Energy efficient data fusion approach using squirrel search optimization and recurrent neural network

Arulkumar Varatharajan[1], Poonkodi Ramasamy[2], Suguna Marappan[3], Devipriya Ananthavadivel[4], Chemmalar Selvi Govardanan[5]
[1]School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
[2]Department of Information Technology, Sri Eshwar College of Engineering, Coimbatore, India
[3]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
[4]Department of Computing Technologies, SRM Institute of Science and Technology, Chennai, India
[5]School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India

Article Info

Article history:

Received Aug 27, 2022
Revised Feb 25, 2023
Accepted Mar 12, 2023

Keywords:

Data fusion
Energy efficient data fusion
Recurrent neural network
Squirrel search optimization
Wireless sensor networks

ABSTRACT

Sensor networks have helped wireless communication systems. Over the last decade, researchers have focused on energy efficiency in wireless sensor networks. Energy-efficient routing remains unsolved. Because energy-constrained sensors have limited computing capabilities, extending their lifespan is difficult. This work offers a simple, energy-efficient data fusion technique employing zonal node information. Using the witness-based data fusion technique, the evaluated network lifetime, energy consumption, communication overhead, end-to-end delay, and data delivery ratio. Energy-efficient data fusion optimizes energy utilization using squirrel search optimization and a recurrent neural network. The method allows the system to recognize a sensor with excessive energy dissipation and relocate data fusion to a more energy-efficient node. The proposed model was compared against artificial neural network-particle swarm optimization (ANN-PSO), cuckoo optimization algorithm-back propagation neural network (COA-BPNN), Elman neural network-whale optimization algorithm (ENN-WOA), and extreme learning machine-particle swarm optimization (ELM-PSO). The model achieved 94.50% network lifetime, 26.63% communication overhead, 93.85% data delivery ratio, 10.50 ms end-to-end delay, and 282 J energy usage.

Corresponding Author:

Poonkodi Ramasamy
Department of Information Technology, Sri Eshwar College of Engineering
Coimbatore, India
Email: poonkodi.r@sece.ac.in

1. INTRODUCTION

Data fusion combines observations from multiple sensors to comprehensively represent a process or environment [1]. Robotics uses data fusion for object detection, mapping, and localization. Data fusion reduces network traffic and energy usage by combining data from many sensors [2], [3]. Wireless sensor networks (WSN) applications may also have time-constrained communications. Using a dense network (more nodes per m^2) could be beneficial. Dense networks require complicated management solutions. Some data fusion systems can manage message losses, delays, and discards, allowing for greater flexibility in network technology and data distribution [4], [5]. Data fusion analyzes all sensory data to get more exact information. Data fusion improves information collection, network capacity, and longevity [6]. According to the latest survey, routing-based tactics for boosting data fusion remain the most popular compared to classic optimization schemes.

Routing-based techniques place a major emphasis on numerous symptomatic communication problems. If the environment changes, one routing system's solutions may no longer apply to another. Most routing algorithms use a deterministic rather than a probabilistic approach to improve simulation accuracy [7], [8].

The data fusion method did not solve this problem. Traditional optimization relies more on node knowledge and communication, requiring heuristics [9]. Such techniques fail in dynamic mobile sensor networks. New energy-related mathematical models are needed. 90% of the study depends on a single parameter for decision-making, which is incorrect given that the sensor can gather various environmental data [10]. Existing approaches differ from normal data fusion algorithms, so research into enhancing data fusion performance focuses less on WSN energy efficiency. Analytical modeling improves data fusion [11] existing solutions do not address energy dissipation concerns in data fusion systems, resulting in a larger loss of transmission energy [12]. Data fusion approaches need to balance energy economy and algorithm performance satisfactorily. Regarding resolving the issues, they are more symptomatic [13], [14]. Therefore, the issues mentioned above are recognized as being relevant to the proposed research. The problem statement is presented as: “It is a difficult challenge to create a new algorithm that assures energy-efficient data fusion procedures for large-scale sensor networks” [15]. As a result of the proposed effort, a framework that can assure efficient data fusion techniques and prospective energy-saving systems will be developed.

This paper aims to offer a simple and very efficient data fusion technique that uses the nodes' zonal information to accomplish energy-efficient data fusion. The remainder of the paper is divided into the parts that follow. Various literature studies from recent papers are presented in section 2, while the proposed model with structural analysis is discussed in section 3 of this paper. The performance analysis was discussed in section 4, and section 5 represents the conclusion and future works.

2. LITERATURE REVIEW

The research works in [12], [13] discuss wireless routing protocol (WRP), a new protocol that concentrates more on pathways promises to extend system life. In WRP, the sink node aggregates each node’s wake-up time. The sink center is notified to analyse the wake-up times of every sensor node, even if some are far away. Dropped connections are utilized to choose a non-flooding path. Energy aware geographic routing protocol [13] is a WSN energy protocol (EAGRP). This technique increases the scheme's energy efficiency. In the past, researchers employed a normal routing approach without ensuring energy efficiency. Routing algorithms lack redundancy checks, requiring retransmission. Two nodes from different clusters use the same transmission technique. Existing data routing systems are faster but require more energy. Routing techniques fail in evolving wireless sensor networks [14], [15]. The current literature and commonly used approaches only address a subset of research problems, leaving others unresolved. Early research modelled data accumulation, not fusion. These solutions only work with fusion-based methods. No studies have examined data fusion to improve sensor network energy efficiency. Most current techniques improve communication performance by adding complexity. Artificial neural network (ANN), genetic algorithm (GA), and support vector machine (SVM) require more iterations and longer epoch values, making them unfeasible for sensor nodes with low resources. These aspects are crucial for establishing WSN data fusion to boost energy efficiency.

3. METHOD

The primary objective of data fusion was to avoid redundant data transmissions and, as a result, extend the life of sensor nodes in WSNs. Data fusion is a process of one or more sensors gathering perceived results from various sensors and processing them before transmitting them to the sink or base station (BS). The most basic type of data fusion functionality was duplicated suppressions, which means that if more than one or additional sources transmit identical data to the node of data fusion, it will transmit just a single data copy [15]–[17]. Data fusion was critical in WSN because it included direct transmissions of fused data to the BS, which was costly. After all, the BS may be located from a long distance, and sensor nodes in the networks require greater energy power for transferring data over longer distances. As a result, a preferable method was that fewer nodes might send information to this long distance and were referred to as cluster heads (CH) of separate clusters in a WSN. The data fusion node in this witness-based fusion does not directly transfer its result to the BS but rather evaluates the result’s message authentication code (MAC) (MAC a proof). When the data fusion node receives this information, it sends the proofs to the BS. If the data fusion node is compromised and delivers an invalid fusion result to the BS, it must produce invalid proofs on the erroneous result [18]. Figure 1 represents the flowchart of the method.

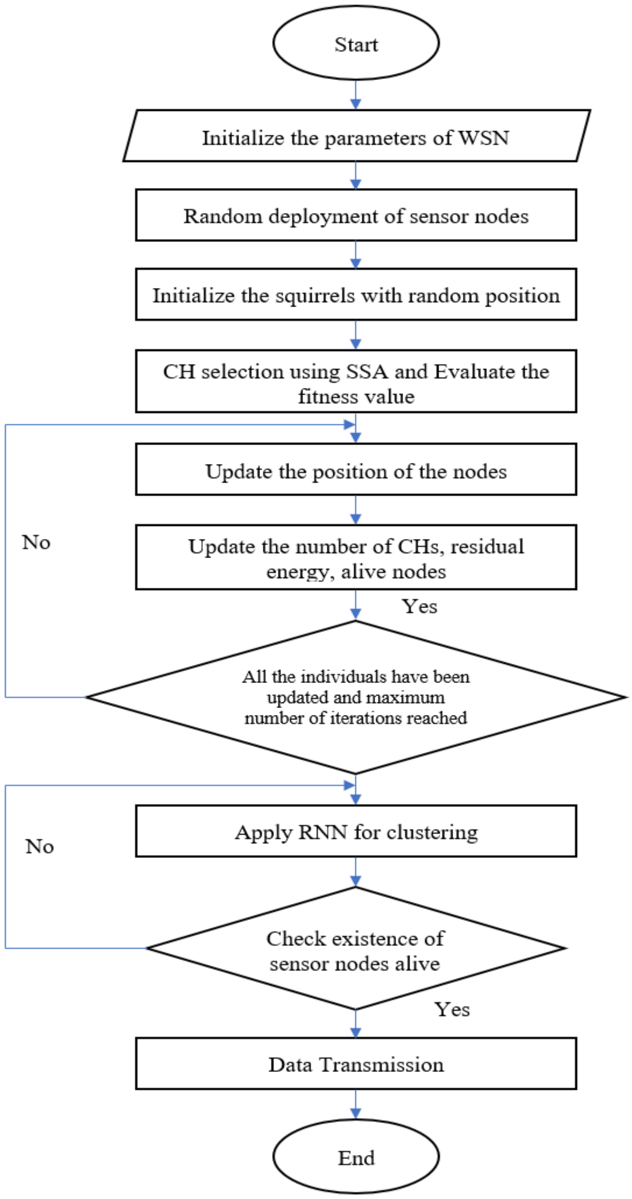

Figure 1. Flowchart of the proposed model

3.1. Squirrel search algorithm

In the squirrel search algorithm (SSA) implementation, the energy of the sensor node was believed to be similar to that of the food supply of squirrels for energy-efficient clustering. Similarly, the movement of the squirrel in SSA corresponds to a change in the position of the cluster head. Flying squirrel movements from acorn or regular tree to hickory are linked to the clustering of less energy nodes going towards higher energy nodes. The best cluster heads are chosen using fitness factors derived from the energy of the sensor node, and the distance between the interacting parts was the primary issue of energy consumption. The WSN is represented by n sensor nodes and k optimum cluster heads, and the cluster head selection is accomplished as:

– Step 1: initialize the network population with k arbitrarily nominated cluster heads.
– Step 2: evaluate the objective function of every sensor node in the network.
– Step 3: sort the node's location in ascending order and divide the nodes into three locations.
– Step 4: generate the new location of the nodes using the squirrel search optimization algorithm.
– Step 5: check the seasonal monitoring condition of the cluster heads.
– Step 6: if the seasonal monitoring condition is true, random relocation of the nodes using the *Levy* distribution.

– Step 7: steps 2 to 6 are repeated up to the maximum limit of the round reached.

The squirrel search algorithm simulates the gliding foraging activities of southern flying squirrels in Europe and Asia's deciduous woodlands. Squirrels glide from tree to tree in quest for food during warm weather. Acorn nuts easily met their energy needs. Then they look for winter-stored hickory nuts (the greatest meal). In cold winters, they store hickory nuts for energy. Flying squirrels become active when the weather warms. The above method was repeated over the squirrels' lifetimes, forming the SSA. The optimization can be mathematically described using squirrel food foraging [19]. Parameterize the algorithm. The SSA's important parameters include maximum iterations (Itermax), population size (NP), decision variable number (n), predator presence probability (P_{dp}), scaling factor (sf), gliding constants (G_c), and decision variable lower and upper bounds (S_L) and (S_U). The SSA function started with these parameters. Squirrel sorting and placement: randomly initializing squirrel locations in search space:

$$S_{i,j} = S_L + rand(\,) * (S_U - S_L), i = 1,2, \dots, NP, j = 1,2, \dots, n \tag{1}$$

where rand () is a random number with a uniform distribution in the interval [0, 1]. The value of fitness $f = (f_1, f_2, \dots, f_{NP})$ of a squirrel's position was computed by changing the decision variable values into the fitness functions in (2) and then, in ascending order, the quality of food sources specified by the fitness value of the flying squirrels' locations is sorted as in (3):

$$f_i = f_i\left(S_{i,1}, S_{i,2}, \dots, S_{i,n}\right), i = 1,2, \dots, NP \tag{2}$$

$$[sorted\_f, sorte\_index] = sort(f) \tag{3}$$

hence categorizing the food source of every squirrel location, three sorts of trees were identified: hickory, oak, and ordinary tree. The best source of food location (i.e., least fitness values) was thought to be the hickory nuts tree (Sht), the location of the next three sources of food were thought to be acorn nut tree (Sat), and the remainder were thought to be ordinary trees (Snt):

$$S_{ht} = S(sorte\_index\ (1)) \tag{4}$$

$$S_{at}(1:3) = S(sorte\_index\ (2:4)) \tag{5}$$

$$S_{nt}(1:NP-4) = S(sorte\_index\ (5:NP)) \tag{6}$$

gliding can be used to create new locations: following the dynamic gliding process of squirrels, three possibilities may emerge [20]. Scenario 1: squirrels on acorn nuts trees prefer hickory nuts trees. The following methods can be used to produce new locations. In (7), dg denotes the gliding distance randomly, $R1$ is the function that returns the values from the distribution of uniform on the range [0, 1], and GC was the gliding constant. Scenario 2: few squirrels on ordinary trees might migrate to an acorn nuts tree to redeem their regular energy requirements. The following methods could be used to produce new locations [21]:

$$S_{at}^{new} = \begin{cases} S_{at}^{old} + d_g G_c\left(S_{ht}^{old} - S_{at}^{old}\right), if\ R_1 \geq P_{dp} \\ random\ location, \qquad otherwise \end{cases} \tag{7}$$

$$S_{nt}^{new} = \begin{cases} S_{nt}^{old} + d_g G_c(S_{at}^{old} - S_{nt}^{old}), if\ R_2 \geq P_{dp} \\ random\ location, \qquad otherwise \end{cases} \tag{8}$$

where R_2 was the function that return the values on the range [0, 1] from the uniform distribution. Scenario 3: few squirrels on ordinary trees might go to hickory nuts trees if its regular energy requirement has been fulfilled. In this instance, the squirrels' new locations could be given by:

$$S_{nt}^{new} = \begin{cases} S_{nt}^{old} + d_g G_c\left(S_{ht}^{old} - S_{nt}^{old}\right), if\ R_3 \geq P_{dp} \\ random\ location, \qquad otherwise \end{cases} \tag{9}$$

in (9), $R3$ was the function that returned the values on the range [0, 1] from the uniform distribution. In every scenario, the gliding distances dg were assumed to be within 9 and 20 m. Still, this value was huge and could generate huge perturbations in (7)-(9), resulting in unacceptable algorithm performance. To obtain adequate algorithm performance, the scaling factor (sf) was applied as the divisor of dg. Seasonal fluctuations have a substantial impact on the foraging behavior of squirrels. As a result, the seasonal monitoring parameter was added to the algorithms to avoid being stuck in local optima solutions. First, the constant seasonal SC and its minimum values are computed.

$$S_c^t = \sqrt{\sum_{k=1}^{n}\left(S_{at,k}^t - S_{ht,k}\right)^2},\ t = 1,2,3 \tag{10}$$

$$S_{cmin} = \frac{10E-6}{365^{Iter/(Iter_{max})/2.5}} \tag{11}$$

The seasonal monitoring status is then examined. The winter is gone under the condition of $S_c^t < S_{cmin}$, and the squirrels who have lost their ability to seek the forests would relocate their searching places randomly for food source. The Levy distribution was the reliable mathematical method that could be used to improve the global search ability of most optimization algorithms. r_a and r_b are two functions that returns the values from uniform distributions on range [0, 1], β was the constant, and σ was computed like in (14), where $\Gamma(x)$=(x-1). Stopping criteria: if the count of iteration was reached maximum, the algorithm was terminated. Else, new location generations and assessing the monitoring state of seasonal are repeated [22].

$$S_{nt}^{new} = S_L + Levy(n) \times (S_U - S_L) \tag{12}$$

$$Levy(x) = 0.01 \times \frac{r_a \times \sigma}{|r_b|^{1/\beta}} \tag{13}$$

$$\sigma = \left(\frac{\Gamma(1+\beta)\times\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\times\beta\times 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{1/\beta} \tag{14}$$

3.2. Recurrent neural network

Recurrent neural networks (RNNs) can observe and represent the dynamic features of a non-linear system. The nodes of RNN have their dynamics, with interlinking weights among them, equivalent to how every sensor node in a wireless sensor network has its dynamics. Like normal neural networks, recurrent networks feature feedback loops [23]. Here, $y(k)$ represents the neural network outputs, $y(k\text{-}1)$ represents past NN outputs, and $u(k\text{-}1)$ represents inputs comprising previous input. The nonlinear functionality F_{NN} was evaluated utilizing the feedforward neural network, which is given as the matrix by:

$$y(k) = F_{NN}(y(k-1), y(k-2), \dots, y(k-m), u(k),$$
$$u(k-1), u(k-2), \dots, u(k-n)) \tag{15}$$

$$F_{NN}(x) = W^T\sigma(V^T x) \tag{16}$$

where x was the neural net inputs, V was the initial-layers weight, W was the second-layers weights, and $\sigma(.)$ was the NN activation functions, which was often a simple sigmoid functionality. The activation function of output was the linear functionality. Above is a two-layer neural network with customizable thresholds, weights, output, and hidden layers. Hidden layers have L neurons, while input layers integrate delayed inputs $u(k)$ and outputs $y(k)$. Using a two-layer neural network with the right weight, smooth functions can be randomly approximated on compact sets. W and V are layer weights. Every continuous functionality f can be randomly approximated by linear combinations of sigmoidal function, according to the neural network global approximation property [24], [25].

$$f(x) = W^T\sigma(V^T x) + \varepsilon(x) \tag{17}$$

Here $\varepsilon(x)$ denotes the neural network error of approximation. On the compact sets S, the reconstruction error was bounded by $\|\varepsilon(x)\| < \varepsilon_N$. Furthermore, for any ε_N, there is a neural network such that $\|\varepsilon(x)\| < \varepsilon_N$ for each $x \in S$. Figure 2 shows a dynamic RNN composed of the sets of dynamic nodes that would offer internal feedbacks to their input. They could be utilized to simulate dynamical system like the sensor networks. WSNs are made up of the huge count of sensors, each with its unique set of dynamics. They communicate with one another and with the base station, which runs the network. To dynamically describe these sensors with no generality loss, it is assumed that one sensor/node [26], [27].

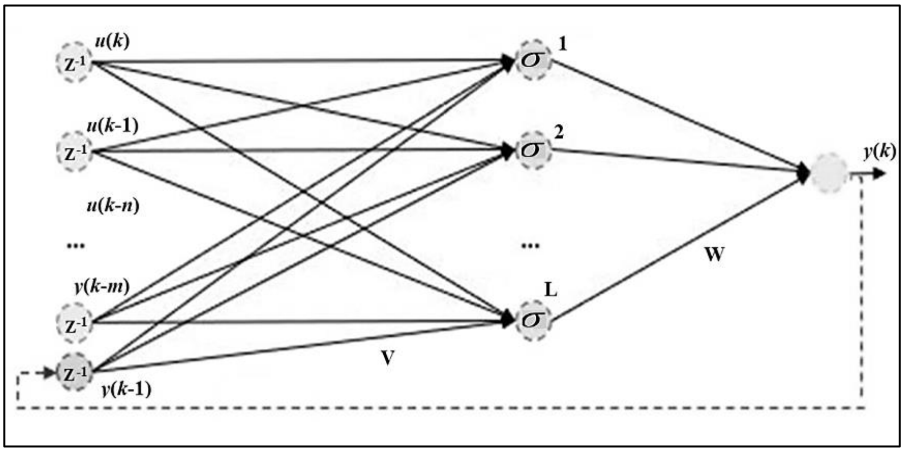

Figure 2. Architecture of the proposed neural network model

Additional sensors/node would simply extend the RNN's size. Consider the following nonlinear dynamical sensor model. Here $u_i(k)$ and $y_i(k)$ were the sensor output and input at the sample *k*, and f_i's were unidentified nonlinear functions. The function f_i must be invertible for the sensors to be operational and the user can specify the true sensor input.

$$y_i(k) = f_i(y_i(k-1), y_i(k-2), \dots, y_i(k-m), u_i(k)) \quad (18)$$

$$u_i(k) = f_i^{-1}(y_i(k-1), y_i(k-2), \dots, y_i(k-m), y_i(k)) \quad (19)$$

According to (19), to identify the physical inputs at the samples *k*, the current and previous *m* sensor output must be known. Every sensor was expected to have model of similar order. If this is not a case, the evaluation could still be performed with minor changes. Assumption 1: sensor node has the same order nonlinear models as stated by (18). Assumption 2: the function f_i was universally Lipschitz function, with L_i representing its Lipschitz constants. While WSN nodes are spread in the region, it is considered that the measured physical number of surrounding nodes is bounded. The assumption was described mathematically as follows. Assumption 3: measurement events at neighbouring sensor nodes differ by the limited constants, i.e., given a sensor node's neighbours *a* and *b*,

$$u_a(k) - u_b(k) = e_{ab}(k) \text{ and } \|e_{ab}(k)\| < e \quad (20)$$

with a model of sensor node *i*, each assumption, and the nodes neighbours that includes node $i_1, i_2, \dots, i_{Ni}$, the sensor node's output could be approximated utilizing RNNs with input comprising of the past output from nodes *i* and its nearby nodes [25]. Here *j*=l, 2, ..., N_i, and *c* is a short-bounded constant. As a result of assumption 3, where *j*=1, 2, ..., N_i. Alternatively, the inputs $u_i(k)$ was defined by,

$$y_i(k) = RNN_i\left(y_i(k-1), y_i(k-2), \dots, y_i(k-m), y_{ij}(k), y_{ij}(k-1), \dots, y_{ij}(k-m)\right) + c \quad (21)$$

$$u_i(k) = \frac{1}{N_i}\sum_{j=1}^{N_i} u_{ij}(k) + e_{ij}(k) \quad (22)$$

hence, one has,

$$y_i(k) = f_i(y_i(k-1), y_i(k-2), \dots, y_i(k-m), \frac{1}{N_i}\sum_{j=1}^{N_i} u_{ij}(k) + e_{ij}(k)) \quad (23)$$

as a result, using the (19), one has.

$$y_i(k) = f_i(y_i(k-1), y_i(k-2), \dots, y_i$$
$$(k-m), \frac{1}{N_i}\sum_{j=1}^{N_i} f_{ij}^{-1}(y_{ij}(k-1), y_{ij}(k-2), \dots, y_{ij}(k-m), y_{ij}(k)) + e_{ij}(k)) \quad (24)$$

Knowing that *f* was the Lipschitz function results in as given in (25). Where *j*=1, 2, ..., N_i, and $\|d\| \leq emax(L_j)$. The RNN that estimate the unknown functions g_i using the neural network parameterization property, thereby:

$$y_i(k) = g_i(y_i(k-1), y_i(k-2), \dots, y_i(k-m), y_{ij}(k), y_{ij} \\ (k-1), \dots, y_{ij}(k-m) + d \tag{25}$$

$$g_i(y_i(k-1), y_i(k-2), \dots, y_i(k-m), y_{ij}(k), y_{ij} \\ (k-1), \dots, y_{ij}(k-m) = RNN_i(x) + \varepsilon_i(x) \tag{26}$$

$$x = \left[y_i(k-1), y_i(k-2), \dots, y_i(k-m), y_{ij}(k), y_{ij}(k-1), \dots, y_{ij}(k-m)\right] \tag{27}$$

$$c = \varepsilon_M + emax(L_j) \tag{28}$$

the vector *x* was specified in (27) and after that, the bounded constant *c* is provided as in (28). This demonstrates that output of the sensor nodes may be approximated as the RNN with input of *m* prior output sample from the similar nodes, as well as present and *m* prior output samples from surrounding sensor. Prior result considers communication of ideal links.

$$y_i(k) = RNN_i(y_i(k-1), y_i(k-2), \dots, y_i(k-m), C_{ji}y_{ij} \\ (k), C_{ji}y_{ij}(k-1), \dots, C_{ji}y_{ij}(k-m)) + c \tag{29}$$

Whether there were communication links uncertainties, the actual values of $y_{ij}(k)$ was not present. Rather, the outputs value of surrounding sensor node was utilized together with confidence factor, i.e., $C_{ji}y_{ij}(k)$. Hence, the RNN sensor node models were given as in [17], [28], [29]. The signal strength among node *i* and the neighbours determines the confidence factors for *i* [30]–[32].

4. RESULTS AND DISCUSSION

The MATLAB 2017a simulation software was used to compare and evaluate the performances of the proposed data fusion model to that of other existing approaches. The sensor nodes were arranged randomly and uniformly in the 2D space of 500×500 m^2, with a total of 100 sensor nodes. There are 400 simulation rounds. The sensor node delivers data packet from the source to sink nodes; each packet is 4 kb in size. The parameter setup for the simulations were represented in Table 1.

Table 1. Simulation setup

Parameter	Numerical value
Network size	500×500 m^2
Number of nodes	100
Communication radius	80 m
Initial energy	1 J
E_{elec}	50 nJ/bit
E_{fs}	10 pJ/bit/m^2
E_{mp}	0.0013 pJ/bit/m^4
L	4000 bits
d_0	$\sqrt{E_{fs}/E_{mp}} = 87m$

Tables 2 to 6 presents the performance analysis comparison of the model evaluated in terms of network lifetime, communication overhead, data delivery ratio, end-to-end delay and energy consumption, individually. The system's architecture is based on an energy-efficient network paradigm and the Figure 3 Shows the graphical representation of the all network performance parameters and in Figure 3(a) shown the performance analysis comparison of network lifetime and in Figure 3(b) shown the communication overhead for the transmission and receiveing. In Figure 3(c) shown the data delivery ratio and Figure 3(d) shown the energy consumptions and all these shows the performance analysis comparison with other models artificial neural network-particle swarm optimization (ANN-PSO) [18], cuckoo optimization algorithm-back propagation neural network (COA-BPNN) [19], Elman neural network-whale optimization algorithm (ENN-WOA) [18], and extreme learning machine-particle swarm optimization (ELM-PSO) [15]. When reviewing models with l bits of data to be processed using a specified level of energy and a distance d, the empirical component of the research entails analyzing models with a given quantity of energy and distance d_0. Eelec is the amount of energy needed to process a unit bit of data for both the receiver and the transmitter. Distance d_0, calculated using the square root of E_{fs} and E_{mp}, reveals the amount of energy lost due to inefficient transmission technology. Network lifespan is the maximum number of data fusion repetitions before a percentage of nodes expires, which varies by application. Data delivery ratio was calculated by dividing the total number of readings received at the base station (sink) by the total number of readings generated. End-to-end average delay was calculated by taking the average of each surviving data packet from the source to the destination. Energy consumption is computed by examining the total energy utilized from the process's beginning node to the process's end node. The proposed model achieved highest network lifetime as 94.50%, communication overhead of 26.63%, data delivery ratio of 93.85%, end to end delay of 10.50 ms, and energy consumptions of 282 J. The proposed model outperforms in all the parameters with improved and better performances.

Table 2. Comparison of network lifetime performance

Mobility (bps)	Network lifetime (%)				
	ANN-PSO	COA-BPNN	ENN-WOA	ELM-PSO	Proposed
10	76.80	78.00	80.00	83.00	86.00
20	78.00	80.45	82.00	85.00	87.00
30	80.50	82.00	84.00	87.70	89.00
40	82.80	84.00	86.50	89.00	91.00
50	84.00	86.50	89.00	92.00	94.50

Table 3. Comparison of communication overhead performance

Nodes	Communication overhead (%)				
	ANN-PSO	COA-BPNN	ENN-WOA	ELM-PSO	Proposed
20	42.86	41.00	26.90	23.86	19.10
40	43.90	42.50	28.10	24.00	21.00
60	44.46	43.00	30.60	26.02	23.22
80	46.02	44.90	31.22	27.00	24.80
100	47.82	45.01	32.20	28.90	26.63

Table 4. Comparison of data delivery rate performance

Number of nodes	Data delivery rate (%)				
	ANN-PSO	COA-BPNN	ENN-WOA	ELM-PSO	Proposed
20	40.00	57.00	83.23	85.90	88.70
40	42.20	58.20	84.75	86.10	89.06
60	43.23	59.18	85.86	85.20	90.10
80	44.91	60.88	86.02	89.20	92.00
100	47.00	62.00	88.20	90.22	93.85

Table 5. Comparison of end-to-end delay performance

Pause time (ms)	End to end delay (sec)				
	ANN-PSO	COA-BPNN	ENN-WOA	ELM-PSO	Proposed
20	17.50	15.80	12.06	10.72	8.01
40	19.89	17.23	13.64	11.10	8.40
60	21.26	18.02	14.95	12.26	9.02
80	22.04	18.90	15.66	13.53	9.47
100	23.50	20.01	17.50	15.02	10.50

Table 6. Comparison of energy consumption performance

Number of nodes	Energy consumption (J)				
	ANN-PSO	COA-BPNN	ENN-WOA	ELM-PSO	Proposed
20	279.00	266.00	270.00	256.00	240.00
40	285.00	278.00	280.00	268.00	248.00
60	299.00	284.00	290.00	276.00	259.00
80	308.00	291.00	299.00	288.00	271.00
100	320.00	300.00	310.00	295.00	282.00

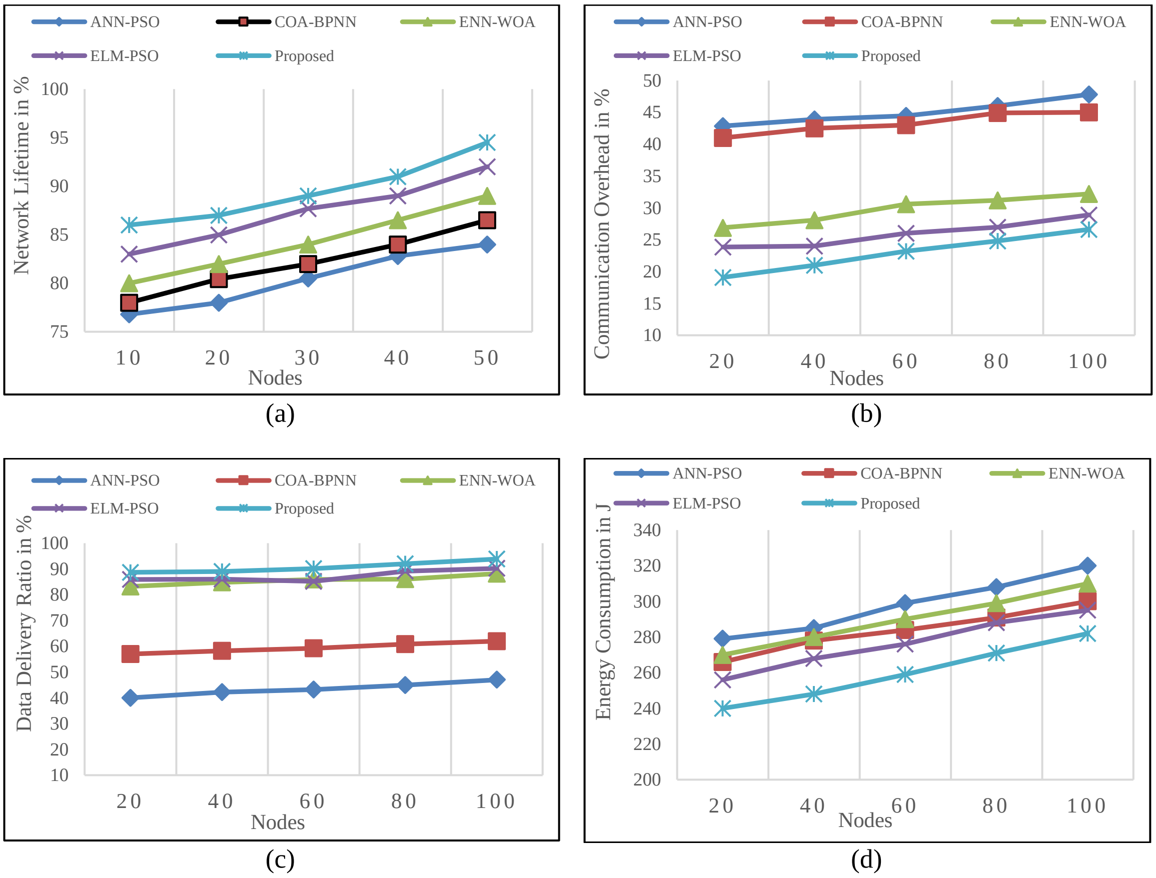

Figure 3. Graphical representation of (a) performance analysis comparison of network lifetime, (b) communication overhead, (c) data delivery ratio, and (d) energy consumption

5. CONCLUSION

This research proposes a squirrel search optimization and recurrent neural network-based energy-efficient data fusion technique for WSN. This work provided an effective and accurate data fusion technique that employs zonal node information for energy-efficient fusion. Energy efficient data fusion optimizes energy utilization using squirrel search optimization and a recurrent neural network. This work used witness-based data fusion to examine network lifetime, energy consumption, communication overhead, data delivery ratio, and end-to-end delay. This technique allows the system to quickly identify the sensor with a higher energy dissipation rate and shift data fusion responsibility to reach more energy-efficient nodes. The proposed model was compared against ANN-PSO, COA-BPNN, ENN-WOA, and ELM-PSO. The proposed model improves all parameters. By simplifying and refining the data fusion technique, the suggested model's performance can be improved and applied to large-scale WSNs.

REFERENCES

[1] R. Tan, G. Xing, B. Liu, J. Wang, and X. Jia, "Exploiting data fusion to improve the coverage of wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 450–462, Apr. 2012, doi: 10.1109/TNET.2011.2164620.

[2] P. Singh, A. K. Ahlawat, and V. Pareek, "Designing an energy efficient network using integration of KSOM, ANN and data fusion techniques," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 9, no. 3, Apr. 2022, doi: 10.17762/ijcnis.v9i3.2769.

[3] B. S. Jayasri and G. R. Rao, "Need for energy efficient data fusion in wireless sensor networks," *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, no. 1, pp. 1663–1668, 2014, doi: 10.17577/IJERTV3IS10619.

[4] Q. Chen, Y. Hu, J. Xia, Z. Chen, and H. W. Tseng, "Data fusion of wireless sensor network for prognosis and diagnosis of mechanical systems," *Microsystem Technologies*, vol. 27, no. 4, pp. 1187–1199, Apr. 2021, doi: 10.1007/s00542-018-4144-3.

[5] E. Brumancia and S. J. Samuel, "An enhanced cluster head technique for energy consumption in data fusion," in *2020 International Conference on Inventive Computation Technologies (ICICT)*, Feb. 2020, pp. 756–761, doi: 10.1109/ICICT48043.2020.9112583.

[6] J. Wang, O. T. Tawose, L. Jiang, and D. Zhao, "A new data fusion algorithm for wireless sensor networks inspired by hesitant fuzzy entropy," *Sensors (Switzerland)*, vol. 19, no. 4, p. 784, Feb. 2019, doi: 10.3390/s19040784.

[7] N. R. Roy and P. Chandra, "Energy dissipation model for wireless sensor networks: a survey," *International Journal of Information Technology (Singapore)*, vol. 12, no. 4, pp. 1343–1353, Dec. 2020, doi: 10.1007/s41870-019-00374-y.

[8] X. Liu *et al.*, “Sponge Supercapacitor rule-based energy management strategy for wireless sensor nodes optimized by using dynamic programing algorithm,” *Energy*, vol. 239, p. 122368, Jan. 2022, doi: 10.1016/j.energy.2021.122368.

[9] M. Shokouhifar and A. Hassanzadeh, “An energy efficient routing protocol in wireless sensor networks using genetic algorithm,” *Advances in Environmental Biology*, vol. 8, no. 21, pp. 86–93, 2014.

[10] A. El-Aalaoui and A. Hajraoui, “Energy efficiency of organized cluster election method in wireless sensor networks,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 18, no. 1, pp. 218–226, Apr. 2019, doi: 10.11591/ijeecs.v18.i1.pp218-226.

[11] A. Das and A. Swaroop, “Energy efficient routing protocol for linear wireless sensor network,” *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017*, no. 21, pp. 703–707, 2017, doi: 10.1109/CCAA.2017.8229892.

[12] M. Bhardwaj, “Faulty link detection in cluster based energy efficient wireless sensor networks,” *International Journal of Scientific Research in Network Security and Communication*, vol. 5, no. 3, pp. 1–8, 2017.

[13] W. Pak, “Ultra-low-power media access control protocol based on clock drift characteristics in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 7, Jul. 2017, doi: 10.1177/1550147717722155.

[14] A. Gaafar and A. Elrahim, “EAGRP: Energy aware geographic routing Protocol for wireless sensor networks,” *International Journal of Computer Science and Telecommunications*, vol. 5, no. 11, pp. 11–16, 2014.

[15] P. Rezayat, M. Mahdavi, M. GhasemZadeh, and M. AghaSarram, “A novel real-time routing protocol in wireless sensor networks,” *Proceedings of the 2009 International Conference on the Current Trends in Information Technology, CTIT 2009*, vol. 10, no. 4, pp. 190–195, 2009, doi: 10.1109/CTIT.2009.5423117.

[16] A. D. Salman, O. I. Khalaf, and G. M. Abdulsaheb, “An adaptive intelligent alarm system for wireless sensor network,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 15, no. 1, pp. 142–147, 2019, doi: 10.11591/ijeecs.v15.i1.pp142-147.

[17] L. Cao, Y. Cai, and Y. Yue, “Data fusion algorithm for heterogeneous wireless sensor networks based on extreme learning machine optimized by particle swarm optimization,” *Journal of Sensors*, vol. 2020, pp. 1–17, Aug. 2020, doi: 10.1155/2020/2549324.

[18] V. Arulkumar, S. Sridhar, G. Kalpana, and K. S. Guruprakash, “Real-Time big data analytics for improving sales in the retail industry via the use of internet of things beacons,” in *Expert Clouds and Applications: Proceedings of ICOECA 2022*, 2022, pp. 111-126, doi: 10.1007/978-981-19-2500-9_8.

[19] N. Lavanya and T. Shankar, “Energy efficient cluster head selection using squirrel search algorithm in wireless sensor networks,” *Journal of Communications*, vol. 15, no. 6, pp. 528–536, 2020, doi: 10.12720/jcm.15.6.528-536.

[20] H. Setiadi and K. O. Jones, “Power system design using firefly algorithm for dynamic stability enhancement,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 1, no. 3, pp. 446–455, Mar. 2016, doi: 10.11591/ijeecs.v1.i3.pp446-455.

[21] R. Poonkodi and N. Saravanaselvam, “Implementation of Wireless Sensor’s Integration Possibilities and Attacks on Wireless Network Control,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 5725-5730, 2019, doi: 10.35940/ijrte.D8483.118419.

[22] O. Obst, “Distributed fault detection in sensor networks using a recurrent neural network,” *Neural Processing Letters*, vol. 40, no. 3, pp. 261–273, Dec. 2014, doi: 10.1007/s11063-013-9327-4.

[23] K. Kolanowski, A. Świetlicka, R. Kapela, J. Pochmara, and A. Rybarczyk, “Multisensor data fusion using Elman neural networks,” *Applied Mathematics and Computation*, vol. 319, pp. 236–244, Feb. 2018, doi: 10.1016/j.amc.2017.02.031.

[24] K. W. Al-Ani, A. S. Abdalkafor, and A. M. Nassar, “An overview of wireless sensor network and its applications,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 17, no. 3, 2019, doi: 10.11591/ijeecs.v17.i3.pp1480-1486.

[25] W. Wu, B. Xu, and M. Cao, “A data fusion method of WSNs based on glowworm swarm algorithm optimized BP neural networks,” *Revista Ibérica de Sistemas e Tecnologias de Informação; Lousada*, vol. 17A, pp. 73–83, 2016, doi: 10.17013/risti.17A.73-83.

[26] O. A. Mahdi, Y. R. B. Al-Mayouf, A. B. Ghazi, M. A. Mohammed, A. W. A. Wahab, and M. Y. I. Bin Idris, “An energy-aware and load-balancing routing scheme for wireless sensor networks,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 12, no. 3, pp. 1312–1319, Dec. 2018, doi: 10.11591/ijeecs.v12.i3.pp1312-1319.

[27] V. Srikanth, R. Walia, P. J. Augustine, R. Venkatesh, J. Simla, and B. Jegajothi, “Chaotic whale optimization based node localization protocol for wireless sensor networks enabled indoor communication,” in *Proceedings of the International Conference on Electronics and Renewable Systems, ICEARS 2022*, Mar. 2022, pp. 702–707, doi: 10.1109/ICEARS53579.2022.9751953.

[28] M. Alqdah, “Intrusion detection attacks classification using machine learning techniques,” *Journal of Computational Science and Intelligent Technologies*, vol. 2, no. 2, pp. 1–6, Aug. 2021, doi: 10.53409/mnaa/jcsit/2201.

[29] R. Jayamma, “Improving the performances of WSN using data scheduler and hierarchical tree,” *Journal of Computational Science and Intelligent Technologies*, vol. 2, no. 2, pp. 7–16, Aug. 2021, doi: 10.53409/mnaa/jcsit/2202.

[30] G. N. Srikanth and M. K. Venkatesha, “Performance analysis of AI models for audio digit utterance detection,” *Journal of Computational Science and Intelligent Technologies*, vol. 3, no. 3, pp. 14-29, 2022, doi: 10.53409/MNAA/JCSIT/e202203031429.

[31] S. Mathupriya, S. S. Banu, S. Sridhar, and B. Arthi, “WITHDRAWN: Digital twin technology on IoT, industries & other smart environments: A survey,” *Materials Today: Proceedings*, vol. 3, no. 1, pp. 17–23, 2020, doi: 10.1016/j.matpr.2020.11.358.

[32] R. Khilar *et al.*, “Artificial intelligence-based security protocols to resist attacks in internet of things,” *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–10, Apr. 2022, doi: 10.1155/2022/1440538.

BIOGRAPHIES OF AUTHORS

Dr. Arulkumar Varatharajan is currently a Assistant Professor Senior G1 in the School of Computer Science and Engineering, Vellore Institute of Technology, Vellore. He has completed his Ph.D. degree in 2020 from Anna University Chennai. He received the B.Tech. degree in Information Technology from Anna University Coimbatore, India, in 2010 and the M.Tech. degree in Information Technology from Anna University Chennai, India, in 2013, respectively. His research interests include image processing, cyber security, machine learning, artificial intelligence, cloud technologies. He has attended several seminars and workshops in the past years. He is a life member of ISTE and the IEEE computer society of India. He has reviewed many research articles from reputed SCOPUS and SCI journals. He can be contacted at email: arulkumaran.ckpc@gmail.com.

Poonkodi Ramasamy completed her bachelor's degree B. Tech Information Technology in 2006 from Maharaja College of Engineering, Coimbatore. She completed her master's degree M.E. Computer Science and Engineering in 2011 from Sasurie college of Engineering and Technology, Coimbatore. Her areas of interest are wireless sensor networks, wireless communication and machine learning. Currently, she is working as an Assistant Professor in Eshwar College of Engineering and pursuing. Her research in the area of wireless sensor networks. She can be contacted at email: poonkodi.r@sece.ac.in.

Dr. Suguna Marappan is working as an Assistant Professor Senior Grade -II at the School of Computer Science and Engineering in Vellore Institute of Technology, Chennai. Her research interests include data analytics, health care analytics, cloud computing and agile project management. Currently, she has completed a Ph.D. in Information and Communication Engineering at Anna University, Chennai, Undergraduate in B.E(CSE) at Kumaraguru College of Technology, Coimbatore, Master's in M.E(CSE) at Government College of Technology, Coimbatore. She has published more than 20 research articles in international journals and 15 international conferences. She is a member of ISTE, IAENG. She can be contacted at email: suguna.m@vit.ac.in.

Dr. Devipriya Ananthavadivel currently working as an Associate Professor in the Department of Computer Technologies at SRM Institute of Science and Technology, Kattankulathur. Anna University, Chennai awarded. Her Ph.D. in Information and Communication Engineering. Her teaching experience exceeds 17 years. She has published over 20 papers in reputable journals, including 4 SCI and 12 Scopus. She has authored 2 books, 2 book chapters, and 4 patents with Springer, Taylor & Francis and presented over 10 papers at national and international conferences. She has guest lectured in Machine Learning, Artificial Intelligence, and Deep Learning at top universities. She judged national project contests, symposiums, and international conferences. She received the 2014-2015 Excellence in Leadership Award from SriGuru Institute of Technology and the 2018-2019 Best Faculty Award from Sri Krishna College of Engineering and Technology in Coimbatore, Tamilnadu. She mentored Xebia Academy Hackathon. MHRD Toycathon 2021 evaluator. She participates in ISTE and ACM. She can be contacted at email: devipria@srmist.edu.in.

Dr. Chemmalar Selvi Govardanan is currently working as an Assistant Professor Senior in the School of Information Technology and Engineering in Vellore Institute of Technology, India since 2015. The author has 11+ years of teaching cum research experience and research interest lies in machine learning and explainable AI. The author has published few research papers in reputed international journals, published 4 book chapters and presented papers in international conferences held at Singapore and Vietnam. She can be contacted at email: chemmalar06@gmail.com.