.1016/j.suscom.2024.100983