Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 8s (2025)

A Cross-Layer-Routing Protocol with Malicious node detection using BiRNN in WSN

[1]S.Saritha, [2]Dr.E.Sreenivasa Reddy
[1]Research Scholar, [2]Professor
[1]Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India
[2]School of Computer Science and Engineering, VIT-AP University, Amaravathi, Andhra Pradesh, India
Email: [1]sarithanune@gmail.com, [2]sreenivasareddy.e@vitap.ac.in

Article History:
Received: *08-11-2024*
Revised:*23-12-2024*
Accepted:*08-01-2025*

Abstract:
A wireless sensor network (WSN) is a standalone device that consists of a discrete group of sensor nodes (SN) for information gathering, device monitoring, and environmental position sensing. The main challenge is to present an energy-efficient framework and conserve energy while building a route path alongside each sensor node at SN due to the limited energy resources available. Nonetheless, a great deal of energy-efficient methods concentrated heavily on energy harvesting and decreased energy usage, but they were unable to enable energy-efficient routing in WSNs with low energy usage. The research work uses Kinetic Gas Molecules Optimization (KGMO) based Modified Reptile Search Algorithm for cluster head selection to minimise the energy consumption in order to address this issue. Utilising a modified version of the Reptile search algorithm, the inertia weight of the KGMO is determined by evaluating trust (direct and indirect) based on energy, distance, probability of risk, delay, and Received Signal Strength Indicator (RSSI), with the optimal CH being chosen. Lastly, to find the shortest path, use an energy-efficient cross-layer-based expedient routing protocol (E-CERP), which lowers network overhead dynamically. Assuring security for the massive amounts of data being generated by sensors is the most difficult task. It must balance the trade-offs with a number of other factors, including power consumption, delay, latency, and data aggregation, in order to ensure security and make room for different types of research. The work's concept is to use a Bidirectional Recurrent Neural Network (BiRNN) model to identify the malicious nodes. When compared to previous methods, the proposed model demonstrated exceptional results by achieving a Packet Delivery Ratio (PDR) of 0.99, Throughput of 0.98, Packet Delay reduced to 0.05, Power Consumption minimized to 1.52, and an outstanding Accuracy of 99.21%. These findings underscore its efficacy in enhancing both network performance.
Keywords: Wireless Sensor Network, Reptile Search Algorithm, Energy-Efficient Cross-Layer-based Expedient Routing Protocol, Clustering Head, Kinetic Gas Molecules Optimization.

1. Introduction

Small, reasonably priced sensor nodes make up a wireless sensor network (WSN). It has been demonstrated that WSNs are among the best methods for moving data from remote locations to a central data processing centre. By self-organizing, these sensors create a multihop network that can adapt and data compression and send it to a base station [1]. In addition to collecting more data for applications like Multimedia WSNs, inexpensive healthcare, intelligent buildings, military surveillance, agricultural and industrial monitoring, they also facilitate the transmission of multimedia data, including images and videos. More resources may be needed by larger sensor nodes, and multimedia data is usually larger. This problem has been studied and attempted to be resolved by some researchers [2]. One of the most important ways to increase energy conservation extends the lifespan of sensor nodes in wireless sensor networks. Most energy is used in packet sending and receiving [3].

In WSNs, sensor nodes frequently use batteries. The complexity of battery charging stems from the network of devices, and the battery's capacity becomes the most crucial resource for WSNs [4]. Energy

conservation therefore becomes crucial for WSNs. It is necessary to create a new optimisation algorithm in order to maximise network lifespan and energy efficiency. One of WSNs' power management features is clustering, which divides the network into several clusters, each of which has one node, referred to as the cluster head (CH) [5]. The base station (BS)'s overhead is decreased by the CH by merging the information it gets from every node and sending it to the BS. In scenarios where resources are limited, the WSN conserves energy since less nodes send data to the BS. In WSNs, clustering algorithms help cut down on power consumption [6]. The formation and stabilisation phases make up each round of the clustering algorithm's operation. The nodes are arranged into discrete groups, or clusters. Every group is assigned a CH [7]. The CH gathers data from sensor nodes and transmits the sensed data to the recipient. Moreover, while there are other factors that contribute to clustering, the reduction of energy consumption has been examined from that angle in some survey reports [8]. Most publications that cluster data contrast and compare the efficacy of different clustering techniques, but they hardly ever examine the goals of the strategies. As far as we are aware, no survey study has looked at the characteristics of WSN networks, like heterogeneity and mobility, that are made possible by current clustering algorithms [9].

As a result, cluster-oriented techniques also contributed to the network's ability to expand its lifespan. The techniques that are most frequently used are Low Energy Adaptive Clustering Hierarchy (LEACH) and Fuzzy C-Means (FCM). Additionally, based on the calculated likelihood, the cluster-oriented LEACH approach operates in a dispersed manner and favours CH. As a result, increasing the WSN's energy efficiency is crucial since it shortens the network's lifespan. Using clustering to reduce transmission energy consumption and increase network lifetime is a successful strategy [10]. Clustering prevents packet collisions, increases throughput, and expands network scalability. Moreover, clustering reduces total energy usage. Up until now, several meta-heuristic scheme-based central cluster-oriented techniques have been presented. Particle Swarm Optimisation (PSO), the Harmony Search Algorithm (HSA), and other schemes are examples of specific, broad schemes. Further difficult aspects of modelling the routing schemes are the network's EE, QoS, and lifespan. The main drawback of using a mobile sink is information loss, which is why they are used in the conventional model in spite of their low energy consumption and high throughput [11]. When data packets are transferred using the suggested architecture from the destination node from the source node, information loss is prevented. Utilising the static node also lowers energy consumption. When paired with a mobile sink, a cluster head's mobility results in a higher transmission loss of data. When used in conjunction with a static sink, the cluster head's immobility reduces the likelihood of data loss [12].

Motivation:

In Wireless Sensor Networks (WSNs), locating malicious nodes is critical to maintaining network security, dependability, and data integrity. Through the identification and isolation of these rogue elements, the network can continue to function, guaranteeing dependable operation and accurate data transmission. Malicious node detection protects against denial-of-service attacks, unauthorised access, and data manipulation while fostering greater trust among network users. Proactive steps not only reduce possible risks but also make WSNs more durable and effective. This promotes innovation and progress in a number of industries that depend on WSNs, including industrial automation, environmental monitoring, and healthcare systems.

Main Contributions:

- Introduction of a KGMO-based Modified Reptile Search Algorithm for efficient cluster head selection in WSNs, prioritizing factors like distance, energy, security, delay, trust evaluation, and RSSI.

- Development of an energy-efficient cross-layer-based routing protocol (E-CERP) to dynamically minimize network overhead by determining the shortest route.
- Proposal of an BiRNN model to detect malicious nodes, addressing security concerns without compromising on factors like power consumption, delay, latency, and data aggregation.
- Evaluation of the proposed method's performance through metrics including error estimation, network lifetime, packet loss rate, packet delivery ratio (PDR), packet delay, throughput, power consumption, and packet loss rate, demonstrating superior results compared to existing methods.

Organization of the Paper:

The following is the order of the residual sections: Related works are exhibited in Section 2, dataset details are provided in Section 3, the suggested approach is presented in Section 4, the experimental analysis is presented in Section 5, and a conclusion is reached in Section 6.

2. Literature Survey

In the work by Nouman, M., et al. [13], In order to address several security concerns and enable nodes to be registered using login credentials, blockchain technology was deployed on base stations (BSs) and cluster heads (CHs). To further categorise the nodes as malicious or legitimate, a machine learning (ML) classifier known as the histogram gradient boost (HGB) was applied to the BSs. The node's registration was removed from the network in the event that it was discovered to be malicious. On the other hand, the data of a node that was determined to be authentic was kept in an Interplanetary File System (IPFS). After creating IPFS stored the data in chunks and used it as a hash that were subsequently recorded in the blockchain. Verifiable Byzantine Fault Tolerance (VBFT) was also employed to carry out consensus and validate transactions in place of Proof of Work (PoW). Furthermore, extensive simulations were run with the WSN-DS (Wireless Sensor Network) dataset. The balanced dataset as well as the original dataset were used to evaluate the model.

Moundounga, A. R. A., et al. [14] intended to create an anomaly identification method that would increase the accuracy and security of the sensor network. In order to achieve this, a detection system that recognised network malicious entries by learning from routing datasets was defined using machine learning techniques. The models relied on the stochastic assumptions of the Gaussian Mixture Model (GMM) and the Hidden Markov Model (HMM). In addition, the most pertinent features for the training were chosen using the dimensionality reduction technique. A dataset representing various network scenarios, including both normal and attacked cases, was used in the independently executed experimentation phase. A 2 HMM/3 GMM classifier was used to achieve a 92.18% classification accuracy as the method's output.

The study conducted by Gebremariam, G. G., et al. [15] outlined a technique for secure attack localization and detection that improves security and service delivery in IoT-WSNs. Prior to beacon nodes transmitting information to the base station, the method generated blockchain trust values using a hierarchical design and Blockchain-based trust assessment and cascade encryption. According to simulation results, nodes' trust value was measured by cascading encryption and feature assessment, which rewarded nodes for their trust and service provisioning. This led to the removal of malicious nodes, which compromised the network's quality of service and localization accuracy. Federated machine learning, which combined unprocessed device data and added malicious threat intelligence to the blockchain, enhanced data security and transmission. Federated learning was utilised to categorise malicious nodes by applying a feature evaluation procedure. This method combined support vector machines, gradient boost, ensemble learning, hybrid random forests, and k-means clustering.

The paper by Lai, Y et al. [9] presented a correlation theory-based malicious-node identification technique to prevent fault data injection attacks. Initially, anomalies between related types of sensor data were found using time correlation. Malicious nodes were then identified using spatial correlation. In the end, event correlation was employed to confirm the presence of the detected malicious nodes.

Ahlawat, P., et al. [16] presented the convolution coding approach, a method based on a sophisticated encoding technique. The first bits were assigned based on the requirements of the network, and each node's final code was generated using the convolution technique. Since it was a digital method, the binary number system could be used to represent the codes. Every node within the network possessed its unique final binary code, represented by the letter C. Before transmitting the data, each node was verified by comparing the generated code C together with the security code during a set amount of time. This procedure made it easier to identify hostile or intruding nodes. Additionally, the system was divided into clusters to improve flexibility and performance.

The paper by Sharma, T., et al. [17] aims to increase wireless sensor networks' dependability and security by introducing a malicious node detection algorithm based on the density-based unsupervised learning technique known as the DBSCAN algorithm. The main goal of the algorithm was to create a routing system that could identify malicious nodes, increase network stability, and extend the life of the network. Within machine learning, two widely used methods were clustering and classification that worked well across a range of applications. In many different fields, density-based clustering was a well-liked and widely applied method. The most well-known and widely used density-based clustering algorithm, known for its ability to identify clusters of any shape, was called DBSCAN. The study focused on two WSN anomalies: malicious node identification and spatial redundancy. In order to save energy and prevent data falsification by malicious nodes, an algorithm was proposed in the article to identify suspicious nodes and reduce redundant data transmission.

Ding, J., et al. [18] used an algorithm for reinforcement learning (RL) to model a selective forwarding attack against malicious nodes with intelligence. The purpose of the double-threshold density peaks clustering (DT-DPC) algorithm was to effectively identify the selective forwarding attack in a difficult setting. Because of persistent abnormalities, abnormal nodes were classified as malicious and isolated. As a result of distinct malicious behaviours and a severe setting that caused agglomerate nodes to malfunction universally, suspicious nodes were identified using the neighbour voting method. DT-DPC increased network throughput even when cunning malicious nodes avoided being discovered by an RL algorithm. As demonstrated by DT-DPC demonstrated a low missed detection rate (MDR) in the simulation results of approximately 10% and a low false detection rate (FDR) of approximately 1%. In a challenging environment, the network throughput increased by roughly 4%.

Research Gaps

Several research gaps persist in the domain of securing Wireless Sensor Networks (WSNs). Firstly, while existing studies propose various detection and prevention techniques, there's a need for more comprehensive approaches that address evolving threats, including sophisticated attacks targeting network infrastructure and data integrity. Additionally, the scalability and efficiency of proposed solutions require further investigation, especially in large-scale WSN deployments. Moreover, the integration of emerging knowledges such as blockchain, federated learning, and reinforcement learning into WSN security frameworks remains relatively unexplored. Bridging these gaps is crucial for enhancing the resilience and reliability of WSNs in the face of emerging security challenges.

3. Methods

Figure 1 shows the proposed work flow of the malicious node discovery model.

Figure 1: Block Diagram

Network Model

By WSN clustering seeks to reduce energy consumption. Common nodes constantly keep an eye on their surroundings and relay sensory information towards the cluster leader. There is only one common node from which the CH node is selected. Transmitting data to the BS from each cluster node requires the CH to perform a critical function. Grouping helps to prevent direct communication between sensors and receivers. In Figure 2, the WSN system model is shown [19].

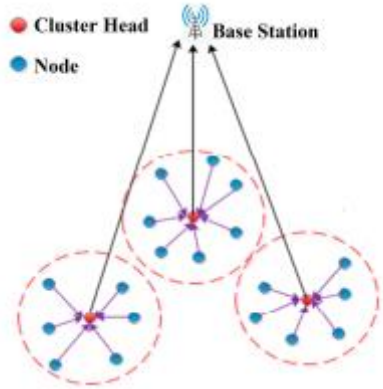

Figure 2. An all-encompassing wireless sensor network architecture.

Conditions for the Ideal Selection of CH

The following are the many criteria that were employed to choose the best CH [20]:

- Energy;
- Security;
- Distance;
- Delay.

Energy Model

Energy exploitation is the main problem with WSNs. Given that the WSN battery does not use the re-energizing method, it is impossible to provide energy in the event that the battery runs out. Furthermore, data from entire SNs is capable of being transmitted to the BS through additional resources. Energy use is essential for data transmission. The network consumes more power as a result of does so many different things, such as aggregation, sensing, transmission, and reception. Normally, Equation (1) implies the necessary energy for the entire data broadcast; under the suggested energy model, however, Equation (2) implies the necessary energy for the entire data broadcast, where E^i denotes initial energy. According to the modules mentioned, the implied electron energy is E_el, in which E_ea indicates the energy used to aggregate data over time. E_e implies that which Equation (4). E_TX (Z:di) shows the total amount of energy used for communication Z packets' bytes at different distances. High energy is necessary for an effective system, utilising the energy of the receiver, E_RX, necessary to obtain Z packet bytes at di is represented by Equation (5) illustrates the energy needed to amplify, E_am.

$$E_{TX}(Z:di) = E\begin{cases} E_{el} \times Z + E_{rs} \times Z \times di^2, \text{if } di < di_0 \\ E_{el} \times Z + E_{pv} \times Z \times di^2, \text{if } di \geq di_0 \end{cases} \quad (1)$$

$$E_{TX}(Z:di) = E\begin{cases} E^i - E_{el} \times Z + E_{rs} \times Z \times di^2, \text{if } di < di_0 \\ E^i - E_{el} \times Z + E_{pv} \times Z \times di^2, \text{if } di \geq di_0 \end{cases} \quad (2)$$

$$E_{el} = E_{TX} + E_{ed} \quad (3)$$

$$E_{RX}(Z:di) = E_{el}Z \quad (4)$$

$$E_{am} = E_{fr}di^2 \quad (5)$$

$$di_0 = \sqrt{\frac{E_{fr}}{E_{pam}}} \quad (6)$$

In the above equations, di_0 stands for threshold distance; E_"pam " represents energy of PA; E_fr represents the necessary energy when using the free space method; E_1 symbolises the energy of the whole inactive state; E_C shows the cost of energy for the entire sense phase. Equation (7) indicates the total energy required to transmit data.

$$E_{\text{total}} = E_{TX} + E_{RX} + E_1 + E_C \quad (7)$$

Security

Mode of security: This chooses the CH that meets the requirements for security. In Equation (8), q_r&q_s mentions, in that order, security rank and CHS-related security requirements. The symbols for the nodes are CH, if q_s≤q_r.

Risky mode: To obtain the best CH for capturing every risk, an existing CH is chosen in this case. Consequently, during the CHS procedure, this mode is referred to as insistent mode.

γ-risky mode: In γ-risky mode, the CH who are subject to a severe risk are selected. Furthermore, γ-risk is referred to as u_"risk " . Then, as in the other two modes, γ indicates the probability metric with η=0 and γ=1.

Equation (8) illustrates the probability of security variables. Furthermore, if the selected CH accomplishes the state q_s>q_r, less than 50% of the risk should exist. If the circumstance is 0<q_s-q_r≤1, selection procedure would be adhered to, and should the state be 1<q_s-q_r≤2, The selection procedure will take longer than expected. But the CHS procedure wouldn't end there, and the state should keep performing the related role 2<q_s-q_r≤5. Consideration is given to the risk factor in the security analysis. As a result, there must be little security.

$$Se = \begin{cases} 0 & \text{if} \quad q_s - q_r \leq 0 \\ 1 - e^{\frac{(q_s - q_r)}{2}} & \text{if} \quad 0 < q_s - q_r \leq 1 \\ 1 - e^{\frac{3(q_s - q_r)}{2}} & \text{if} \quad 1 < q_s - q_r \leq 2 \\ 1 & \text{if} \quad 2 < q_s - q_r \leq 5 \end{cases} \quad (8)$$

Distance

Equation (9). computes the packet communication distance from regular nodes to the CH and BS from the CH. A good system needs to have a short distance.

$$Di = \frac{Di_{\text{dist}}^{(m)}}{Di_{\text{dist}}^{(n)}} \quad (9)$$

$$\text{where, } Di_{\text{dist}}^{(m)} = \sum_{q=1}^{d} \sum_{t=1}^{M_e} \left\| d_q^{\text{norm}} - M_c^t \right\| + \left\| M_c^t - I_k \right\| \quad (10)$$

$$Di_{\text{dist}}^{(n)} = \sum_{q=1}^{d} \sum_{t=1}^{d} \left\| d_q^{\text{norm}} - d_t^{\text{norm}} \right\| \quad (11)$$

Delay

Equation (12), in which d denotes the network's total cluster count, is used to calculate delay and M_c^e suggests the related CH. Minimum delay is necessary for a good system.

$$De_{del} = \frac{M_c^e}{\frac{M_c^t(M_c^t)}{d}} \quad (12)$$

Trust

In order to gauge the degree of trust between hops and adjacent hops, higher trust is offered by every hop in the WSN. Equation (13) indicates that trust can be assessed using two factors: direct and indirect trust. A good system needs a high degree of trust.

$$\text{Tr} = \left\{ \text{Tr}^d + \text{Tr}^{id} \right\} \quad (13)$$

Direct trust: computed as indicated by Equation (14), where, $(Tr^d)_y^z$ indicates unwavering trust for the yth transaction and the zth time frame, sm stands for the satisfaction metric, z denotes Time interval, y signifies a transaction, o indicates Hop Estimation, and $o+1$ denotes Hop for assessment.

$$(O^d)_y^z(o, o+1) = \text{sm}_y^z\ (o, o+1) \quad (14)$$

In Equation (15), sm is assessed as in Equation (15), wherein sm_v indicates the level of satisfaction with the current gearbox, $sm_{y-1}^z(o, o+1)$ denotes $y-1$ value of transmission satisfaction at the zth time interval, where η stands for weight.

$$sm_y^z(0, o+1) = \eta \times sm_v + (1-\eta) \times m_{y-1}^z(0, o+1) \quad (15)$$

$$sm_v = \begin{cases} 0;\ \text{if transmission is unsatisfactory} \\ 1;\ \text{if transmission is satisfactory} \\ \in (0,1);\ \text{else} \end{cases} \quad (16)$$

Indirect trust: indirect faith in the hop with reference to $(o+1)$ th is evaluated in accordance with Equation (17), where V represents the agent group communicating with $o+1, a$ denotes hop, and K_y^z indicates the creditability of the feedback. K_y^z is evaluated using Equation (18), which L_y^z indicates similarity. Equation (19), where l is the similarity deviation constant, is used to evaluate the similarity between hops, $\delta\&\omega$ indicates the factor of reward and punishment, and $\Re_y^z(0,0+1)$ denotes personalized difference.

$$\left(\text{Tr}^{id}\right)_y^z(o, o+1) = \begin{cases} \frac{\sum_{a \in U-\{o\}} K_y^z(o,a) \times \left(\text{Tr}^d\right)_y^z(a, o+1)}{\sum_{a \in U-\{o\}} K_y^z(o,a)} \\ 0;\ \text{if } |v-\{o\}| > 0 \end{cases} \quad (17)$$

$$K_y^z(o, o+1) = \begin{cases} 1 - \frac{\ln\left(L_y^z(o,o+1)\right)}{\ln \varphi};\ \text{if } \left(L_y^z(o, o+1)\right) > \varphi \\ 0;\ \text{else} \end{cases} \quad (18)$$

$$L_y^z(0,0+1)=\begin{cases} L_{y-1}^z(0,0+1)+\frac{1-L_{y-1}^z(0,0+1)}{\omega};\ \text{if}\ \Re_y^z(0,0+1)<l \\ L_{y-1}^z(0,0+1)-\frac{1-L_{y-1}^z(0,0+1)}{\delta};\ \text{else} \end{cases} \quad (19)$$

RSSI

According to Fris, RSSI becomes distorted when the inverse square of the separation between the sender and the recipient. Equation (20) can be used to model it, and Equation (21) can be used to compute B, which stands for distance. An effective system requires a high RSSI.

$$\text{RSSI}=-36\times\log(\text{B})-55 \quad (20)$$

$$\text{B}=10^{(\text{RSSI}+55)/-36} \quad (21)$$

Objectives

Equation (22) illustrates the goal of the method based on BEA-SSA for choosing the best CH, where di is the distance, E^p is the energy, Se is the security, De is the delay, Tr is the trust, and the term "Received Signal Strength Indicator" refers to RSSI. Below is a representation of the goal function. The goal is to minimise the parameters of distance, security, and delay while maximising the RSSI, energy, and trust parameters. In this case, security analysis considers the risk factor. So, the lowest possible risk factor is necessary.

$$Obj=\min\begin{bmatrix}(we_1\times di)+\left(we_2\times(1-E^p)\right)+(we_3\times(Se)) \\ +(we_4\times De)+(we_5\times(1-\text{Tr}))+(we_6(1-\text{RSSI}))\end{bmatrix} \quad (22)$$

we_1-we_6 are weighing variables that vary from 0 to 1; we_1 is paired with $0.2, we_2$ is paired with $0.3, we_3$ is paired with$0.1, we_4$ is paired with $0.2, we_5$ is paired with 0.1 , and we_5 is paired with 0.1 .

Clustering using KGMO based MRSA

The initial use of KGMO [21] in this paper has an inertia weight disadvantage. To solve this problem this paper uses a MRSA. A recently created optimisation method that effectively addresses a variety of optimisation issues is called the RSA. However, the RSA has certain disadvantages, involving high computational complexity, sluggish convergence, and local minima trapping, when attempting to solve nonconvex, high-dimensional optimisation problems. As a result, some modifications to the original RSA algorithm are suggested in order to address these problems [22].

It is necessary for the solution candidates to look as far across the search space as they can to avoid local minima trapping. Consequently, a sine operator was added to the high walking stage of the RSA algorithm to improve exploration. The sine cosine algorithm's dynamic exploration mechanism (SCA) served as the inspiration for this modification. One can perform global exploration with the help of the sine operator. Therefore, by performing a thorough search of the solution space, one way to avoid local minima trapping in the IRSA is to include a sine operator. Using the sine operator, the MRSA equation was changed to the following one.

$$x_{jk}(\tau+1)=\text{Best}_k(\tau)+\left(r_1\times\sin(\text{rand})\times\left|r_2\times\text{Best}_k(\tau)-x_{jk}\right|,\ \text{for}\ \tau\le\frac{T}{3}\right. \quad (23)$$

where r_1, r_2, and Randomly selected numbers are known as rand 0 and 1. x_{jk} is the situation as it stands now, and Best $_k$ is the ideal remedy. The Levy distribution function is followed by a random process known as the Levy flight.

$$\text{levy}=0.01\times\frac{u}{v^{\frac{1}{\zeta}}} \quad (24)$$

where u and v obey normal distribution.

$$u \sim (0, \sigma_u^2), v \sim (0, \sigma_v^2) \quad (25)$$

$$\sigma_u = \left(\frac{\delta(1+\zeta)\sin\frac{\pi\zeta}{2}}{\delta\left[\frac{1+\zeta}{2}\right]\zeta * 2^{\zeta-\frac{1}{2}}} \right)^{1/\zeta} \quad (26)$$

$$\sigma_v = 1 \quad (27)$$

where δ is a typical gamma function. ζ is a crucial factor that establishes the jump size in the Levy fight. The reduced amount of ζ leads to sporadic, tiny steps. This enhances exploitation capabilities by allowing the region nearest the obtained solution should be searched by the solution candidates. The enhanced utilisation ensures worldwide convergence. In the last stages of MRSA, the position is updated using the Levy operator

$$x_{jk}(\tau+1) = \text{Best}_k(\tau) + \text{randn} \times \text{levy} \oplus \left(x_{jk} - \text{Best}_k(\tau)\right), \text{for } \tau \le T \text{ and } \tau > 3\frac{T}{4} \quad (28)$$

where $\oplus$ indicates multiplication by entry, and rand n is a random number with a uniform distribution.

The algorithm's complexity is significantly reduced by these improvisations. Nevertheless, we are able to remove these equations from the algorithm by making the aforementioned modifications. This implies that the time complexity of the suggested MRSA algorithms is reduced by nearly three to four times since they are not required to calculate these equations.

The suggested technique shows improvements in global exploration, speed, low time high efficiency, and high efficiency. When combined, the MRSA's pseudocode is displayed in Algorithm 1.

Algorithm 1 Pseudocode of MRSA

Initialize random population x

Initialize iteration counter τ=0, extreme iteration T, alpha, beta

while τ<T

Evaluate fitness of potential candidates

Determines the best solution

Update E_s, $P_{(j,k)}$

for j=1:p

for k=1:n

If τ≤T/3

Solve using Equation (23)

else if τ≤2 T/4 and τ>T/3

else if τ≤3 T/4 and τ>2 T/4

else

Solve using Equation (28)

end if

end for

end for

t=t+1

end while

Return best solution

WSN Routing Algorithm for E-CERP

The energy-efficient cross-layer-based transfers data over the shortest paths in a scalable and energy-efficient manner. It lowers packet delays and improves the communication link's dependability [19]. There are various issues with the current CORP algorithm, this is an opportunistic routing protocol that operates across layers:

- Because of the limited computing power of the WSNs, an increasing number of constraints results in high data transmission complexity.
- It is difficult to integrate them.
- They use a lot of power and have high delivery rates, packet losses, and communication delays.

The objective of the suggested E-CERP technology, which addresses the issues with traditional routing protocols, is to optimise the broadcast power of the nodes by making use of the network's residual energy.

(i) Local broadcast

The ID, path cost, and hop every node is infinite, it may vary from round to round. Considering the HELLO messages that it gets, every node arranges its list of neighbours. The received signal strength indicator (RSSI) of incoming correspondence is used to calculate signal strength. The link reliability metric is calculated using the average RSSI $L(n,m)$, expressed in Equation (29).

$$L_{n,m} = T_{mtp} - T_{arp} \tag{29}$$

The following is how the link cost is estimated using the Lagrange multiplier:

$$\cos(n,m) = X_{cir} + X_{arp} + \frac{\ln I}{v} \cdot \frac{1}{h_{nm}} \tag{30}$$

(ii) Routing algorithm

A parent Par (n) must be selected to serve as the subsequent hop sensor node for a given sensor node n in order to send information to the BS from node n. The formula in mathematics can be expressed in the following way:

$$\text{Par}(n) = \arg_{m \in N_i(n)} \min[\text{cost}(m) + \text{cost}(n,m)] \tag{31}$$

A cost-based route is constructed using the parent route selection and the cost function as indicated in Equation (10). Updates are made to the parent preferences in every round.

(iii) Transmission Power Control (TPC)

Equation (32) can be used to determine the optimal transmit power based on hop count:

$$T_{tx(n)} = \frac{1}{h_n}\left[\ln\ I + \ln\left(h_n \sum_{n=1}^{H} \frac{1}{h_n}\right)\right] \quad (32)$$

Relatively speaking, the transmitted execution is simplified, and Equation (33) can be used to determine the upper bound:

$$\left(\sum_{n=1}^{H} \frac{1}{h_n}\right) \ln I \leq \sum_{n=1}^{H} X_{tx(n)} \leq \left(\sum_{n=1}^{H} \frac{1}{h_n}\right)(\ln I + \ln H) \quad (33)$$

Equation (34) is thus used to estimate the upper bound for the given sensor node.

$$X_{tx}(n) = \frac{1}{h_{nm}}[\ln\ I + \ln\ H(n)] \quad (34)$$

where " I " is the intended end-to-end success likelihood and " H " symbolises the number of hops. After assessing the transmitting power, the optimal path for data broadcast and reception has been identified. It uses very little energy conservation and has very little packet loss and delay.

Malicious node detection

WUSTL-IIOT-2018 Dataset

Targeting reconnaissance attacks on SCADA is the goal of the ICS data set WUSTL-IIoT-2018 for SCADA cyber security studies. Among recognition's components the use of examining tools by intruders to find network devices and potential vulnerability sites [23]. The subsequent reconnaissance attacks Address scanning, port scanning, and device identification and exploitation were carried out against the testbed. Using FS, the writers were able to pinpoint traits that hold the greatest promise for the success of the data set, even though the basic data contained 25 networking features: The total transaction packet count (Totpkts), total transaction bytes (TotBytes), source/destination packet count (SrcPkts), and destination/source packet count (DstPkts) are all displayed. These features serve as a prototype for researching model evaluation and machine learning decisions. Additionally, a system for audit record production and implementation was in place to monitor all network traffic, of which 6.07% was malicious and 93.93% was benign.

Classification using Bidirectional Recurrent Neural Networks

The network that is bidirectional in rather than eliminating individual components, the model begins with a spatial dropout layer that eliminates entire feature maps [24]. The bidirectional RNN (BiRNN) layer, which links two hidden layers with opposing directions (forward and backward) to the same output, is then fed the result of this layer. GRU is the foundation of BiRNN. The output of the BiRNN layer is subsequently simultaneously received by the global average and global maximum pooling layers. The next step's new input is created by combining the outputs of these two layers.

A feature map grid (window) or several windows are created from each input feature map. The mean of an n-sized window is determined by the average pooling function in the following manner:

$$\frac{x_1 + x_2 + \cdots + x_n}{n} \quad (35)$$

The input window's maximum value number is selected by max-pooling $\{x_1, .. x_n\}$. The objective of average and maximising the number of dimensions in the data while preserving important information is the goal of pooling [25].

The one GRU cell has the following functions:

$$z_{ti} = \sigma(W_{xz}x_{ti} + W_{sz}s_{ti-1} + b_z) \quad (36)$$
$$e_{ti} = \sigma(W_{xe}x_{ti} + W_{se}s_{ti-1} + b_e) \quad (37)$$
$$g_{ti} = \tanh\left(W_{xg}x_{ti} + W_{hg}(e_{ti} \odot h_{ti-1}) + b_g\right) \quad (38)$$
$$s_t i = (1 - z_{ti}) \odot s_{ti-1} + z_{ti} \odot g_{ti}. \quad (39)$$

where x_{ti} is the GRU cell's input at that moment ti. W_{xg}, W_{xz} and W_{xe} are the input-receiving weight matrices X_{ti}. W_{sg}, W_{se} and W_{sz} are the weight matrices with the prior cell state vector as their input.

tanh is an activation function for hyperbolic tangents, and σ is an activation function for sigmoid. b_e, b_g, and b_z the bias units. s_{ti} is the result at that moment ti. $\odot$ alludes to the Hadamard item [26].

Each GRN cell in a BiRNN computes the hidden state moving forward $\overrightarrow{s_{tı-1}}$ and the reverse path $\overleftarrow{s_{tı+1}}$ As a result, Features that benefit the BiGRU in both directions. The concept of BiRNNs is explained by the following equation.

$$s_{ti} = \overrightarrow{s_{tı-1}} \oplus \overleftarrow{s_{tı+1}} \quad (40)$$

where $\oplus$ represents the elementwise sum of the two vectors in the left and right directions.

4. Results and Discussions

Experimental Setup

Python was utilised in the implementation of the suggested approach. A number of Python libraries were used for preprocessing, cleaning the data, and implementing the model. The scikit-learn library was used to assess the deep learning model after it was put into practice using the Keras library. Rather than utilising a graphics processing unit, it employed the tensor cloud-based Python notebook designed to foster teamwork. Table 1 lists the parameters taken into account when setting up the simulation.

Table 1. Configuration parameters for the simulation.

Parameters	Value
Number of nodes	500
Deployment area	500 × 500
Total Clusters	6
Packet size	512 bytes
Packet sending rate	1 packet/s
Initial energy	0.5J
Data testers	55

Performance Metrics

Packet Delivery Ratio (PDR):

$$\text{PDR} = \frac{\text{Number of Packets Received}}{\text{Number of Packets Sent}} \times 100\% \quad (41)$$

Packet Delay:

$$\text{Packet Delay} = \frac{\text{Total time taken for all packets to reach destination}}{\text{Number of Packets Sent}} \quad (42)$$

Throughput:

$$\text{Throughput} = \frac{\text{Total amount of datatransferred}}{\text{Total time taken}} \quad (43)$$

Power Consumption:

$$\text{Power Consumption} = \text{Voltage} \times \text{Current} \times \text{Time} \quad (44)$$

Accuracy:

$$Accuracy\ (ACC) = \frac{No.of\ correctly\ expressions}{Total\ no.of\ images} \times 100 \quad (31)$$

Precision:

$$precision\ (PR) = \frac{TP}{TP+FP} \times 100 \quad (32)$$

F1-score:

$$F1 - score\ (F1) = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100 \quad (33)$$

Recall:

$$Recall\ (RC) = \frac{TP}{TP+FN} \times 100 \quad (34)$$

Specificity:

$$Specificity\ (SP) = \frac{TN}{TN+FP} \times 100 \quad (35)$$

Clustering Evaluation

Table 2-5 shows the performance analysis of the projected KGMO based MRSA with various number of nodes.

Table 2: PDR validation

Models	**100**	**200**	**300**	**400**	**500**
RSA	0.94	0.93	0.92	0.91	0.90
MRSA	0.95	0.94	0.93	0.92	0.91
KGMO	0.97	0.96	0.95	0.94	0.93
Proposed KGMO based MRSA	0.99	0.98	0.97	0.96	0.95

In Table 2 and figure 3, the Packet Delivery Ratio (PDR) validation results for various models across different nodes are presented. The models evaluated include the Reptile Search Algorithm (RSA), which achieved PDR values of 0.94, 0.93, 0.92, 0.91, and 0.90 for nodes of 100, 200, 300, 400, and 500 nodes respectively. The Modified Reptile Search Algorithm (MRSA) exhibited slightly higher PDR values, with measurements of 0.95, 0.94, 0.93, 0.92, and 0.91 for the same respective network sizes. Meanwhile, the Kinetic Gas Molecules Optimization (KGMO) algorithm demonstrated even better performance, yielding PDR values of 0.97, 0.96, 0.95, 0.94, and 0.93. Notably, the proposed approach, leveraging a KGMO based Modified Reptile Search Algorithm, showcased superior PDR values across all nodes, reaching 0.99, 0.98, 0.97, 0.96, and 0.95 correspondingly. These results underscore the efficacy of the proposed method in significantly enhancing packet delivery reliability compared to conventional algorithms.

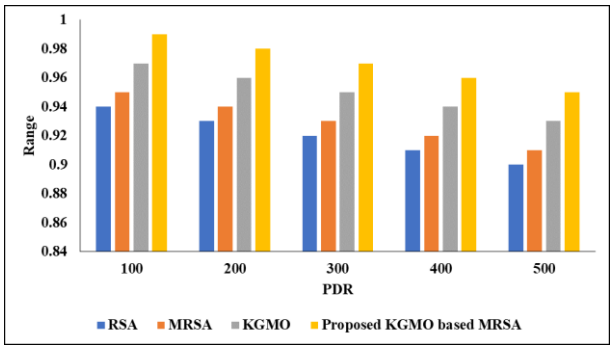

Figure 3: PDR analysis of the malicious node detection

Table 3: Throughput validation

Models	**100**	**200**	**300**	**400**	**500**
RSA	0.93	0.93	0.93	0.92	0.92
MRSA	0.94	0.94	0.95	0.93	0.93
KGMO	0.96	0.97	0.96	0.95	0.95
Proposed KGMO based MRSA	0.98	0.98	0.97	0.97	0.97

Table 3 and figure 4 outlines the results of Throughput validation for different models across varying nodes. The models evaluated include the Reptile Search Algorithm (RSA), which achieved Throughput values of 0.93, 0.93, 0.93, 0.92, and 0.92 for nodes of 100, 200, 300, 400, and 500 nodes respectively. The Modified Reptile Search Algorithm (MRSA) showed slightly improved Throughput, with measurements of 0.94, 0.94, 0.95, 0.93, and 0.93 for the same respective nodes. On the other hand, the Kinetic Gas Molecules Optimization (KGMO) algorithm demonstrated even higher Throughput performance, yielding values of 0.96, 0.97, 0.96, 0.95, and 0.95. Notably, the proposed KGMO based MRSA approach exhibited superior Throughput values across all nodes, reaching 0.98, 0.98, 0.97, 0.97, and 0.97 correspondingly. These results highlight the efficacy of the proposed method in enhancing data transfer efficiency compared to conventional algorithms.

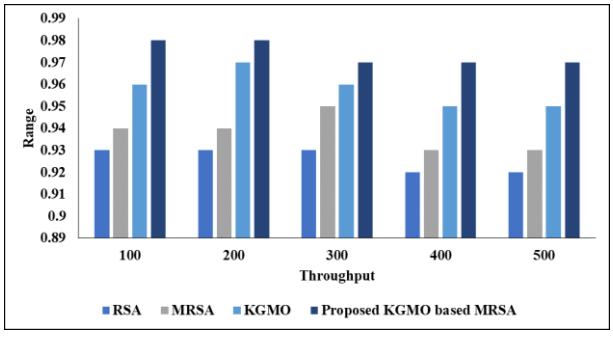

Figure 4: Throughput analysis

Table 4: Packet delay validation

Models	**100**	**200**	**300**	**400**	**500**
RSA	2.94	4.14	5.19	7.32	8.72
MRSA	1.96	3.62	4.16	5.15	6.11
KGMO	1.25	2.13	3.26	4.24	5.16
Proposed KGMO based MRSA	0.05	1.06	2.12	3.18	4.98

Table 4 and figure 5 displays the results of Packet Delay validation for different models across various nodes. The Reptile Search Algorithm (RSA) yielded Packet Delay values of 2.94, 4.14, 5.19, 7.32, and 8.72 for nodes of 100, 200, 300, 400, and 500 nodes respectively. The Modified Reptile Search Algorithm (MRSA) exhibited improved Packet Delay, with measurements of 1.96, 3.62, 4.16, 5.15, and 6.11 for the corresponding nodes. In comparison, the Kinetic Gas Molecules Optimization (KGMO) algorithm demonstrated even lower Packet Delay, with values of 1.25, 2.13, 3.26, 4.24, and 5.16. Particularly noteworthy is the proposed KGMO based MRSA approach, which significantly reduced Packet Delay across all nodes, achieving values as low as 0.05, 1.06, 2.12, 3.18, and 4.98 correspondingly. These results underscore the effectiveness of the proposed technique in minimizing data transmission delays compared to conventional algorithms.

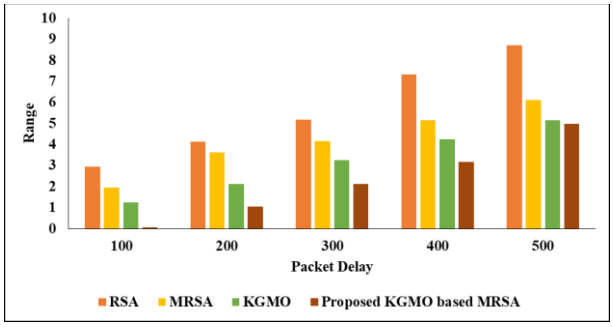

Figure 5: Packet Delay validation

Table 5: Power consumption validation

Models	**100**	**200**	**300**	**400**	**500**
RSA	8.45	9.42	10.21	11.13	11.67
MRSA	7.74	5.52	7.89	7.49	7.56
KGMO	5.43	4.23	3.55	4.25	5.14
Proposed KGMO based MRSA	1.52	1.09	2.32	2.16	3.21

In Table 5 and figure 6, the Power Consumption validation results for different models across various nodes are presented. The Reptile Search Algorithm (RSA) exhibited power consumption values of 8.45, 9.42, 10.21, 11.13, and 11.67 for nodes of 100, 200, 300, 400, and 500 nodes respectively. The Modified Reptile Search Algorithm (MRSA) showed varied power consumption, with measurements of 7.74, 5.52, 7.89, 7.49, and 7.56 for the corresponding nodes. In contrast, the Kinetic Gas Molecules Optimization (KGMO) algorithm demonstrated lower power consumption, with values of 5.43, 4.23, 3.55, 4.25, and 5.14. Particularly notable is the proposed KGMO based MRSA approach, which significantly reduced power consumption across all nodes, achieving values as low as 1.52, 1.09, 2.32, 2.16, and 3.21 respectively. These results highlight the effectiveness of the projected method in minimizing energy usage compared to conventional algorithms, thus contributing to enhanced efficiency and sustainability in wireless sensor networks.

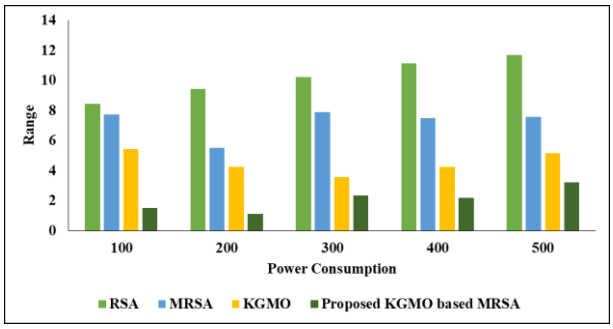

Figure 6: Power Consumption analysis

Classification validation

Table 6 presents the malicious node detection validation of the proposed BiRNN classical.

Table 6: Attack detection validation using various DL models

Models	ACC (%)	PR (%)	RC (%)	F1 (%)	SP (%)
AE	85.22	85.17	85.14	85.12	85.11
DBN	87.54	88.21	82.26	89.53	88.47
ANN	88.38	89.52	87.36	92.23	88.27
RNN	94.1	95.72	95.26	95.23	95.92
Proposed BiRNN model	99.21	99.13	98.02	98.09	98.77

Table 6 and figure 7 summarizes the results of Attack Detection validation using various Deep Learning (DL) models. The models evaluated include Autoencoder (AE), Deep Belief and the proposed Bidirectional Recurrent Neural Network (BiRNN) model. Each model's performance is assessed based on several metrics, including Accuracy (ACC), Precision (PR), Recall (RC), F1-score (F1), and Specificity (SP). The Autoencoder (AE) achieved an accuracy of 85.22% with corresponding precision, recall, F1-score, and specificity values of 85.17%, 85.14%, 85.12%, and 85.11% respectively. The Deep showed an accuracy of 87.54% with precision, recall, F1-score, and specificity values of 88.21%, 82.26%, 89.53%, and 88.47% respectively. The Artificial Neural Network (ANN) exhibited an accuracy of 88.38% with precision, recall, F1-score, and specificity values of 89.52%, 87.36%, 92.23%, and 88.27% respectively. The Recurrent Neural Network (RNN) demonstrated the highest accuracy of 94.1% with precision, recall, F1-score, and specificity values of 95.72%, 95.26%, 95.23%, and 95.92% respectively. Notably, the proposed Bidirectional Recurrent Neural Network (BiRNN) model outperformed all other models with an impressive accuracy of 99.21% and consistently superior capability in accurately detecting attacks in wireless sensor networks.

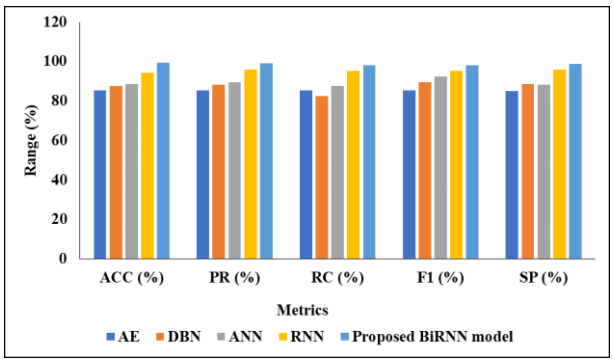

Figure 7: Attack classification analysis of the malicious node

5. Conclusion

To sum up, this study tackles the vital issue of security and energy efficiency in wireless sensor networks (WSNs). To ensure optimal routing while minimising energy consumption, the study incorporates various parameters like distance, energy, security, delay, trust evaluation, and RSSI. For cluster head selection, it uses a KGMO-based Modified Reptile Search Algorithm. Furthermore, by figuring out the shortest path, the suggested energy-efficient cross-layer-based expedient routing protocol (E-CERP) dynamically lowers network overhead. Moreover, implementing the BiRNN model makes it easier to identify malicious nodes, improving network security. The suggested approach outperforms current methods in terms of power consumption, delay, throughput, and error estimation, among other extensive assessment metrics. All things measured, this work makes a substantial contribution to the development of safe and energy-efficient routing protocols for WSNs, opening the door to improved network presentation and dependability in a variety of applications. The proposed model exhibited superior performance across various metrics of Packet Delivery Ratio (PDR) of 0.99, Throughput of 0.98, Packet Delay reduced to 0.05, Power Consumption minimized to 1.52, and an impressive Accuracy of 99.21%, indicating its effectiveness in enhancing network efficiency. Future work may explore ensemble DL models, edge computing integration, and adaptive security mechanisms for enhanced attack detection in WSNs.

References

[1] Mittal, N.; Singh, U.; Salgotra, R. Tree-based threshold-sensitive energy-efficient routing approach for wireless sensor networks.Wirel. Pers. Commun. 2019, 108, 473–492
[2] Gomathi, S., & Gopala Krishnan, C. (2020). Malicious node detection in wireless sensor networks using an efficient secure data aggregation protocol. Wireless Personal Communications, 113, 1775-1790.
[3] Ramasamy, L. K., KP, F. K., Imoize, A. L., Ogbebor, J. O., Kadry, S., & Rho, S. (2021). Blockchain-based wireless sensor networks for malicious node detection: A survey. IEEE Access, 9, 128765-128785.
[4] Kumar, M., Mukherjee, P., Verma, K., Verma, S., & Rawat, D. B. (2021). Improved deep convolutional neural network based malicious node detection and energy-efficient data transmission in wireless sensor networks. IEEE Transactions on Network Science and Engineering, 9(5), 3272-3281.
[5] Yang, H., Zhang, X., & Cheng, F. (2021). A novel algorithm for improving malicious node detection effect in wireless sensor networks. Mobile Networks and Applications, 26, 1564-1573.
[6] Gomathy, V., Padhy, N., Samanta, D., Sivaram, M., Jain, V., & Amiri, I. S. (2020). Malicious node detection using heterogeneous cluster based secure routing protocol (HCBS) in wireless adhoc sensor networks. Journal of Ambient Intelligence and Humanized Computing, 11, 4995-5001.
[7] Kshirsagar, P. R. (2021). Malicious Node Detection in Adhoc Wireless Sensor Networks Using Secure Trust Protocol. Research Journal of Computer Systems and Engineering, 2(2), 12-16.
[8] Morsi, A. M., Barakat, T. M., & Nashaat, A. A. (2020). An efficient and secure malicious node detection model for wireless sensor networks. International Journal of Computer Networks & Communications (IJCNC) Vol, 12.
[9] Lai, Y., Tong, L., Liu, J., Wang, Y., Tang, T., Zhao, Z., & Qin, H. (2022). Identifying malicious nodes in wireless sensor networks based on correlation detection. computers & security, 113, 102540.
[10] Teng, Z., Pang, B., Du, C., & Li, Z. (2020). Malicious node identification strategy with environmental parameters. IEEE Access, 8, 149522-149530.
[11] Ramalingam, V., Mariappan, D. B., Gopal, R., & Baalamurugan, K. M. (2020). An effective social Internet of Things (SIoT) model for malicious node detection in wireless sensor networks. Artificial Intelligence Techniques in IoT Sensor Networks, 181.
[12] Rani, K. S. K., & Vijayalakshmi, R. (2021, May). Experimental Evaluations of Malicious Node Detection on Wireless Sensor Network Environment. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 185-191). IEEE.
[13] Nouman, M., Qasim, U., Nasir, H., Almasoud, A., Imran, M., & Javaid, N. (2023). Malicious Node Detection Using Machine Learning and Distributed Data Storage Using Blockchain in WSNs. IEEE Access, 11, 6106-6121.
[14] Moundounga, A. R. A., Satori, H., Boutazart, Y., & Abderrahim, E. (2023). Malicious attack detection based on continuous Hidden Markov Models in Wireless sensor networks. Microprocessors and Microsystems, 101, 104888.
[15] Gebremariam, G. G., Panda, J., & Indu, S. (2023). Blockchain-Based Secure Localization against Malicious Nodes in IoT-Based Wireless Sensor Networks Using Federated Learning. Wireless Communications and Mobile Computing, 2023.

[16] Ahlawat, P., Singhal, P., Goyal, K., Yadav, K., & Bathla, R. (2022). Malicious Node Detection Using Convolution Technique: Authentication in Wireless Sensor Networks (WSN). In Advances in Malware and Data-Driven Network Security (pp. 94-111). IGI Global.
[17] Sharma, T., Mohapatra, A. K., & Tomar, G. (2022). SDBMND: Secure Density-Based Unsupervised Learning Method with Malicious Node Detection to Improve the Network Lifespan in Densely Deployed WSN. Wireless Communications and Mobile Computing, 2022.
[18] Ding, J., Wang, H., & Wu, Y. (2022). The detection scheme against selective forwarding of smart malicious nodes with reinforcement learning in wireless sensor networks. IEEE Sensors Journal, 22(13), 13696-13706.
[19] Cherappa, V., Thangarajan, T., Meenakshi Sundaram, S. S., Hajjej, F., Munusamy, A. K., & Shanmugam, R. (2023). Energy-Efficient Clustering and Routing Using ASFO and a Cross-Layer-Based Expedient Routing Protocol for Wireless Sensor Networks. Sensors, 23(5), 2788.
[20] Gurumoorthy, S., Subhash, P., Pérez de Prado, R., & Wozniak, M. (2022). Optimal Cluster Head Selection in WSN with Convolutional Neural Network-Based Energy Level Prediction. Sensors, 22(24), 9921.
[21] Mano, A., & Anand, S. (2023). Local Average Based Kinetic Gas Molecular (LA-KGMO) Optimized MR Brain Image Segmentation Using Modified Self Organizing Map (MSOM). Wireless Personal Communications, 128(4), 2703-2723.
[22] Khan, M. K., Zafar, M. H., Rashid, S., Mansoor, M., Moosavi, S. K. R., & Sanfilippo, F. (2023). Improved Reptile Search Optimization Algorithm: Application on Regression and Classification Problems. Applied Sciences, 13(2), 945.
[23] Ahakonye, L. A. C., Nwakanma, C. I., Lee, J. M., & Kim, D. S. (2023). Agnostic CH-DT technique for SCADA network high-dimensional data-aware intrusion detection system. IEEE Internet of Things Journal.
[24] Alotaibi, M., Alotaibi, B., & Razaque, A. (2021). A multichannel deep learning framework for cyberbullying detection on social media. Electronics, 10(21), 2664.
[25] Thirumalraj, A., & Rajesh, T. (2023). An Improved ARO Model for Task Offloading in Vehicular Cloud Computing in VANET.
[26] Aruna, T. M., Kumar, P., Srinidhi, N. N., Divyaraj, G. N., Asha, K., Thirumalraj, A., & Naresh, E. (2023). Effective utilisation of Geospatial Data for peer-to-peer communication among autonomous vehicles using Optimized Machine learning algorithm.