scientific reports

Check for updates

OPEN

EDSSR: a secure and power-aware opportunistic routing scheme for WSNs

Ruili Yang[1], Manoj A. Patil[2,3], Potu Narayana[4], D. Jayaram[5], K. Keerthi[6], N. SudhakarYadav[5], Premkumar Chithaluru[7]✉, Sunil Kumar[8,9], Diaa Salama Abd Elminaam[10,11] & Deema Mohammed Alsekait[12]✉

Motivated by the pivotal role of routing in Wireless Sensor Networks (WSNs) and the prevalent security vulnerabilities arising from existing protocols, this research tackles the inherent challenges of securing WSNs. Many current WSN routing protocols prioritize computational efficiency but lack robust security measures, making them susceptible to exploitation by malicious actors. The prevalence of reactive protocols, chosen for their lower bandwidth consumption, exacerbates security concerns, as proactive alternatives require more resources for maintaining network routes. Additionally, the ad hoc nature and energy constraints of WSNs render conventional security models designed for wired and wireless networks unsuitable. In response to these limitations, this paper introduces the Secured Energy-Efficient Opportunistic Routing Scheme for Sustainable WSNs (EDSSR). EDSSR is designed to enhance security in WSNs by continuously updating neighbor information and validating the legitimacy of standard routing parameters. Critically, the protocol is power-aware, recognizing the vital importance of energy considerations in the constrained environment of WSNs. To assess the efficacy of EDSSR in mitigating WSN vulnerabilities, simulation experiments were conducted, evaluating the protocol's performance on key metrics such as throughput, average End-to-End delay (E^2 delay), energy consumption (EC), network lifetime (alive nodes), and malware detection rate. The results demonstrate that the EDSSR protocol significantly improves performance. It shows substantial gains in sum goodput relative to throughput, average E^2 delay, EC, and alive nodes. Specifically, the EDSSR protocol is 2–3% faster than DLAMD and 10–13% faster than EEFCR. Additionally, the malware detection rate increases by 23%.

The advent of Wireless Networks (WN) has sparked a revolution in global communication, outpacing wired networks in popularity. This shift is attributed to the advantages of wireless networks, including mobility, the absence of physical media, and easy installation[1]. The increasing reliance on wireless communication for daily tasks has led to a substantial rise in cellular subscribers, encompassing both voice and data services. This surge can be attributed to the decreasing cost of wireless hardware equipment, resulting in a reduced setup cost for wireless infrastructure. Ongoing research in wireless communication has facilitated the simultaneous sharing of voice and data on the same channel. The ubiquitous availability of wireless services at any time and place has firmly established it as the preferred mode of communication in contemporary times.

In the realm of WN, communication entails the wireless transmission of electromagnetic waves between different nodes[2,3]. A notable characteristic of wireless communication is its limited signal range, allowing

[1]FESCO Adecco, Shanghai 200000, China. [2]Department of Computer Science and Engineering, Christ (Deemed to be University) School of Engineering and Technology, Kengeri, Bengaluru, Karnataka 560074, India. [3]Department of Computer Science and Engineering, G. Narayanamma Institute of Technology and Science, Shaikpet, Hyderabad, Telangana 500104, India. [4]Stanley College of Engineering and Technology for Women, Abids, Hyderabad, India. [5]Department of Information Technology, Chaitanya Bharathi Institute of Technology (A), Gandipet, Hyderabad 500075, India. [6]Methodist College of Engineering and Technology, Abids, Hyderabad, India. [7]Department of Information Technology, Mahatma Gandhi Institute of Technology, Telangana 500075, India. [8]Department of Computer Engineering and Applications, GLA University, Mathura, India. [9]Department of Computer Science, Graphic Era Hill University, Dehradun 248001, India. [10]Faculty of Computers and Artificial Intelligence, Benha University, Banha, Egypt. [11]Jadara Research Center, Jadara University, Irbid 21110, Jordan. [12]Department of Computer Science, Applied College, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia. ✉email: chpremkumar_it@mgit.ac.in; Dmalsekait@pnu.edu.sa

only nodes within the proximity of the transmitter to receive the signal. Moreover, the transmission range is influenced by factors such as energy/power levels and the topography of the transmission region.

Routing attacks in WSN

In WSNs, routing is fundamental for data transfer, yet prevalent protocols lack robust security features, increasing vulnerability[4]. In the pursuit of efficiency, designers of routing protocols often compromise on security. Their focus on enhancing efficiency involves increased participation of all nodes in the routing process, inadvertently amplifying security risks[5]. The ensuing section introduces key routing attacks prevalent in WSNs.

- **Flooding attack:** This attack floods the network with an overwhelming volume of dummy data, causing congestion and resource depletion, such as node energy and bandwidth[6].
- **Black hole attack:** In this attack, a malicious node falsely claims to have the shortest route, diverting data towards itself and subsequently dropping the packets, denying service to the requesting node[7].
- **Spoofing:** Attackers steal the identity of a legitimate node, joining the network and disseminating fake routes to disrupt communication. Implementing security measures at each routing step is a potential remedy, albeit at the cost of increased computational overhead[8].
- **Wormhole attack:** Malicious nodes collaborate to extract crucial information, such as routing and control packets, from the network, disrupting the overall WN functionality[9].
- **Sybil attack:** This attack involves a malicious node overwhelming the destination node with a sudden surge of packets, causing disturbances in the routing process[10,11].

Defense mechanism for routing attacks in WSN

Adhering to the proverb "Prevention is Better than Cure," researchers prioritize designing routing protocols with built-in preventive measures for potential security threats in WSNs. However, the predominant strategy employed by developers is integrating encryption or key management solutions into routing procedures. Unfortunately, these solutions contribute to increased computational overhead, further depleting the already limited resources of WN nodes, including memory and energy. The following outlines existing methodologies addressing the control of routing attacks:

- **Trust-based techniques:** When direct communication is hindered by the short transmission range of nodes, multi-hop communication becomes essential. Nodes rely on their neighboring nodes to forward packets, emphasizing the importance of a strategic working strategy for quality transmission in the network[12,13].
- **Filtering techniques:** These techniques are typically employed to counter flooding attacks by filtering network traffic. Specific filters, based on different parameters of routing protocols, are implemented to assess the legitimacy of traffic between nodes. Nodes failing the condition test trigger actions to secure the network[14].
- **Cryptography and key management techniques:** Various encryption techniques are applied during data transmission to prevent unauthorized access to the network's data and resources[15].
- **Intrusion detection system-based:** This strategy focuses on detecting malicious activities in WSNs, presuming that the WN has already been attacked and a malicious node is present. Designing an Intrusion Detection System (IDS) in wireless environments is complex due to the characteristics of ad-hoc networks, posing challenges compared to wired architecture[16,17].
- **Agent-based techniques:** This technique involves embedding a software code in the WN to act as an agent that traverses the entire network, detecting potential malicious nodes. The agent exhibits adaptability, easily adjusting to sudden disconnections and other resource limitations within the WN. It randomly visits all nodes at varying intervals, tracing a random trajectory[18,19].

Motivation

The motivation for this research is rooted in the critical role that routing plays in WSNs and the substantial security vulnerabilities inherent in existing routing protocols. WSNs are pivotal in a wide range of applications, including environmental monitoring, healthcare, industrial automation, and smart cities. These networks rely on efficient and secure routing protocols to ensure reliable data transmission across sensor nodes. However, many current WSN routing protocols focus heavily on computational efficiency, often at the expense of robust security measures. This trade-off leaves WSNs vulnerable to a variety of malicious attacks, including eavesdropping, data tampering, and node capture, which can compromise the integrity and functionality of the entire network.

Reactive routing protocols are often preferred for their lower bandwidth consumption and reduced overhead. However, these protocols exacerbate security issues because they do not maintain consistent and secure routes, making the network more susceptible to dynamic threats. On the other hand, proactive routing protocols, while offering better security, require significant resources to maintain updated routes, which is impractical given the limited energy and processing capabilities of WSN nodes. This dichotomy creates a challenging landscape where the need for security conflicts with the need for efficiency.

The ad hoc nature and severe energy constraints of WSNs further complicate the implementation of traditional security models. These models, developed for more stable and resource-rich wired and wireless networks, are often too resource-intensive and inflexible to be effectively deployed in WSNs. Consequently, there is an urgent need for a new approach that balances security and efficiency, specifically tailored to the unique characteristics of WSNs.

Research gap

The research gap identified in this study is the significant lack of robust security measures in current WSNs routing protocols. These protocols often prioritize computational efficiency to conserve the limited resources of

sensor nodes but fail to adequately address the security vulnerabilities that make WSNs susceptible to a variety of attacks. Existing reactive protocols are favored for their lower bandwidth consumption, yet this comes at the cost of increased security risks since these protocols do not proactively secure network routes. Proactive protocols, while offering more security, demand greater resource expenditure, which is impractical for the constrained environments in which WSNs operate.

Moreover, the unique characteristics of WSNs, such as their ad hoc nature and severe energy constraints, are not effectively accommodated by traditional security models developed for more stable and resource-rich wired and wireless networks. These traditional models are often too resource-intensive and inflexible to be deployed in WSNs, resulting in insufficient protection against malicious activities like eavesdropping, data tampering, and node capture. The dynamic topology and intermittent connectivity typical of WSNs further complicate the implementation of conventional security solutions.

This research gap underscores the necessity for an innovative approach that simultaneously addresses both the security and energy efficiency needs of WSNs. By focusing on the creation of a proposed EDSSR, this research aims to fill this gap by providing a solution that not only enhances the security of WSNs but also takes into account their energy limitations. The development of EDSSR will involve continuous updating of neighbor information and validation of routing parameters to ensure secure communication paths. Additionally, it will incorporate power-aware features to extend the operational lifetime of the network, thereby maintaining a balance between security and efficiency.

The intention behind this research is to bridge the existing divide between computational efficiency and robust security in WSN routing protocols. By addressing these dual challenges, the proposed EDSSR protocol aims to significantly improve the resilience and reliability of WSNs, making them more effective for critical applications such as environmental monitoring, healthcare, and smart cities. The research will also contribute to the broader field by providing insights and methodologies that can be adapted to other types of ad hoc networks with similar constraints.

Problem statement

WSNs are essential for various applications, ranging from environmental monitoring to healthcare, due to their ability to gather and transmit data from numerous sensor nodes to centralized locations. However, the inherent characteristics of WSNs, such as limited power resources, dynamic topology, and wireless communication, pose significant challenges in maintaining both efficiency and security. Current WSN routing protocols often prioritize computational efficiency at the expense of robust security measures, leaving networks vulnerable to attacks like eavesdropping, data tampering, and node compromise. Reactive routing protocols, while efficient in bandwidth usage, lack adequate security features, whereas proactive protocols require more resources, making them unsuitable for the resource-constrained environment of WSNs. Furthermore, traditional security models developed for wired and wireless networks do not adapt well to the ad hoc and energy-constrained nature of WSNs, complicating the implementation of effective security mechanisms. These challenges underscore the urgent need for a security-focused and energy-efficient routing solution tailored specifically for WSNs.

The intrinsic ad-hoc characteristics of WSNs expose them to vulnerabilities, making them prone to attacks aimed at disrupting overall performance, especially by compromising the routing process. Despite dedicated research efforts to detect and prevent such attacks, achieving comprehensive protection remains a challenging endeavor. The complexities stem from the need to design protocols capable of adapting to the dynamic nature of networks, accommodating mobility, and addressing security concerns-all while operating within the constraints of limited resources. This challenging task underscores the necessity for innovative solutions that can adeptly navigate the intricacies of WSNs, providing robust security in the face of evolving threats.

The objective of the paper is as follows:

- Develop a Secured Energy-Efficient Opportunistic Routing Scheme (EDSSR) to enhance the security of WSNs.
- Continuously update neighbor information to maintain up-to-date routing data and improve network resilience.
- Validate the legitimacy of standard routing parameters to prevent unauthorized access and data manipulation.
- Design a power-aware protocol to address the energy constraints of WSN nodes, ensuring prolonged network operation.
- Evaluate the performance of EDSSR through simulation experiments, focusing on key metrics such as throughput, average E^2 delay, EC, network lifetime (alive nodes), and malware detection rate.
- Compare the efficacy of EDSSR with existing protocols, demonstrating its superiority in mitigating security vulnerabilities and improving overall network performance.The following section depicts the general structure of the study method: Section "Literature review" discusses the various peer-competing existing protocol and their limitations. Section "Proposed methodology" presents the proposed method and is tested with different scenarios. The result analysis of the proposed and existing protocols is presented in section "Results and discussion". Finally, section "Conclusion and future scope" concludes the paper and highlights the future scope.

Literature review

WSNs are known for their dynamic reconfigurability, self-organization, and rapid deployment, making them suitable for various applications, including military operations, rescue missions, and vehicular networks[20]. Despite their advantages, WSNs face security concerns due to their ad-hoc nature, wireless communication, lack of central administration, dynamic topology, and node mobility[21]. These challenges have led to various security threats, including flooding attacks, Sybil attacks, and false detection issues.

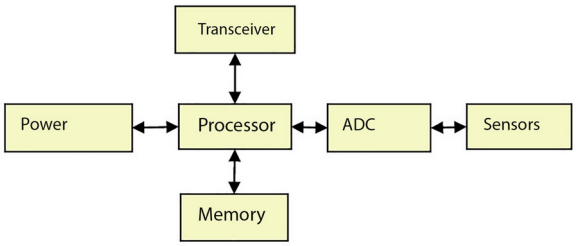

Fig. 1. Internal components of sensor node in WSN.

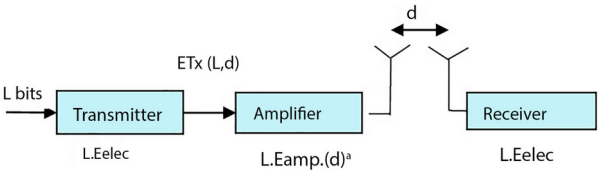

Fig. 2. Energy model of WSN.

In addressing the issue of flooding attacks, a proposed algorithm in[22] introduces predefined values for transmission, blacklisting, and whitelisting. The algorithm processes Route Request (RREQ) based on these values, discarding requests below the transmission value and blacklisting nodes exceeding the blacklist value. However, this approach may cease to function if predefined values are disturbed.

Another scheme in[23] utilizes a reputation-based model to tackle false detection problems. Nodes declared malicious receive near-zero reputation, limiting their RREQ. The reputation is periodically reset, allowing suspected nodes to regain acceptance if their reputation improves. In[24], a method leverages past RREQ data to prevent flooding attacks. The approach maintains a history table of previous RREQ and calculates an average limit, applying a discard limit to thwart flooding attacks. A collaborative non-distributed model in[25,26] organizes WSN nodes into a square grid, selecting Cluster Heads (CHs) based on connectivity and energy. This model aims to detect flooding attacks in a structured manner.

To address IP spoofing-based flooding,[27] explores a filtering technique deployed in autonomous systems. Probability and hop count filtering based on round trip time in[28] reduce packet filtering time, achieving nearly 99% detection of malicious packets. Packet modification during transit is a common security concern in WSN routing. Researchers in[29,30] propose mandatory node sign-ins or signatures to prevent unauthorized modifications and enhance security. A prediction model based on fuzzy logic and dynamic trust in[31] offers a trust-based Source Routing protocol for secured route selection, predicting future behavior based on historical data.

Due to the complexity of Android application features, identifying the best combination to distinguish between benign and malicious software is challenging. This paper proposes DLAMD, an efficient malware detection framework based on a deep neural network, designed for large-scale samples. DLAMD features a two-phase detection process: a rapid pre-detection phase and a deep detection phase. It analyzes Android application packages (APKs) to quickly extract permissions and opcode features that differentiate benign from malicious software. To further refine the feature subset, a random forest is used for importance selection, and a convolutional neural network (CNN) is employed to automatically extract hidden patterns[40].

WSNs, consisting of small sensor nodes with limited resources, play a crucial role in monitoring physical and environmental conditions remotely[34,35]. Their applications include tracking pressure, temperature, humidity, and pollution, making them valuable in various fields such as emergency response[36], transportation monitoring, and nuclear sensing[37,38]. Overall, these security measures and proposed algorithms aim to enhance the robustness of WSNs against various attacks and improve their reliability in critical applications.

Sensor nodes transmit physical phenomenon readings to the Base Station (BS), leveraging the BS's larger memory and processing power compared to the sensor nodes[39]. In WSNs, energy is expended during sensing, processing, transmitting, and receiving data (Fig. 1).

Energy dissipation for sending L-bits from a transmitter to a receiver is depicted in Fig. 2. Despite the advantage of easy deployment and scalability, WSNs face significant security threats due to their ad-hoc nature and energy constraints. Conventional security models used in wired and wireless networks are not directly applicable to WSNs, necessitating the development of power-aware security protocols. Energy-efficient and secure routing strategies have been proposed in the literature, considering the network structure and protocol operations.

An energy-optimized methodology based on a window scheme is proposed in[41] for detecting malicious nodes in WSNs. Computation is primarily performed at the sink node, which monitors the behavior of all nodes using data related to CHs and residual energy.[42] presents a fuzzy logic-based intrusion detection approach using parameters such as node energy, packet transmission rate, number of neighboring nodes, and transmission

Cluster routing protocols	Network	Cluster formation	CH selection	Algorithm complexity	CH role	Process dynamic	Location awareness
AREOR[30]	Homogeneous	Distributed	Random	$o(n^3)$	Relaying	Less	Required
I-AREOR[35]	Heterogeneous	Distributed	Node parameters	$o(n^3)$	Relaying	Less	Required
MTCEE-LLN[46]	Heterogeneous	Centralized	Node parameters	$o(n^3)$	Relaying	Less	Required
EEFCR[47]	Heterogeneous	Centralized	Random	$o(n^3)$	Relaying	Less	Required
Proposed	Heterogeneous	Centralized	Node parameters and network dynamics	$o(n^2)$	Aggregating	Dynamic	Not required

Table 1. Comparative analysis of proposed with peer competing routing protocols.

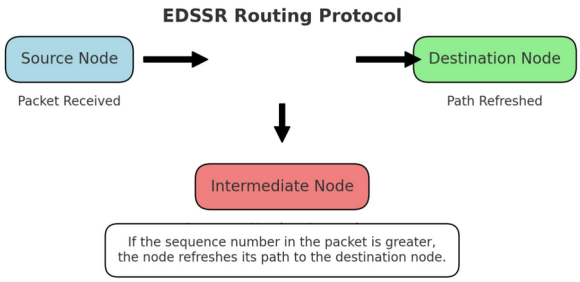

Fig. 3. Architecture of EDSSR.

errors for CH selection. The detection of denial-of-service attacks is based on these features, but manual settings for fuzzy logic systems are required.

In[43], the radio and transmission radius are explored to detect sinkhole attacks in WSNs. Fuzzy logic is employed for detection, but the methodology requires manual settings for the fuzzy logic system.[44] proposes a detection system using neighboring nodes to detect selective forwarding, jamming, and flooding attacks. While performance is satisfactory, the approach involves additional communication overhead, suffers from false alarms, and does not consider power consumption.

A decentralized methodology to prevent and detect node replication attacks is presented in[45], introducing predefined rules for monitor nodes. Monitors share gathered information, enhancing malicious node detection accuracy, though the risk of a monitor node becoming malicious remains.[46] proposes an integrated intrusion detection system with three individual IDSs, aiming to improve detection rates and reduce false positives. The method exhibits high computational complexity due to the back-propagation method and has low detection accuracy with high false alarms.

In[47], a machine learning solution for anomaly detection is introduced, focusing on feature extraction to identify temporal and spatial inconsistencies. Sequences of sensed values are sent to the BS, and clustering is used to identify data from malicious nodes. However, the system has a drawback as not all adverse information is shared among nodes. Table 1 shows the comparative analysis of the proposed method and peer-competing existing routing protocols.

Gaps identified in literature

After an extensive literature survey, it is evident that black hole attacks pose a significant threat to achieving the goals of WSNs. Specialized security mechanisms, such as key establishment, secure localization, secure aggregation, and secure routing, have been developed to address specific types of attacks in WSNs. Various authors have contributed to the prevention and detection of different attack types, either focusing on security or aiming to extend the lifespan of WSNs. However, a comprehensive mechanism that effectively addresses both energy and security concerns is currently lacking.

Upon critical analysis of existing work, several advantages and limitations of these techniques have been identified. While many methods protect WSNs from attacks, they often exhibit drawbacks such as complexity, low accuracy rates in detecting malicious nodes, and limited flexibility for applications. In flooding attacks, some research sets the Request rate limit ($PREQ_{R_L}$) value, but it may not follow the standard set by RFC 6553[32]. Certain preventive approaches utilize blacklisting to block nodes identified as malicious, potentially leading to the blockage of normal nodes. Additionally, some techniques lack a recovery mechanism for data loss caused by attacks[33]. Therefore, the development of a robust defense mechanism is essential to secure WSNs from malicious attacks.

Proposed methodology

The EDSSR architecture enhances security and energy efficiency in WSNs (Fig. 3). Unlike traditional protocols, EDSSR does not store multi-hop route information, simplifying the protocol and reducing memory usage. In this architecture, sensor nodes within the network communicate directly with their immediate neighbors, sending,

receiving, and forwarding data packets. Nodes maintain routing tables with direct neighbor information and destination sequence numbers, which help identify the most recent routes to destinations. These tables are periodically exchanged with neighboring nodes to keep the network topology updated.

When forwarding a packet, a node uses its routing table to determine the next hop based on the destination sequence numbers. Each packet contains a destination node sequence number, indicating the freshness of the route information. Upon receiving a packet, a node compares the sequence number in the packet with its routing table. If the packet's sequence number is higher, the node updates its routing table and refreshes its path to the destination, ensuring the use of the most recent and reliable route.

The data flow in the network is dynamic, relying on continuous updates of routing information. As nodes forward packets, they inform the network of the latest path information, optimizing routing and ensuring efficient data delivery. Key advantages of this architecture include the absence of multi-hop route storage, which minimizes memory usage, and the use of destination sequence numbers for up-to-date routing, improving reliability and efficiency. The protocol also focuses on reducing unnecessary transmissions and updates, conserving the energy of the sensor nodes. This design is particularly suited for the ad hoc and dynamic nature of WSNs, where nodes frequently join or leave the network, and energy conservation is critical for prolonging the network's operational lifetime.

The proposed EDSSR system for WSNs introduces a unique approach to packet forwarding that enhances security and efficiency (Fig. 4). Unlike traditional systems, EDSSR does not store information about multi-hop routes. Instead, it relies on routing table entries exchanged between neighboring nodes to facilitate packet forwarding. A key feature of this system is the use of destination sequence numbers, which ensure that nodes are aware of the most recent paths to their desired destinations. When a node receives a packet, it compares the destination sequence number in the packet with the last stored sequence number for the destination node. If the sequence number in the packet is higher, indicating a newer route, the node updates its path to the destination node accordingly. This mechanism allows EDSSR to dynamically adapt to changes in the network, ensuring up-to-date routing information and enhancing the robustness of the network against potential security threats such as flooding attacks. This approach not only optimizes the routing process but also strengthens the network's defenses by continuously validating and updating routing paths based on the most recent and reliable information available.

The proposed system has a key feature: it does not store any information about multi-hop routes. Packet forwarding relies on routing table entries exchanged between neighboring nodes in the EDSSR. A unique aspect of EDSSR's operation is the utilization of destination sequence numbers for packet forwarding. These numbers ensure that the node is aware of the most recent path to a desired destination. When a node receives a packet, it compares the destination node's sequence number in the packet to the last stored sequence number. If the sequence number in the packet is greater, the node refreshes its path to the destination node. The path discovery process is illustrated in Fig. 5.

The proposed routing protocol operates in two main phases: path discovery and path maintenance. These phases involve the use of three control messages: Path Request (PREQ), Path Reply (PREP), and Path Error (PERR) messages.

Path discovery phase

Path discovery is a process to find and establish a communication path to a destination node. The source node broadcasts the PREQ packet to all its neighboring nodes. If the recipient node of the PREQ packet is the final destination, it sends a PREP packet back to the original sender. Before forwarding the packet, intermediate nodes store the broadcast ID and the node ID of the previous nodes. A timer is used by intermediate nodes to remove specific entries when no PREP is received for the last request. If there is a response, intermediate nodes again store the broadcast identifier and the ID of the source node. To avoid PREQ repetition, the broadcast

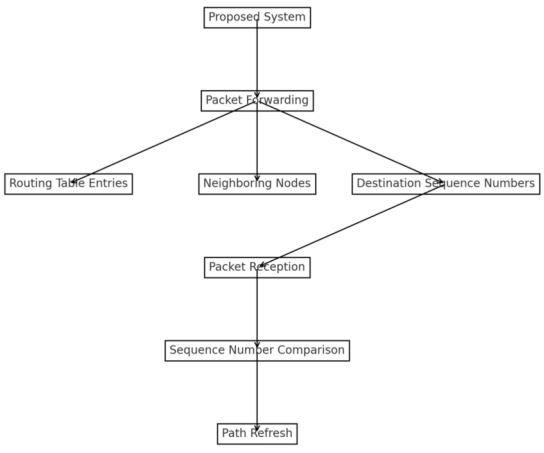

Fig. 4. Framework of EDSSR.

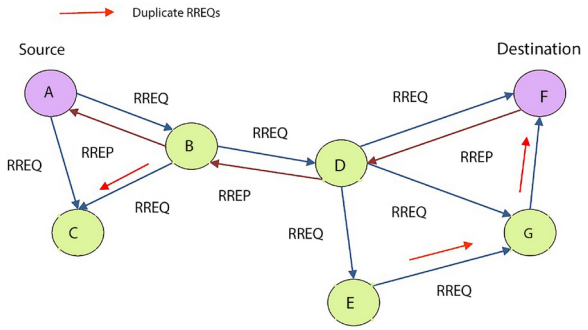

Fig. 5. Path discovery of proposed.

identifier and source ID are employed. If the source node receives more than one response, the decision to choose the option is based on the hop count.

Maintenance of path

Path maintenance is a strategy employed to address broken connections during transmission. Typically, the protocol monitors the failure of connections and neighboring links using "hai" message broadcasting. Once inaccessible nodes are identified, they are marked as invalid. Subsequently, the node nearest to the failed node generates a PERR message containing a list of the inaccessible nodes. The PERR message is then sent to the source node. Upon receiving the PERR message, the source node initiates the process of path re-establishment. The proposed protocol offers a significant advantage in reducing routing overload in large networks. Its unique feature involves the use of destination sequence numbers to keep routes up-to-date. However, the response to link breakdowns in the proposed protocol is relatively slow. Additionally, obtaining a path refers to the initial PREP.

Vulnerable parameters

To comprehend the vulnerabilities that give rise to the initiation of an PREQ Flooding attack during the path discovery phase, it is imperative to furnish a comprehensive description of all operations. The following points need to be outlined to pinpoint the ambiguities that empower attackers to launch the flooding attack:

- **Path discovery phase operations:**
  - **PREQ broadcasting process:** Elaborate on the sequence of steps involved in the broadcast of PREQ packets during the path discovery phase.
  - **Broadcast ID management:** Clarify the mechanisms governing the management and utilization of broadcast IDs during PREQ broadcasting.
  - **Storage of node IDs:** Provide an explanation of the procedures for storing and managing the node IDs of previous nodes during the forwarding of PREQ packets.
- **Destination node processing:**
  - **PREP generation process:** Elaborate on the steps involved in the destination node's processing of incoming PREQ packets and the subsequent generation of PREP packets.
  - **Transmission of PREP:** Provide an explanation of how the destination node initiates the transmission of PREP packets back to the original sender.
- **Intermediate node operations:**
  - **PREQ forwarding process:** Outline the mechanism by which intermediate nodes forward PREQ packets to their neighboring nodes.
  - **Handling of broadcast ID and node ID:** Elaborate on how intermediate nodes manage and handle broadcast IDs and node IDs during the forwarding of PREQ packets.
- **Repetition prevention mechanisms:**
  - **Utilization of broadcast ID and source ID:** Elaborate on the specific methods by which broadcast IDs and source IDs are utilized to prevent the redundancy of PREQ packets.
  - **Handling multiple responses:** Explain the decision-making process when the source node receives multiple responses and how hop count is considered.
- **Binary exponential back-off:** A binary exponential back-off is employed to alleviate network congestion caused by repeated attempts from the source node to establish a path to the destination. This mechanism determines the waiting time for the PREP, governing the timing of the next attempt for the PREQ data packet.

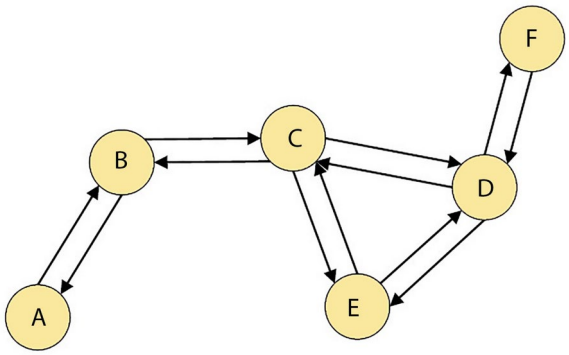

Fig. 6. Exchange of packets between L_N with TTL.

$Node_{IP}$ address	Link availability (L_{avail})	$Node_{stat}$
A	1	1
B	2	2

Table 2. L_N of node B in normal case.

- **Time-to-live (TTL):** The TTL value increments with each new attempt of the PREQ trail. During the expanding ring search process, the originating node adjusts the TTL start value (TTL_{Start}) provided in the PREQ packet IP header. Additionally, it adjusts the timeout value for receiving a PREP to the ring traversal time ($RING_{T_T}$) in milliseconds. If the PREQ times out without receiving a PREP, the source node rebroadcasts the PREQ with the TTL incremented (TTL_{Inc}). This process continues until the TTL in the PREQ reaches the TTL threshold (TTL_{Thre}). After reaching this threshold, TTL is set to the network diameter (NET_{Dia}) for each subsequent attempt.By providing a comprehensive description of these operations, the potential weaknesses and ambiguities in the path discovery phase that could be exploited for an PREQ Flooding attack can be identified and addressed. The proposed algorithm for detecting and preventing PREQ flooding attacks in WSNs follows a two-step process: Neighbor Validation and Threshold Validation. The EDSSR procedure is initiated when a node requests data transmission. If the path to the destination is not available, the source node sends an PREQ packet. Upon receiving the PREQ from any node, the node status is updated. The Neighbor Validation process is employed to verify the legitimacy of the source node. Following successful neighbor validation, the threshold validation is executed to prevent flooding attacks.

Neighbor validation

In the initial step of the proposed method, each node maintains a list of legal neighbors (L_N) through the regular exchange of "hai" packets, illustrated in Fig. 6. The purpose of L_N is to track every node generating PREQ packets, distinguishing between legal and malicious nodes. The structure of L_N is detailed in Table 2.

The main purpose of L_N is to store the IP address and status of neighboring nodes. When a node receives a PREQ packet, it verifies the source node by checking its IP address and status in its local list L_N. If the IP address is not found in L_N, the received PREQ is discarded. This mechanism helps prevent unauthorized nodes from entering the network. In EDSSR, messages, as defined in RFC 6553, are employed to determine the L_{avail} of each legal node's neighbor nodes. Local "hai" messages are used to update L_{avail} information, generated at intervals of $Packet_{Intval}$ msec. When a neighbor sends a "hai" message, the node ensures an active path to the neighbor is available.

Upon receiving a "hai" packet from a neighbor, the node logs the values of $Node_{IP}$ address and L_{avail}. A value of 1 in the L_{avail} field indicates an active route, while 0 indicates a lost route. The node status field contains three values derived from the EDSSR's second step: Normal, Suspicious, and Malicious, represented by the values 1, 2, and 3, respectively. For instance, in Fig. 6, nodes 2 and 3 are active neighbors around node 4. After the exchange of "hai" packets, node *B* records the IP addresses and L_{avail} of nodes *A* and *C* in its L_N. As depicted in Table 2, a value of 2 in the L_{avail} field indicates that node *B* has 2 valid neighbor nodes, *A* and *C*.

Validation against the threshold

Upon receiving an PREQ packet, a node examines the TTL field of the IP packet to check its value. If the TTL value is set to the maximum value, the node initiates an investigation to confirm the legality of the PREQ. The node searches its routing table for the same $PREQ_{ID}$ and IP Address to check the history of the TTL value. If a maliciously altered TTL value is found, the L_N table is updated with a status of 3 for the corresponding node; otherwise, the PREQ is forwarded to the next node. After detecting the malicious node, a message containing the malicious node's IP address is broadcast to all nodes in the network (Algorithm 1).

procedure DYNAMIC SECURE ROUTING
  Each node initializes L_N based on Table 2, containing the valid next-hop nodes.
  Validate neighbors according to Table 2.
  if Neighbor is valid **then** ▷ Next-hop authentication
    if Within $PREQ_{Rate_{Limit}}$ **then** ▷ Path threshold authentication
      if TTL value is valid **then**
        if Binary exponential back-off is valid **then**
          Forward PREQ.
        end if
      end if
    end if
  end if
  Mark node as malicious.
  Broadcast the ID of the malicious node to inform other nodes in the network.
end procedure

Algorithm 1. Pseudocode for EDSSR

Methodology for detecting and preventing sybil attacks in WSNs

A new concept of an Associate CH (ACH) is introduced in the proposed system to reduce EC and protect from Sybil attacks. In a cluster-based WSN, a normal flow of data with the presence of ACH is depicted in Fig. 7. The WSN consists of three clusters and a BS. Each cluster includes four sensor nodes (SN1, SN2, SN3, and SN4), (CH1, CH2, and CH3), and (ACH1, ACH2, and ACH3). In each cluster, sensor nodes, after sensing the data, send it to the corresponding ACH. The ACH, upon receiving data from sensor nodes, aggregates it and passes it to the corresponding CH. The BS receives all data from CHs and sends acknowledgments to both CHs and ACHs in the cluster.

In the proposed system, the role of the CH is solely to transmit all aggregated data to the BS. Unlike in peer-existing routing techniques, the responsibilities of the CH are shared with the ACH. Moreover, this optimization of responsibilities helps in optimizing the communication distance between the CH and SNs, ultimately contributing to the elongation of the lifetime of the CH (Fig. 8).

Under a Sybil attack, the data flow is depicted in Fig. 8. The sink node is not receiving any data from CH1, even though CH1 is receiving data from ACH1. If ACH1 does not receive an acknowledgment from the BS within a specified period, it checks the status of CH1 and informs the BS about CH1's status. The BS then needs to take further action.

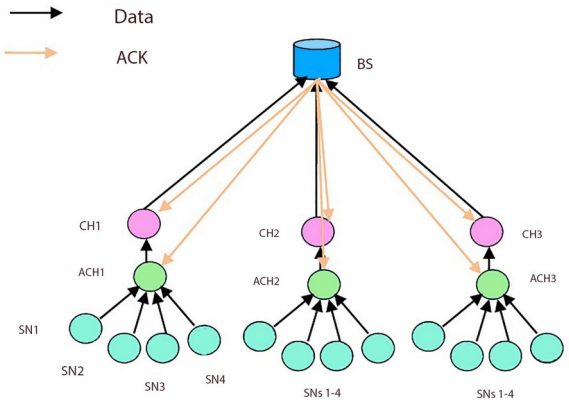

Fig. 7. Network without Sybil attack.

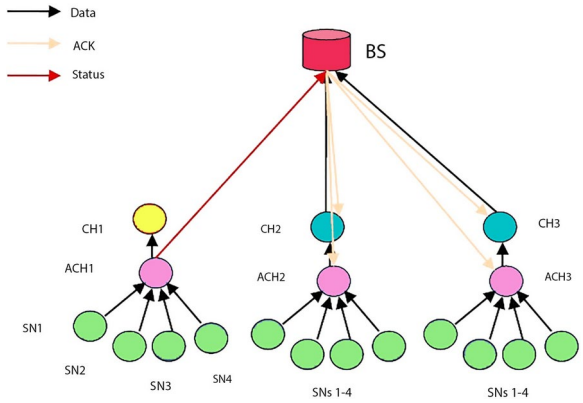

Fig. 8. Network under Sybil attack.

1: **Input:** Node ID, Location, Residual Energy ξ_r
2: **Output:** Cluster formation, CH and ACH selection
3: **procedure** EDSSR
4: **for** each node **do**
5: Send control packet (ID, Location, ξ_r) to BS
6: **end for**
7: BS receives status information from all nodes
8: BS divides network into required number of clusters
9: Calculate average node energy ξ_{avrg} of the cluster (refer to Eq. 1)
10: Create set μ for each cluster where $\xi_r > \xi_{avrg}$
11: Select CH from set μ based on minimum value for $Min_{Val_{CH}}$ (refer to Eq. 2)
12: Select ACH from set μ based on minimum intra-cluster communication cost (refer to Eq. 3)
13: BS broadcasts control packet containing information on clusters, CHs, and ACHs
14: **while** communication rounds continue **do**
15: Check energy level of CHs after each round
16: **if** CH energy level $\leq 7\%$ of initial energy **then**
17: Reelect CH
18: **end if**
19: **end while**
20: **end procedure**

Algorithm 2. EDSSR Procedure

The effectiveness of a Sybil attack is amplified by blocking a large amount of data, making CHs more susceptible to becoming Sybil nodes. This type of attack significantly impacts WSN performance, particularly in terms of E^2delay and throughput. Therefore, it is crucial to detect and prevent Sybil attacks to maintain WSN efficiency. EDSSR addresses this issue through an iterative procedure. The general procedure of EDSSR is outlined below (Algorithm 2):

- **Control packet transmission:** Each node sends a control packet to the BS that includes its ID, location, and residual energy, denoted as ξ_r.
- **Cluster formation:** The BS divides the network into the required number of clusters upon receiving the status information from all nodes.
- **Average node energy calculation:** The BS calculates the average node energy of the cluster, denoted as ξ_{avrg} , using the Eq. 1:

$$\xi_{avrg}(r) = \frac{1}{N}\sum_{i=1}^{N}\xi_i(r) \tag{1}$$

Here, N is the total number of nodes in the cluster, and $\xi_i(r)$ represents the residual energy of the ith node.

- **Set creation based on energy:** The BS creates a set of nodes, denoted as μ, for each cluster. This set includes nodes with residual energy greater than the average node energy, ξ_{avrg}.
- **CH selection:** The sensor node from set μ with the minimum value for the function $Min_{Val_{CH}}$ is selected as the CH. The function $Min_{Val_{CH}}$ is computed based on the distance of the node from the BS and its residual energy, using the Eq. 2:

$$Min_{Val_{CH}} = \frac{\vartheta_{(i,BS)}}{\xi_r} \tag{2}$$

Here, $\vartheta_{(i,BS)}$ represents the distance of the i^{th} node to the BS.

- **Assistant CH (ACH) selection:** The BS selects an ACH from set μ based on the criterion of minimum intra-cluster communication cost. The function $Min_{IntVal_{ACH}}$ is computed for each node as per Eq. 3:

$$Min_{IntVal_{ACH}} = \frac{\vartheta_{avrg} + \vartheta_{(i,CH)}}{\xi_r} \tag{3}$$

Here, ϑ_{avrg} is the average distance of all nodes to the node in μ, and $\vartheta_{(i,CH)}$ is the distance of the *i*th node in μ to the CH.

- **Control packet broadcasting:** After selecting the CH and ACH, the BS broadcasts a control packet to the network. This packet contains information about the clusters, CHs, and ACHs.
- **Energy level monitoring:** After each communication round, the BS checks the energy level of the CHs. If the energy level of a CH drops to 7% of its initial energy, the BS reelects the CH.

Data packets communication

The data transmission phase is segmented into three sub-phases: data collection, data aggregation, and data routing. In the data collection sub-phase, each sensor node transmits the sensed data to its respective ACH. During this phase, the ACH aggregates and compresses the received data once it has been collected from all the member sensor nodes. Following the data aggregation process, the ACH proceeds to the data routing sub-phase, transmitting the aggregated data to the corresponding CH and retaining a copy until it receives an acknowledgment from the BS. The CH, in turn, forwards the received data to the BS. Upon receiving the data, the BS generates an acknowledgment sent to both the corresponding CH and ACH.

Attack detection

The Sybil CH follows the process of receiving data from the ACH and discarding all data without forwarding it to the BS. After a specific duration of data transmission (denoted as *t*), if the ACH does not receive an acknowledgment, it communicates with the BS to inquire about the status of the data sent from the cluster. The BS checks the status of data reception, and if no data has been received, the BS informs the ACH about the status. The ACH then communicates with the CH using "hai" or random information to determine the status (alive or compromised) of the CH. If the CH responds to the ACH, it is considered malicious, and the ACH informs the BS about the malicious CH. The BS takes further action by replacing the CH through the process of reelecting a new CH for the corresponding cluster.

ACH facilitates as following:

- **Increased CH selection time:** The time taken for CH selection increases between two rounds due to the load-sharing mechanism between the CH and ACH.
- **No data loss during sybil attack:** In the presence of a Sybil attack, there is no data loss because the ACH stores all data until it receives an acknowledgment from the BS.

Results and discussion

Simulation environment

Qualnet 5.02 was utilized to conduct extensive simulations of the proposed EDSSR, with Table 3 outlining the parameter settings employed. In assessing the routing performance of WSNs, key metrics such as throughput, average E^2delay, EC, and network lifetime are considered pivotal. Additionally, parameters including network overhead, network load, and network energy consumption contribute to the comprehensive evaluation of network performance. The experimental results from the Proposed EDSSR protocol are compared against those of a peer-existing routing protocol, specifically the Enhanced Energy-Efficient Fuzzy-based Cognitive Radio (EEFCR)[47], with analysis focusing on the following metrics:

- **Average E^2 delay:** The packet E^2 delay, an average measure of the time a packet spends traversing the network, encompasses various delays incurred during its journey. These delays include transmission time delays

Parameter	Value
Simulator type	NS 2.34
Transmission time	100 *s*
Number of packets	10-bits
Sensor nodes	50–200
Flooding nodes	1–5
Network area	1500 *m*× 1500 *m*
Routing protocol	Routing Protocol for Low-Power and Lossy Networks (RPL-LLN)
Traffic	2 CBR applications
Packet size	1024-bits
MAC protocol	IEEE 802.15

Table 3. Simulation parameters used for test-bed.

arising from routing broadcasts, buffer queues, and other network processes. The average E^2 delay is computed as per Eq. 4

$$Avg.\ E^2\ delay = \frac{1}{n}\sum_{i=0}^{n}(\eta_r - \eta_s) \tag{4}$$

In this equation:
- n is the number of applications.
- i is the application ID, ranging from 0 to n.
- η_r is the time at which the first packet is received at the destination.
- η_s is the time at which the first packet is sent by the source.

The term $(\eta_r - \eta_s)$ represents the time difference between when a packet is sent and when it is received, providing the end-to-end delay for each packet. The average E^2 delay is the mean of these delays across all applications.

- **Throughput:** Throughput is defined as the total amount of data received by the destination per second. It is calculated using the Eq. 5:

$$Throughput = \frac{1}{n}\sum_{i=0}^{n}\frac{N \times 1024 \times 16}{\tau_T} \tag{5}$$

In this equation:
- n is the number of applications.
- N is the number of nodes in the network.
- τ_T is the total time over which the data is received.

The term $\frac{N\times 1024\times 16}{\tau_T}$ represents the data rate, where 1024 and 16 are factors to convert the node count into the appropriate data units received per second. The average throughput is the mean of these rates across all applications.

- **EC:** EC evaluation involves considering various parameters, such as the average energy consumption in each round or the remaining energy after each round. In this study, the assessment of energy consumption focuses on calculating the average residual energy of the network after each round. The EC can be expressed as per Eq. 6:

$$EC = \frac{1}{n}\sum_{i=1}^{n} E_{res}(i) \tag{6}$$

where
- n is the number of rounds.
- $E_{res}(i)$ is the residual energy of the network after the i-th round.

- **Network Lifetime (NL):** NL is defined as the duration until the battery power of all nodes is depleted. This metric is critical for understanding the sustainability and operational efficiency of the network over time. The NL can be expressed as per Eq. 7:

$$NL = \sum_{i=1}^{n} T_{round}(i) \tag{7}$$

where
- n is the number of rounds.
- $T_{round}(i)$ is the duration of the i-th round.

To ensure a more realistic simulation environment, the event-based simulator NS 2.34 is utilized to assess the impact of flooding attacks involving varying numbers of attackers. The number of nodes is systematically varied, ranging from 50 to 200, to observe the effects in both sparsely and densely populated environments. Various scenarios are generated with the simulation parameters outlined in Table 3. The simulation output is derived from multiple applications operating in the network. Specifically, 2 Constant Bit Rate (CBR) applications are employed for data transfer. To analyze the network's performance, the average output of the 2 applications is considered. The proposed technique's effectiveness is evaluated in terms of throughput and average E^2delay . The values of performance metrics in any given scenario are influenced by factors such as node movement speed, mobility direction, data flow in the network, and network congestion. Recognizing the complexity of these factors, performance metrics' average values are calculated over multiple simulation runs for a more accurate evaluation.

Analysis of throughput in various scenarios

- **Throughput vs. pause time without attack:** Figure 9 illustrates the performance analysis of EEFCR and EDSSR in terms of throughput as the pause time increases. The evaluation is conducted with 50, 100, and 200 nodes under normal conditions, without any security attack. In the absence of security threats, EEFCR demonstrates superior performance compared to other existing routing protocols. The comparison between EDSSR and EEFCR indicates that EDSSR outperforms EEFCR under normal network conditions.
- **Throughput Vs. node speed without attack:** Figure 10 presents the analysis of throughput against node speed for EEFCR and EDSSR in a scenario without any attack. Both routing protocols exhibit nearly identical performance with varying node speed and density. The throughput of the network tends to decrease as the speed of nodes increases due to an increased occurrence of link failures in the network. As depicted in Fig.

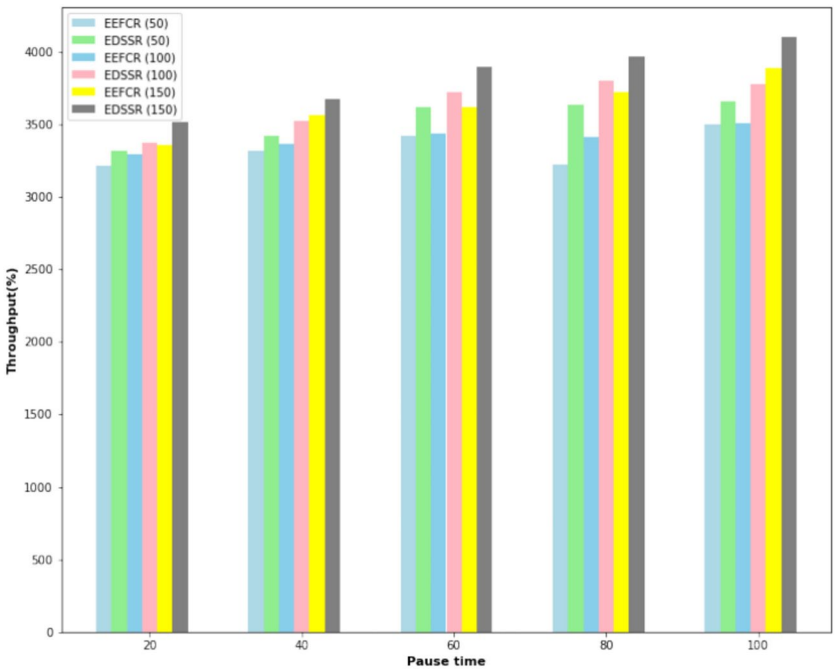

Fig. 9. Comparison of throughput vs. pause time.

10, throughput experiences a continuous decrease with rising node speed up to 40, after which it stabilizes with minor fluctuations in both protocols. Notably, even in normal conditions, EDSSR demonstrates superior performance compared to EEFCR.

Analysis of average E^2 in various scenarios

- **Average E^2 delay vs. pause time without attack:** The performance of EEFCR and EDSSR is evaluated in terms of average E^2delay, considering an increase in pause time with 50, 100, and 200 nodes when no attacks are present. Across various scenarios, the proposed EDSSR consistently competes with EEFCR, demonstrating comparable or better average E^2delay performance.

  In Fig. 11, the influence of an escalating number of attacking nodes on the average E^2delay is evident for both EEFCR and EDSSR. As the number of attacking nodes rises, the average E^2delay increases in both protocols. However, EDSSR demonstrates a more efficient response to the presence of attacking nodes compared to EEFCR. The delay in EEFCR is aggravated by the flooding of fake PREQ packets generated by malicious nodes, hindering genuine data transmission. In contrast, EDSSR identifies malicious nodes, mitigates fake PREQ transmissions, and facilitates normal data transfer, resulting in a more controlled increase in delay primarily due to the rise in normal traffic with an expanding number of nodes.

- **Average E^2 vs. node speed without attack:** In Fig. 12, a comparison of the average E^2delay against node speed is conducted between EEFCR and EDSSR. The evaluation is performed with increasing node density, considering scenarios with 50, 100, and 200 nodes. The results of the experiment illustrate the performance of both protocols under varying node speeds.

  The observation reveals that as the node speed increases, the average E^2delay also increases in both EEFCR and EDSSR. The higher node speed leads to more frequent link breakages and re-connections, contributing to increased delays in data communication. Interestingly, in all three node densities (50, 100, and 200 nodes), EEFCR and EDSSR demonstrate better performance at 100 nodes compared to scenarios with 50 and 200 nodes.

Energy efficiency evaluation

The assessment of energy efficiency involves a comparison between EEFCR and EDSSR in terms of the remaining energy of sensor nodes over multiple rounds.

Figure 13 illustrates the impact of a black hole on the remaining energy of nodes over multiple rounds. The network under a black hole attack exhibits lower energy consumption compared to EDSSR because the black hole CH drops packets received from ACH, resulting in no communication between CH and BS. The graph in Fig. 13 indicates that a certain energy level remains almost constant in the last few rounds because the energy of the black hole node gradually decreases. The performance of EDSSR is justified for the following reasons:

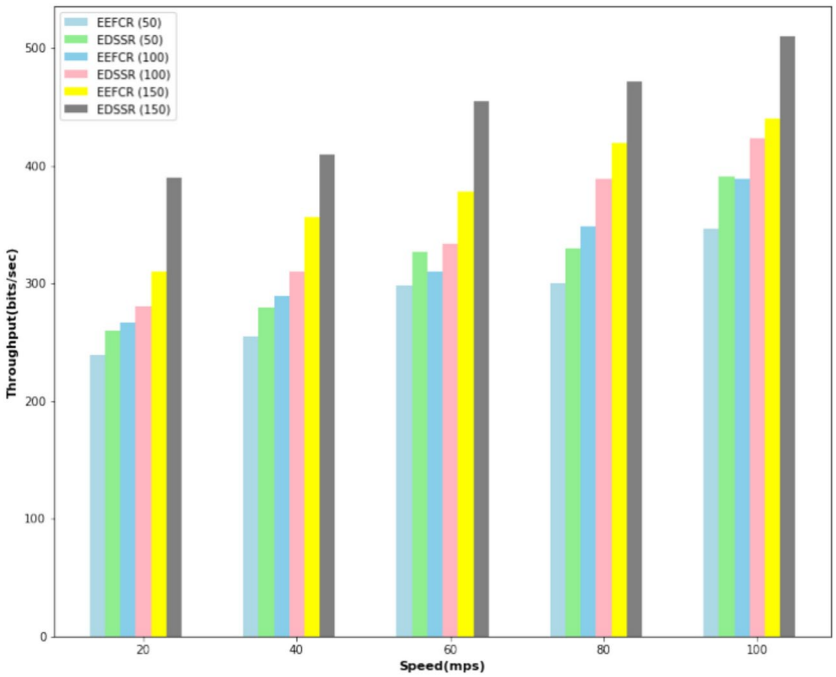

Fig. 10. Comparison of throughput vs. speed.

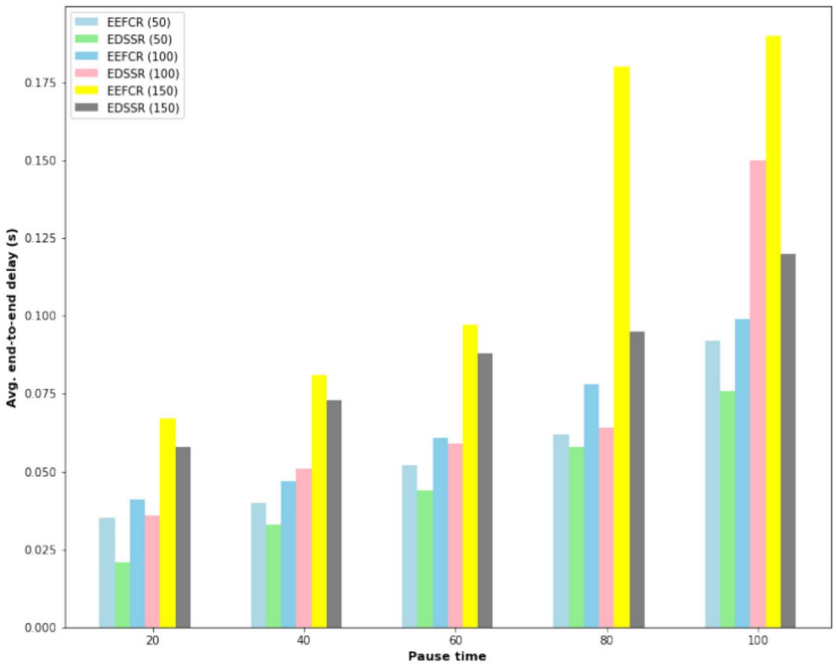

Fig. 11. Comparison of average E^2delay vs. pause time.

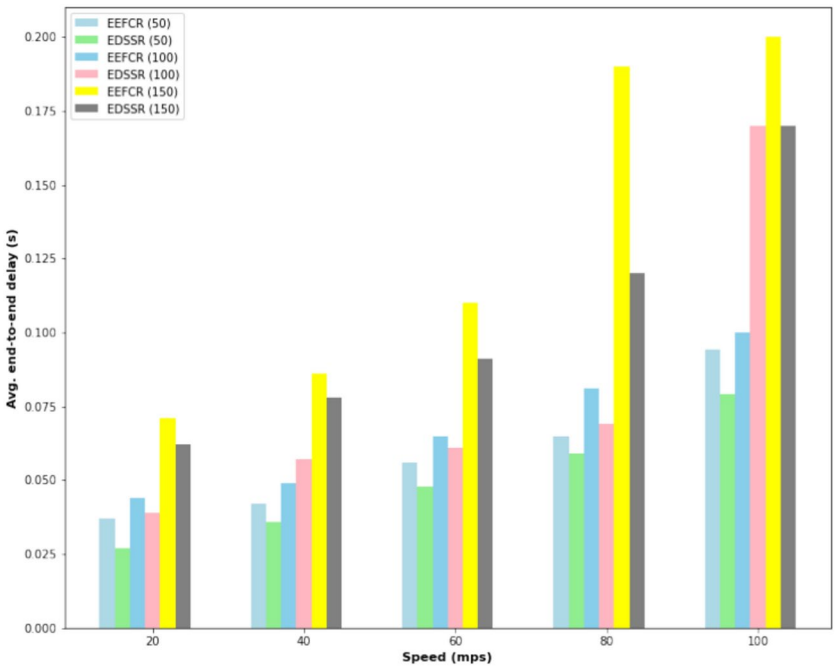

Fig. 12. Comparison of average E^2delay vs. speed.

The utilization of ACH extends the time between the reelection of CH, as ACH is responsible for intra-cluster communications, reducing the energy consumption of CH. CH is only responsible for inter-cluster communication, leading to lower energy consumption. CH is reelected before its energy drops to 10%, and it starts functioning as a normal node.

NL

NL is described in terms of the number of alive nodes (NL) in the network. The more nodes will be alive till the last round, the higher will be the NL. NL is higher in EEFCR and EDSSR.

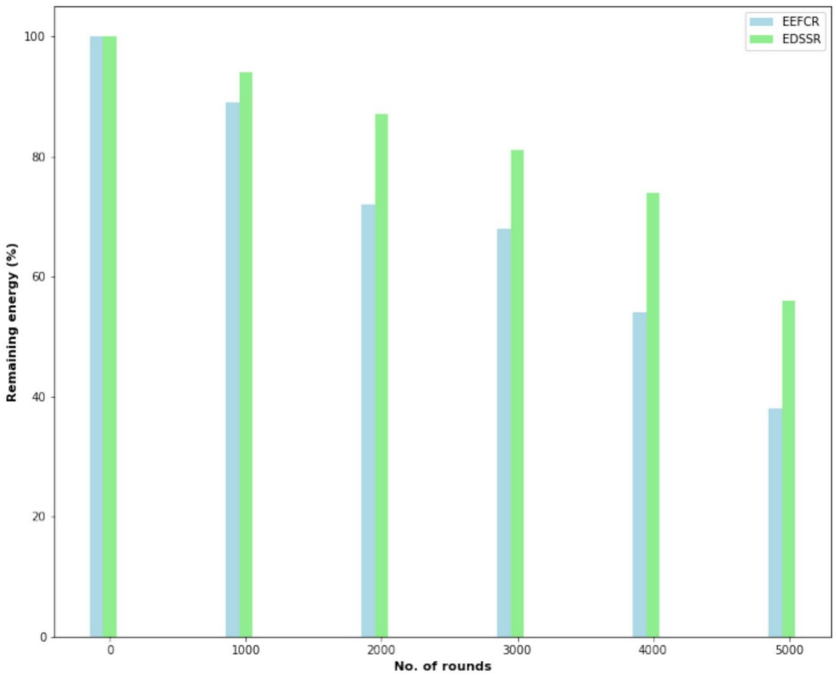

Fig. 13. Comparison of remaining energy under Sybil attack.

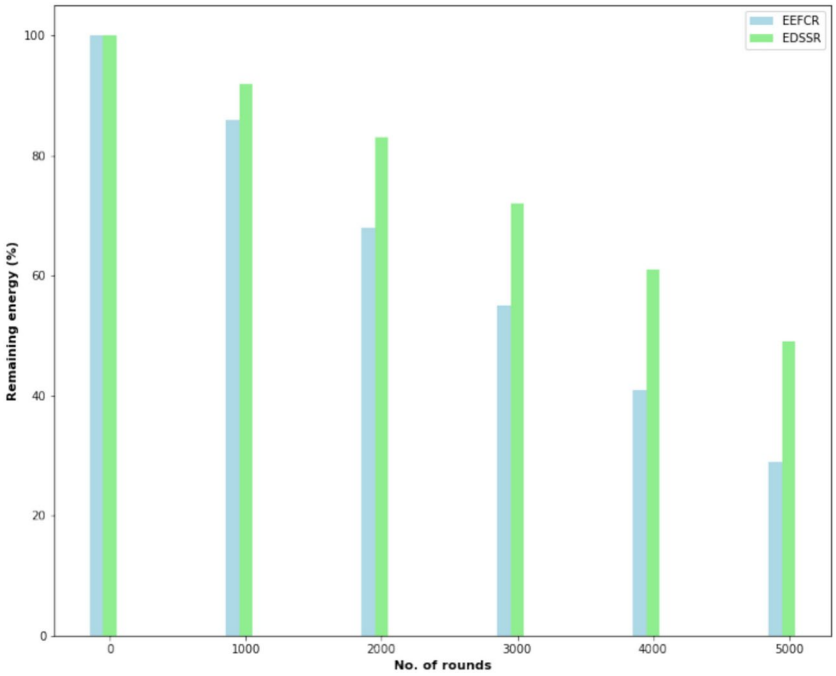

Fig. 14. Comparison of NL under Sybil attack.

In Fig. 14, the network lifetime is depicted. In the case of a black hole attack, the network lifetime is increased compared to the scenario without a black hole. Most of the energy in the cluster is typically consumed in communication between CH and BS. However, under a black hole attack, the black hole CH does not communicate with BS, resulting in significant energy savings. The battery consumption of the black hole nodes is very low compared to other CHs, leading to a substantial increase in network lifetime. In clusters, CH in C1, which acts as a black hole, consumes less energy compared to CHs in C2 and C3, which act as normal nodes. Figure 15 illustrates the performance comparison of the Proposed EDSSR, DLAMD, and EEFCR detection methods based

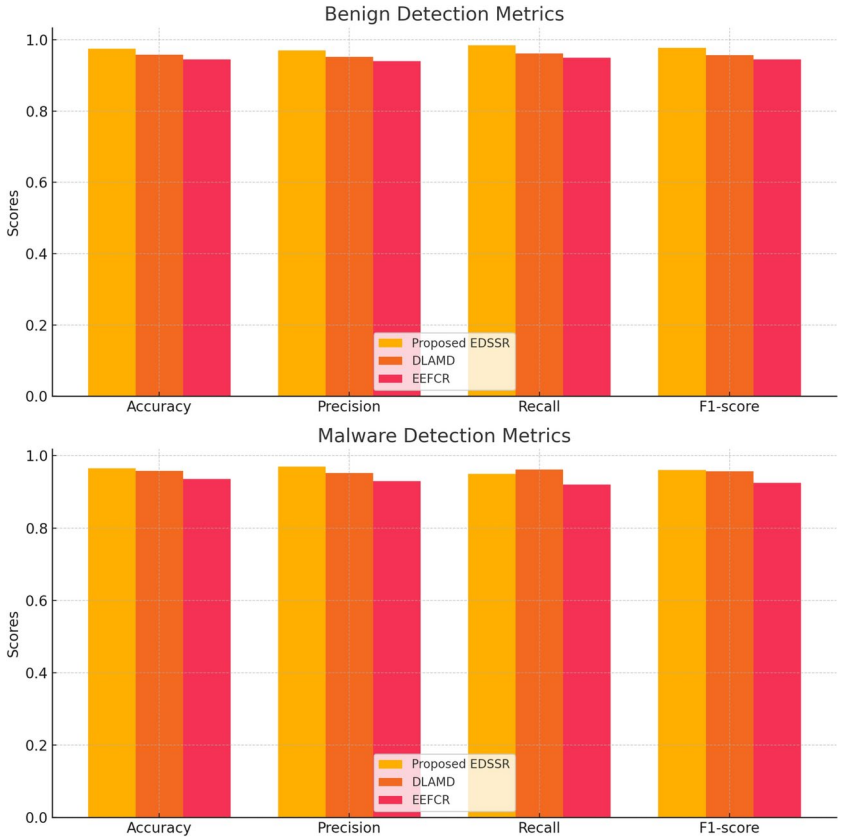

Fig. 15. Comparison of accuracy, precision, recall, and F1-score.

Methods	Benign/Malware	Accuracy	Precision	Recall	F1-score
Proposed EDSSR	Benign	0.9750	0.9700	0.9850	0.9770
	Malware	0.9650	0.9700	0.9500	0.9600
DLAMD	Benign	0.9583	0.9524	0.9615	0.9569
	Malware	0.9583	0.9524	0.9615	0.9569
EEFCR	Benign	0.9450	0.9400	0.9500	0.9450
	Malware	0.9350	0.9300	0.9200	0.9250

Table 4. Comparison of different detection methods.

on four key metrics: Accuracy, Precision, Recall, and F1-score. The results are presented separately for benign and malware detection.

Table 4 evaluates the performance of three detection methods-proposed EDSSR, DLAMD, and EEFCR-in terms of Accuracy, Precision, Recall, and F1-score for detecting benign and malware cases. The proposed EDSSR method showcases superior effectiveness compared to DLAMD and EEFCR.

For benign detection, the proposed EDSSR achieves an outstanding accuracy of 97.5%, with a precision of 97.0%, a recall of 98.5%, and an F1-score of 97.7%. This indicates that the proposed method is highly accurate in identifying benign cases and can detect them consistently and correctly. For malware detection, it maintains strong performance with an accuracy of 96.5%, precision of 97.0%, recall of 95.0%, and an F1-score of 96.0%. These results demonstrate the method's robustness and reliability in identifying malware accurately and effectively.

In comparison, DLAMD also shows strong results with high consistency across both benign and malware detection. It achieves an accuracy of 95.83%, precision of 95.24%, recall of 96.15%, and an F1-score of 95.69% for both detection types. This consistency highlights DLAMD's ability to reliably handle both benign and malware cases, though it slightly lags behind the proposed EDSSR in overall performance.

EEFCR, while effective, generally trails behind the other two methods. It achieves an accuracy of 94.5%, precision of 94.0%, recall of 95.0%, and an F1-score of 94.5% for benign detection, and an accuracy of 93.5%, precision of 93.0%, recall of 92.0%, and an F1-score of 92.5% for malware detection. These results indicate that while EEFCR is useful, it is less effective compared to the other two methods.

The superior performance of the proposed EDSSR can be attributed to several key factors. Firstly, the method's unique architecture focuses on securing and optimizing energy efficiency in WSNs, which helps in maintaining accurate and up-to-date routing information without storing multi-hop routes, thereby reducing memory usage and simplifying the protocol. Secondly, the use of destination sequence numbers ensures that the nodes always have the latest path information, which improves the reliability and efficiency of packet delivery. Thirdly, the dynamic routing and adaptive decision-making capabilities allow the proposed EDSSR to respond effectively to changing network conditions, ensuring optimal performance in various scenarios.

Conclusion and future scope

The research emphasizes the critical need for securing WSNs against various security threats. It provides a comprehensive overview of security issues in WSNs, highlighting the vulnerabilities and limitations of existing techniques. Recognizing the significance of security in optimizing WSN performance, the proposed EDSSR protocol aims to address the specific challenge of flooding attacks, particularly Sybil attacks. While acknowledging the performance merits of the EEFCR routing protocol in normal conditions, the study underscores its vulnerability to security breaches. In response, the EDSSR protocol is introduced as an enhancement to EEFCR, incorporating a security mechanism to fortify WSNs against flooding attacks. The flooding attack is meticulously addressed during the path discovery phase of EEFCR, where attackers exploit various protocol parameters. EDSSR employs a two-step approach to identify malicious nodes, validating the legality of neighboring nodes and comparing parameter values against standard specifications outlined in RFC 6553. The use of a structured L_N facilitates the storage of legal neighbors, aiding in status assignment post-inspection. Despite the promising results, this study has certain limitations. The EDSSR protocol, while effective against specific flooding attacks, has not been tested extensively against other types of security threats, such as "hai" flooding and data flooding. Additionally, the protocol's performance metrics, such as routing overhead, packet drop rate, and a detailed analysis of energy efficiency, require further exploration. These limitations indicate that while the current work provides a solid foundation, there is room for significant improvements and a need for broader evaluations. Looking ahead, future research may explore additional facets of flooding attacks, such as "hai" flooding and data flooding, to devise a comprehensive security system capable of safeguarding against all types of flooding attacks. Furthermore, the evaluation of designed algorithms could extend to include parameters like routing overhead, packet drop rate, and a more nuanced exploration of energy efficiency, contributing to the development of robust and resilient WSN security mechanisms.

Data availability

The data that support the findings of this study are available from Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia; but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are however available from the author [Premkumar Chithaluru] upon reasonable request and with permission of Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Received: 25 April 2024; Accepted: 25 October 2024
Published online: 19 November 2024

References

1. Meenakshi, B. & Karunkuzhali, D. Enhancing cyber security in WSN using optimized self-attention-based provisional variational auto-encoder generative adversarial network. *Comput. Standards Interfaces* **88**, 103802 (2023).
2. Woungang, I. & Dhurandher, S.K. eds., 2019. 2nd International Conference on Wireless Intelligent and Distributed Environment for Communication: WIDECOM 2019 (Vol. 27). Springer.
3. Bao, Y., Li, Y., Zhao, L., Zhang, A. & Wang, Y. MCEN: Maximum cooperative equilibrium WSN based on greedy prediction to reduce opposite transmission. *Comput. Netw.* **221**, 109506 (2023).
4. Suresh, B. & Prasad, G. S. C. An energy efficient secure routing scheme using LEACH protocol in WSN for IoT networks. *Measurement: Sens.* **30**, 100883 (2023).
5. Pandiyaraju, V., Ganapathy, S., Mohith, N. & Kannan, A. An optimal energy utilization model for precision agriculture in WSNs using multi-objective clustering and deep learning. *J. King Saud Univ.-Comput. Inf. Sci.* **3**(10), 101803 (2023).
6. Uthayakumar, G. S. et al. Systematically efficiency enabled energy usage method for an IOT based WSN environment. *Measurement: Sens.* **25**, 100615 (2023).
7. Babu, V., Kumar, C. V., Parthiban, S., Padmavathi, U. & Rahman, M. Z. U. AE-LEACH: An incremental clustering approach for reducing the energy consumption in WSN. *Microprocess. Microsyst.* **93**, 104602 (2022).
8. Gurram, G. V., Shariff, N. C. & Biradar, R. L. A secure energy aware meta-heuristic routing protocol (SEAMHR) for sustainable IoT-wireless sensor network (WSN). *Theoret. Comput. Sci.* **930**, 63–76 (2022).
9. Anitha, S., Saravanan, S. & Chandrasekar, A. Trust management based multidimensional secure cluster with RSA cryptography algorithm in WSN for secure data transmission. *Measurement: Sens.* **29**, 100889 (2023).
10. Rani, S. S. & Sankar, K. S. Improved buffalo optimized deep feed forward neural learning based multipath routing for energy efficient data aggregation in WSN. *Measurement: Sens.* **27**, 100662 (2023).
11. Revanesh, M., Acken, J. M. & Sridhar, V. DAG block: Trust aware load balanced routing and lightweight authentication encryption in WSN. *Futur. Gener. Comput. Syst.* **140**, 402–421 (2023).
12. Patil, V. B. & Kohle, S. A high-scalability and low-latency cluster-based routing protocol in time-sensitive WSNs using genetic algorithm. *Measurement: Sens.* **31**, 100941 (2023).
13. Srividya, P. & Devi, L. N. An optimal cluster & trusted path for routing formation and classification of intrusion using machine learning classification approach in WSN. *Global Transit. Proc.* **3**(1), 317–325 (2022).
14. Salim, A., Osamy, W., Aziz, A. & Khedr, A. M. SEEDGT: Secure and energy efficient data gathering technique for IoT applications based WSNs. *J. Netw. Comput. Appl.* **202**, 103353 (2022).
15. Radhika, M. & Sivakumar, P. Video Traffic Analysis over LEACH-GA routing protocol in a WSN. *Procedia Comput. Sci.* **165**, 701–707 (2019).

16. Zhao, X. et al. A detection probability guaranteed energy-efficient scheduling mechanism in large-scale WSN. *Alex. Eng. J.* **71**, 451–462 (2023).
17. Vinitha, A. & Rukmini, M. S. S. Secure and energy aware multi-hop routing protocol in WSN using Taylor-based hybrid optimization algorithm. *J. King Saud Univ.-Comput. Inf. Sci.* **34**(5), 1857–1868 (2019).
18. Alghamdi, W. Y. Designing a secure and long-lived WSN for data collection. *Procedia Comput. Sci.* **220**, 187–194 (2023).
19. Esmaeili, H., Hakami, V., Bidgoli, B. M. & Shokouhifar, M. Application-specific clustering in wireless sensor networks using combined fuzzy firefly algorithm and random forest. *Expert Syst. Appl.* **210**, 118365 (2022).
20. Dinesh, K. & Santhosh Kumar, S. V. N. Energy-efficient trust-aware secured neuro-fuzzy clustering with sparrow search optimization in wireless sensor network. *Int. J. Inf. Secur.* **23**(1), 199–223 (2024).
21. Thangaramya, K. et al. Intelligent fuzzy rule-based approach with outlier detection for secured routing in WSN. *Soft. Comput.* **24**, 16483–16497 (2020).
22. Selvi, M. et al. An energy efficient clustered gravitational and fuzzy based routing algorithm in WSNs. *Wireless Pers. Commun.* **116**, 61–90 (2021).
23. Santhosh Kumar, S. V. N. et al. Energy efficient secured K means based unequal fuzzy clustering algorithm for efficient reprogramming in wireless sensor networks. *Wireless Netw.* **27**, 3873–3894 (2021).
24. Mehmood, G., Khan, M. Z., Bashir, A. K., Al-Otaibi, Y. D. & Khan, S. An efficient QoS-based multi-path routing scheme for smart healthcare monitoring in wireless body area networks. *Comput. Electr. Eng.* **109**, 108517 (2023).
25. Fanian, F. & Rafsanjani, M. K. CFMCRS: Calibration fuzzy-metaheuristic clustering routing scheme simultaneous in on-demand WRSNs for sustainable smart city. *Expert Syst. Appl.* **211**, 118619 (2023).
26. Malik, A., Khan, M. Z., Qaisar, S. M., Faisal, M. & Mehmood, G. An efficient approach for the detection and prevention of gray-hole attacks in VANETs. *IEEE Access* **11**, 46691–46706 (2023).
27. Shanmugapriya, R. & Santhosh Kumar, S. V. N. An energy efficient Swan Intelligent based Clustering Technique (SICT) with fuzzy based secure routing protocol in IoT. *Peer-to-Peer Netw. Appl.* **17**(4), 1830–1864 (2024).
28. Kavitha, V. & Ganapathy, K. Galactic swarm optimized convolute network and cluster head elected energy-efficient routing protocol in WSN. *Sustain. Energy Technol. Assess.* **52**, 102154 (2022).
29. Ramteke, R., Singh, S. & Malik, A. Optimized routing technique for IoT-enabled software-defined heterogeneous WSNs using genetic mutation-based PSO. *Comput. Standards Interfaces* **79**, 103548 (2022).
30. Chithaluru, P., Tiwari, R. & Kumar, K. AREOR-Adaptive ranking-based energy-efficient opportunistic routing scheme in Wireless Sensor Network. *Comput. Netw.* **162**, 106863 (2019).
31. Dinesh, K. & Svn, S. K. GWO-SMSLO: Grey wolf optimization based clustering with secured modified Sea Lion optimization routing algorithm in wireless sensor networks. *Peer-to-Peer Netw. Appl.* **17**(2), 585–611 (2024).
32. Santhosh Kumar, S. V. N. & Palanichamy, Y. Energy efficient and secured distributed data dissemination using hop by hop authentication in WSN. *Wireless Netw.* **24**, 1343–1360 (2018).
33. Asha, A., Arunachalam, R., Poonguzhali, I., Urooj, S. & Alelyani, S. Optimized RNN-based performance prediction of IoT and WSN-oriented smart city application using improved honey badger algorithm. *Measurement* **210**, 112505 (2023).
34. Hilal, A. M. et al. Trust aware oppositional sine cosine based multihop routing protocol for improving survivability of wireless sensor network. *Comput. Netw.* **213**, 109119 (2022).
35. Chithaluru, P., Al-Turjman, F., Kumar, M. & Stephan, T. I-AREOR: An energy-balanced clustering protocol for implementing green IoT in smart cities. *Sustain. Cities Soc.* **61**, 102254 (2020).
36. Jayashree, S. & Kumar, S. S. An efficient group signature based certificate less verification scheme for vehicular ad-hoc network. *Wirel. Netw.* **30**, 3269–3298 (2024).
37. Khan, T. et al. An efficient trust-based decision-making approach for WSNs: Machine learning oriented approach. *Comput. Commun.* **209**, 217–229 (2023).
38. Mansour, R. F. et al. Energy aware fault tolerant clustering with routing protocol for improved survivability in wireless sensor networks. *Comput. Netw.* **212**, 109049 (2022).
39. Li, C., Wu, J., Zhang, Z. & Lv, A. Energy-harvesting Q-learning secure routing algorithm with authenticated-encryption for WSN. ICT Express (2023).
40. Lu, N. et al. An efficient combined deep neural network based malware detection framework in 5G environment. *Comput. Netw.* **189**, 107932 (2021).
41. Kumar, M. P. & Hariharan, R. Improved trustworthy, speed, and energy-efficient GPSR routing algorithm in large-scale WSN. *Measurement Sens.* **24**, 100576 (2022).
42. Chithaluru, P., Tiwari, R. & Kumar, K. ARIOR: Adaptive ranking based on improved opportunistic routing in wireless sensor networks. *Wireless Pers. Commun.* **116**(1), 153–176 (2021).
43. Chithaluru, P. K., Khan, M. S., Kumar, M. & Stephan, T. ETH-LEACH: An energy-enhanced threshold routing protocol for WSNs. *Int. J. Commun Syst* **34**(12), e4881 (2021).
44. Rajkumar, Y. & Kumar, S. S. An elliptic curve cryptography based certificate-less signature aggregation scheme for efficient authentication in vehicular ad hoc networks. *Wireless Netw.* **30**(1), 335–362 (2024).
45. Chithaluru, P., Kumar, S., Singh, A., Benslimane, A. & Jangir, S. K. An energy-efficient routing scheduling based on fuzzy ranking scheme for internet of things. *IEEE Internet Things J.* **9**(10), 7251–7260 (2021).
46. Chithaluru, P., Al-Turjman, F., Kumar, M. & Stephan, T. MTCEE-LLN: Multilayer threshold cluster-based energy-efficient low-power and lossy networks for industrial internet of things. *IEEE Internet Things J.* **9**(7), 4940–4948 (2021).
47. Liu, X., Yu, J., Yu, K., Wang, G. & Feng, X. Trust secure data aggregation in WSN-based IIoT with single mobile sink. *Ad Hoc Netw.* **136**, 102956 (2022).

Acknowledgements

The authors would like to acknowledge the support of Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R435), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author contributions

Conceptualization: R.Y., M.A.P., P.N., D.J., K.K., P.C.; Methodology: R.Y., M.A.P., N.S.Y.; Formal analysis and data curation: P.N., D.J., N.S.Y.; Writing—original draft preparation: R.Y., M.A.P.; writing—review and editing: R.Y., M.A.P., P.C.; Supervision: S.K., D.S.A.E., D.M.A.; All authors have read and agreed to the published version of the manuscript.

Funding

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R435), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Declarations

Competing interests
The authors declare no conflict of interest.

Additional information
Correspondence and requests for materials should be addressed to P.C. or D.M.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.