

Article

Not peer-reviewed version

A Novel Adaptive AI-Based Framework for Node Scheduling Algorithm Selection in Safety-Critical Wireless Sensor Networks

[Issam Al-Nader](#) , [Rand Raheem](#) ^{*} , [Aboubaker Lasebae](#)

Posted Date: 22 August 2025

doi: 10.20944/preprints202508.1562.v1

Keywords: WSN; AI-driven framework; dependable WSN; scheduling algorithm selection; real-time systems; scenario-aware optimization; QoS in WSN



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

A Novel Adaptive AI-Based Framework for Node Scheduling Algorithm Selection in Safety-Critical Wireless Sensor Networks

Issam Al-Nader, Rand Raheem * and Aboubaker Lasebae

Middlesex University, The Burroughs, London NW4 4BT, UK

* Correspondence: R.H.Raheem@mdx.ac.uk

Abstract

Wireless Sensor Networks (WSNs) play a critical role in diverse applications, from environmental monitoring to mission-critical operations. In this type on applications, selecting the most suitable node scheduling algorithm for a given scenario remains a challenge, as no single approach consistently outperforms others under all conditions. Consequently, this study presents an AI-driven selection framework that evaluates scenario-specific requirements—such as coverage, connectivity, and network lifetime—to identify the optimal scheduling algorithm from a pool of algorithms including Hidden Markov Models (HMM), BAT, Bird Flocking, Self-Organizing Feature Maps (SOFM), and Long Short-Term Memory (LSTM). This framework employs a trained neural network, informed by a simulated dataset, to match algorithms to five real-world scenarios: healthcare monitoring, military operations, industrial IoT monitoring, forest fire detection, and disaster recovery. Experimental results show that LSTM frequently achieves near-optimal performance across scenarios, improving dependability e.g., excelling in adaptability, latency reduction, and fault tolerance, while other algorithms demonstrate valuable strengths in specific metrics such as connectivity (HMM) and lifetime (Bird Flocking). These findings demonstrate the effectiveness of the proposed framework in selecting scenario-aware, high-performance WSN scheduling solutions. They also highlight the complementary strengths of different algorithms in meeting diverse operational requirements across scenarios.

Keywords: WSN; AI-driven framework; dependable WSN; scheduling algorithm selection; real-time systems; scenario-aware optimization; QoS in WSN

1. Introduction

WSNs have become a fundamental component of the Internet of Things (IoT), playing a crucial role in everyday applications such as environmental monitoring, industrial automation, healthcare, and defence systems [1–3]. In these systems, it is essential to optimize critical performance metrics such as energy consumption, coverage, network lifetime, latency and other QoS metrics [4,5]. Among the various techniques used in WSNs, sleep/wake scheduling stands out as particularly effective for energy optimization [6]. This technique allows nodes to alternate between active and idle states, thereby minimizing energy consumption while maintaining other network functionality such as coverage and connectivity.

Selecting the most suitable scheduling algorithm for a given WSN scenario remains a significant challenge, especially in safety-critical environments where system failure can result in severe consequences, including loss of human life, economic damage, or catastrophic events. This challenge is compounded by the highly dynamic and context-dependent nature of WSNs. Factors such as node density, communication load, energy availability, latency tolerance, and security requirements can vary significantly across different deployments and over time [6,7]. As a result, a scheduling algorithm that performs well under one set of conditions may prove suboptimal or even harmful

under another. Additionally, manually tuning or selecting algorithms for each scenario is impractical in large-scale or rapidly changing networks. These scenarios include healthcare monitoring, military operations, industrial IoT monitoring, forest fire detection, and disaster recovery as shown in Figure 1. This challenge further emphasizes the need for intelligent and adaptive selecting solution.

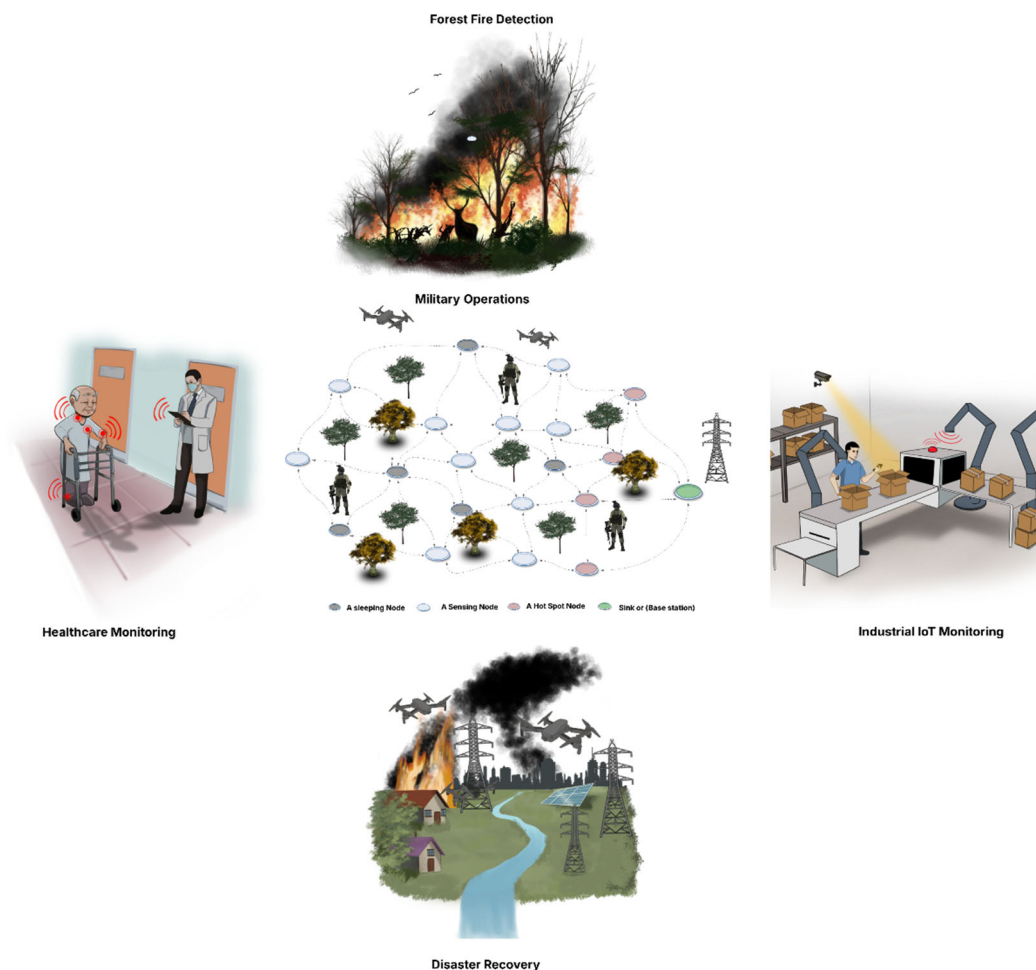


Figure 1. Example of Safety-Critical WSN systems.

To address this challenge, researchers have explored various intelligent scheduling strategies. Many of which are either static, focus on single-objective optimization, or are narrowly designed for specific applications. This highlights the need for a generalizable, adaptive, and intelligent framework that can dynamically select the most suitable scheduling algorithm based on the real-time conditions of the network [9].

The main contributions of this paper are:

1. **Generalizable Selection Framework:** Development of an AI-based adaptive framework capable of evaluating multiple scheduling algorithms and selecting the most suitable one for a specific scenario context.
2. **Multi-Scenario Evaluation:** Application of the framework across five distinct real-world WSN scenarios—healthcare monitoring, military operation, industrial IoT monitoring, forest fire detection, and disaster recovery—to validate its dependability and robustness.
3. **Comparative Insights:** Analysis of the relative strengths of five diverse algorithms—HMM [10], BAT Algorithm [5], Bird Flocking [11], SOFM [12], and LSTM [8]—highlighting both the overall near-optimal performance of each algorithm and specialized their advantages. These algorithms represent a diverse range of approaches, including heuristic, probabilistic, bio-inspired, and neural models [13].

The proposed framework uses feature engineering to capture scenario-specific requirements, such as low latency for emergency alerts or extended lifetime for remote environmental monitoring. These features are mapped against performance profiles of all candidate algorithms, enabling a trained neural network to predict and recommend the optimal choice. Implemented and validated in MATLAB, the framework demonstrates quantitative improvements—such as reduced latency and extended network lifetime, coverage, connectivity—as well as qualitative benefits, including adaptability to evolving conditions and robustness under uncertainty.

The remainder of the paper is organized as follows: Section 2 provides a review of existing node scheduling strategies in WSNs, with emphasis on adaptive algorithm selection methods and prior AI-based approaches. Section 3 details the proposed AI-driven adaptive framework, including its feature engineering process, data normalization, neural network model training, and the algorithm selection mechanism. Section 4 describes the experimental methodology, including the simulation setup, scenario configurations, datasets, algorithm implementations, and performance evaluation metrics. Section 5 presents and discusses the results, comparing the performance of the five candidate algorithms across healthcare monitoring, military operations, industrial IoT monitoring, forest fire detection, and disaster recovery scenarios, and validating the framework's selection accuracy and adaptability. Section 6 summarizes the key findings, discusses the implications for intelligent WSN deployments, and outlines potential directions for future research, including incorporating additional performance objectives such as fault tolerance and security.

2. Related Work

Finding the most suitable node scheduling algorithm for a given WSN scenario remains a significant challenge—particularly in safety-critical systems, where the cost of a fault propagating into a failure is high, and the performance requirements are stringent. These environments demand high service availability, reliability, low latency, and robust fault tolerance, where even minor errors can escalate into system-wide failures [15]. The complexity lies in identifying which scheduling approach will deliver optimal performance under varying operational conditions, such as node density, traffic load, energy utilisation, latency tolerance, and environmental dynamics.

Numerous strategies exist to address this challenge, including duty cycling [16], routing optimization [17], and clustering [18]. Among these, node scheduling algorithms are particularly effective due to their ability to maximize energy savings, maintain network connectivity, and ensure adequate coverage by intelligently controlling when nodes are active or asleep. The literature includes several classical and advanced approaches. For instance, the Randomized Coverage-Based Node Scheduling (RCS) algorithm has been recognized for attempting to balance multiple objectives such as coverage and network lifetime. However, it suffers from several limitations including poor load balancing, unrealistic assumptions, and insufficient reliability under dynamic or safety-critical conditions, making it suitable only for short-term, low-priority deployments such as environmental monitoring [8].

To overcome such challenges, researchers have increasingly turned to Artificial Intelligence (AI) and Machine Learning (ML) frameworks. These models analyze network parameters and adaptively schedule nodes based on predicted conditions. Techniques such as Support Vector Machines (SVM), Reinforcement Learning (RL), and Deep Neural Networks (DNNs) have shown promise in predicting traffic trends and enhancing scheduling efficiency [2][3]. More advanced, context-aware AI systems dynamically adjust to multiple constraints including Quality of Service (QoS) and energy budgets. Adaptive systems like meta-learning agents, policy-based neural networks, and federated learning frameworks can switch strategies in real-time, supporting resilient WSN operation [4][5].

In this context, several AI-based node scheduling algorithms stand out. Perceptron models assist in binary decisions for node activation. HMMs improve over RCS by leveraging probabilistic predictions to manage node transitions, optimize load distribution, and enhance reliability. Their simplified state modeling keeps computational overhead low while improving performance in industrial and IoT environments [11]. On the other hand, bio-inspired algorithms such as BAT and

Bird Flocking use swarm intelligence to avoid local optima and achieve better global scheduling results. These are especially effective in disaster response, military surveillance, and large-scale IoT deployments due to their adaptability and energy efficiency [5].

The SOFM introduce unsupervised learning to cluster sensor nodes based on role or data similarity, thereby improving efficiency in data-heavy applications like forest fire detection and smart grid monitoring. SOFM outperforms bio-inspired methods in highly correlated data environments due to its structured spatial optimization[19]. Meanwhile, the LSTM networks are distinguished by their strength in temporal learning. They model time-dependent behaviors such as traffic load fluctuations or sensor failures, enabling real-time adaptive scheduling and long-term coverage stability. This makes LSTM ideal for safety-critical domains like healthcare monitoring and precision agriculture, where reliability and responsiveness are paramount[20].

Despite these advancements, current approaches typically focus on optimizing a single algorithm for a narrow application domain. This leaves a critical gap: the absence of a unified, generalizable framework capable of selecting the most suitable scheduling algorithm for a given scenario. Without such a framework, network designers must rely on manual tuning or trial-and-error, which is impractical in large-scale or fast-changing deployments.

To address this gap, our work introduces a novel AI-driven adaptive framework that autonomously selects the optimal node scheduling algorithm based on real-time, scenario-specific requirements. The framework evaluates five diverse candidate algorithms—HMM, BAT, Bird Flocking, SOFM, and LSTM—across a range of operational contexts, including healthcare, industrial IoT, forest fire detection, military operations, and disaster recovery. By employing feature engineering to extract key scenario characteristics and mapping them to algorithm performance data, a trained neural network can predict and recommend the best-fit algorithm in real time, without human intervention. This approach bridges the long-standing divide between static algorithm application and adaptive, intelligent scheduling, offering a scalable solution for complex and safety-critical WSN deployments.

3. Proposed AI-Based Selection Framework for WSN

3.1. Framework Architecture

The core challenge lies in identifying the most suitable algorithm to ensure optimal performance in WSN applications. This selection challenge exhibits more complexity when the deployment is for safety-critical WSN applications, where stringent requirements such as high service availability, reliability, low latency, and robust fault tolerance must be consistently met. To effectively address this problem, it is essential to employ AI based framework capable of capturing, identifying and aligning the fundamental specification requirements of a given application scenario. These requirements—such as energy constraints, node density, traffic load, latency tolerance etc., —must be accurately captured and transformed into meaningful input for intelligent decision-making. The architecture of the proposed framework is illustrated in Figure 3, comprising the following core modules:

1. **Input Processing** – Captures deployment scenario parameters, including environmental, network, and application constraints.
2. **Feature Engineering** – Converts parameters into normalized numerical feature vectors for AI processing.
3. **Evaluation and Neural Network Core** – A pre-trained neural network processes feature vectors to determine algorithm suitability.
4. **Selection Logic** – Compares probability scores and selects the algorithm with the highest confidence value.
5. **Output Decision** – Returns the best-fit algorithm along with ranked alternatives for hybrid or fallback use.

3.2. Key Metrics

The framework evaluates algorithms against three key metrics—coverage, connectivity, and network lifetime—which form the foundation for optimal WSN performance. To ensure adaptability for future deployments, the design may also anticipate integration of fault tolerance and security as secondary selection metrics.

3.3. Algorithm Pool

The framework evaluates a pool of node scheduling algorithms widely applied in WSNs:

- **HMM Algorithm** – Probabilistic model suited for temporal state transitions in coverage management.
- **BAT Algorithm** – Bio-inspired metaheuristic optimized for balancing coverage and energy efficiency.
- **Bird Flocking Algorithm** – Swarm-based approach focusing on distributed coordination and connectivity.
- **SOFM Algorithm** – Unsupervised neural clustering for adaptive coverage optimization.
- **LSTM Algorithm** – Deep learning model effective in predicting and adapting to temporal traffic patterns.

3.4. Neural Network Training

The intelligent core of the framework is a neural network with two hidden layers (64 and 32 neurons) and ReLU activation functions [21]. During training, the network learns the mapping between engineered features and the optimal scheduling algorithm by using a dataset that is combined from simulated results.

- Training Process:
 - Dataset: Aggregated from prior deployments and synthetic simulations.
 - Feature Normalization: All input features scaled between 0–1.
 - Cross-Validation: 5-fold validation to ensure generalization.
 - Performance Tracking: Real-time monitoring of accuracy and loss reduction across epochs.
- Inference Process:
 1. Extract features from deployment scenario.
 2. Input normalized vector into neural network.
 3. Generate probability scores for each algorithm.
 4. Select top-ranked algorithm based on confidence score.

The model produces transparent decision outputs via radar plots, showing how each candidate algorithm aligns with scenario requirements.

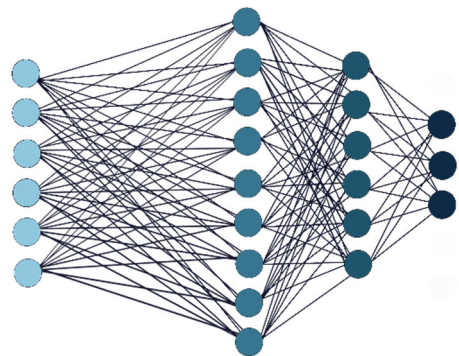


Figure 2. Neural Network Model.

Figure 2 presents the neural network architecture, consisting of an input layer with 9 feature vectors, two hidden layers (64 and 32 neurons using ReLU activation) [18], and an output layer for prediction, consistent with the described training process

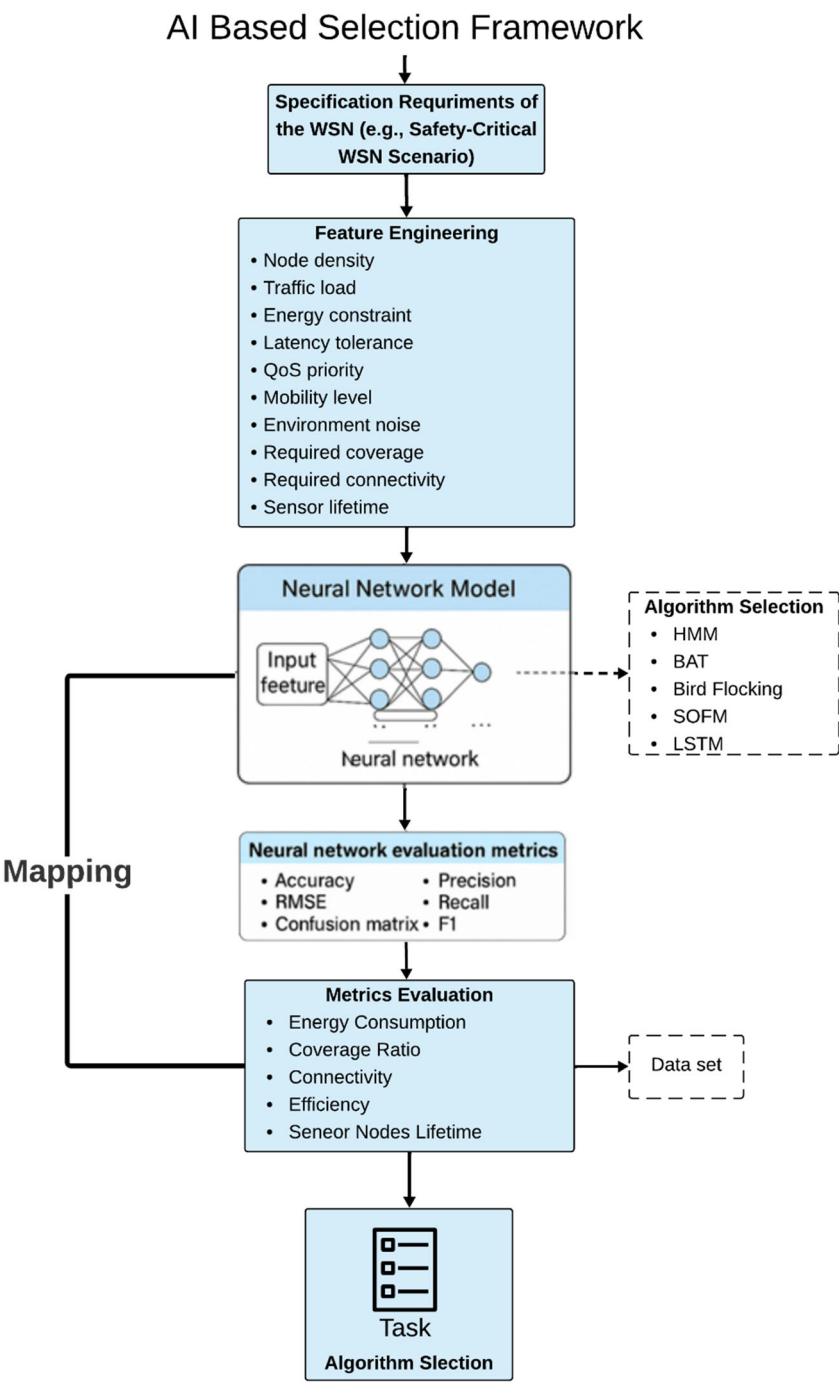


Figure 3. AI Based Framework.

3.5. Selection Process - Pseudocode Summary

The pseudocode below outlines the step-by-step logic for extracting features, running inference, and selecting the best-fit algorithm

Pseudocode. AI Based Framework Using a Neural Network

```
1:  Input: DeploymentScenario
2:  Output: SelectedNodeSchedulingAlgorithm
3:  Begin
4:
5:    // Step 1: Feature extraction function
6:    Function ExtractFeatures(DeploymentScenario):
7:      features = {}
8:      // General specification requirements
9:      features['node_density']      = MeasureNodeDensity(DeploymentScenario)
10:     features['traffic_load']      = MeasureTrafficLoad(DeploymentScenario)
11:     features['energy_constraint']  = MeasureEnergyLevel(DeploymentScenario)
12:     features['latency_tolerance']  =
        MeasureLatencyRequirement(DeploymentScenario)
13:     features['qos_priority']      = EvaluateQoSPriority(DeploymentScenario)
14:     features['mobility_level']    = AssessMobility(DeploymentScenario)
15:     features['environment_noise'] = AssessInterferenceLevel(DeploymentScenario)
16:     features['required_coverage'] = RequiredCoverage(DeploymentScenario)
17:     features['required_connectivity'] = RequiredConnectivity(DeploymentScenario)
18:     features['sensor_lifetime']   = DesiredSensorLifetime(DeploymentScenario)
19:     return Normalize(features)
20:
21:    // Step 2: Load pre-trained neural network model
22:    model = LoadTrainedNeuralNetwork()
23:
24:    // Step 3: Extract features from the given scenario
25:    input_features = ExtractFeatures(DeploymentScenario)
26:
27:    // Step 4: Predict best-fit node scheduling algorithm
28:    prediction_probabilities = model.Predict(input_features)
29:
30:    // Step 5: Select algorithm with highest probability
31:    max_index = ArgMax(prediction_probabilities)
32:    algorithm_list = ['HMM', 'BAT', 'Bird Flocking', 'SOFM', 'LSTM']
33:    SelectedNodeSchedulingAlgorithm = algorithm_list[max_index]
34:
35:    // Step 6: Return the selected algorithm (output top recommendation and ranked
36:    alternatives if needed)
37:    return SelectedNodeSchedulingAlgorithm
End
```

3.6. Mathematical Proof of AI-Based Framework for Node Scheduling Algorithm Selection

This subsection presents a formal mathematical formulation and proof of correctness for the AI-based framework described in the pseudocode. The framework addresses the problem of selecting the most suitable node scheduling algorithm for WSNs application scenarios based on deployment-specific features, using a pre-trained neural network.

Given:

- A deployment scenario S with measurable attributes (features).
- A finite set of scheduling algorithms: $A = \{HMM, BAT, Bird\ Flocking, SOFM, LSTM\}$
- A pre-trained neural network classifier $f_\theta: R^n \rightarrow [0,1]^{|A|}$, where θ are the learned weights, n is the number of features, and:

$$\sum_{i=1}^{|A|} f_\theta(x)_i = 1, f_\theta(x)_i \geq 0 \quad (1)$$

for all $x \in R^x$.

Step 1: Feature Extraction Function

Let $\phi: S \rightarrow R^x$ be the feature extraction mapping from a scenario S to a normalized feature vector:

$$x = \phi(S) = [d, \tau, e, \iota, q, m, \eta, c_r, k_r, L]^T \quad (2)$$

where each term represents a measurable scenario attribute such as node density d , traffic load τ , energy level e , latency requirement ι , QoS priority q , mobility level m , environmental noise/interference η , required coverage c_r , required connectivity k_r , and desired sensor lifetime L .

Claim 1: ϕ is deterministic and well-defined for all $S \in S$, given measurable metrics.

Step 2: Neural Network Decision Rule

The model computes:

$$p = f_\theta(x) = [p_1, p_2, \dots, p_{|A|}]^T \quad (3)$$

where $p_i = P(A_i | x; \theta)$ is the model-estimated probability that A_i is the optimal scheduling algorithm for scenario S .

By construction of the softmax output layer:

$$p_i = \frac{\exp(z_i)}{\sum_{j=1}^{|A|} \exp(z_j)} \quad (4)$$

Where $z = g_\theta(x)$ are the logits from the network.

Step 3: Optimal Selection Criterion

The framework selects:

$$A^* = \arg \max_{A_i \in A} p_i \quad (5)$$

Lemma 1 (MAP Decision Rule):

If p_i are posterior probabilities $P(A_i | x)$, the Maximum A Posteriori (MAP) estimate that minimizes the 0–1 loss is:

$$\hat{A} = \arg \max_{A_i} p_i(A_i | x) \quad (6)$$

Proof: Standard Bayes risk minimization with loss function $L(\hat{A}, A) = I[\hat{A} \neq A]$. Expected risk is minimized by choosing the class with maximal posterior probability.

Step 4: Correctness Proof

Theorem (Framework Correctness):

Given:

1. The neural network f_θ is trained to approximate $P(A_i | x)$ with sufficient accuracy.
2. Feature extraction ϕ produces a consistent feature vector x for any S .
3. The decision rule selects $\arg \max_{A_i} p_i$

Then the algorithm outputs A^* that is optimal under the MAP criterion.

Proof:

1. Input $S \rightarrow x = \phi(S)$ (well-defined by Claim 1).

2. $p = f_{\theta}(x)$ returns posterior probabilities for each A_i (softmax ensures normalization).
3. By Lemma 1, selecting $A^* = \arg \max A_i p_i$ minimizes classification error probability.
4. Therefore, the output A^* is optimal given p and the training objective.

Step 5: Future Extensions in Proof Form

If additional metrics (e.g., fault tolerance, and security) are added, $\varnothing$ can be extended to $\varnothing': S \rightarrow R^{n+k}$. The proof remains valid since the decision rule still corresponds to the MAP estimate over the expanded feature space.

3.7. Use Case of For Forest Fire Detection System

To demonstrate the practical value of the proposed neural network-based framework, we consider its application in a critical real-world scenario: forest fire detection. This domain presents stringent operational requirements, where rapid decision-making, wide-area monitoring, and system resilience are essential. The framework’s ability to evaluate and recommend scheduling algorithms—based on both learned patterns and multi-criteria analysis—makes it particularly suitable for such high-stakes environments. By analyzing the specific needs of forest fire detection systems, we can illustrate how the model supports intelligent algorithm selection, ensuring optimal performance under dynamic and challenging conditions.

In a typical forest fire detection deployment, approximately 2,000 sensor nodes are distributed across a vast area to collect environmental data such as temperature, humidity, gas concentrations, and smoke levels. The scenario requires:

- Alert transmission within 1–2 minutes
- Low power consumption (~6.5W)
- High detection accuracy (>95% recall, >90% precision)
- Wide area coverage with reliable connectivity

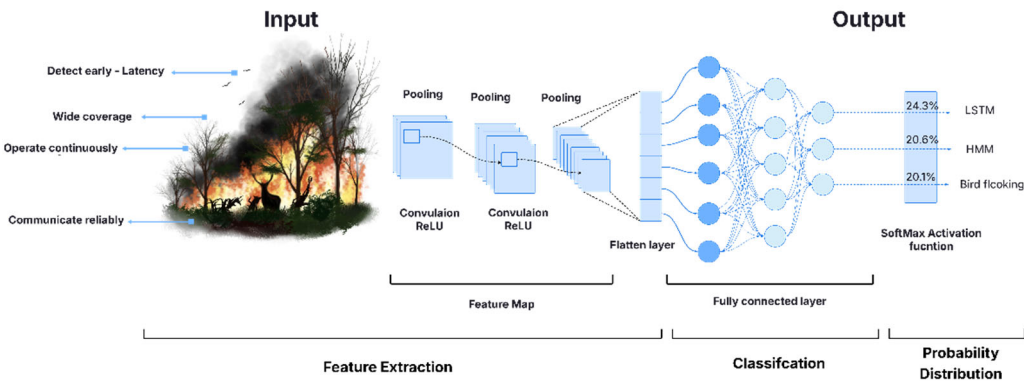


Figure 4. Convolution Neural Network (CNN) in Forest Fire Detection systems.

The process for selecting the optimal solution for the forest fire detection scenario can be summarized as follows:

Step 1: Feature Engineering – Input Vector Creation

The first step in the neural network-based framework involves converting scenario-specific requirements into a structured numerical format suitable for machine learning. This process, known as feature engineering, extracts key characteristics from the given scenario and maps them to normalized values ranging from 0 to 1. These values form the input vector for the neural network.

In the case of a forest fire detection system, the system must balance multiple performance goals such as high coverage, low latency, and long operational lifetime. The relevant features and their corresponding normalized values are outlined in Table 1 below.

Table 1. Feature Engineering Extraction.

Feature Engineering	Description	Normalized Value
Node density	High node coverage needed	0.80
Traffic load	Frequent sensor data transmissions	0.75
Energy constraint	Moderate (must last but still transmit actively)	0.40
Latency tolerance	Very low (alerts must be fast)	0.20
QoS priority	High priority application	1.00
Mobility level	Fixed nodes in terrain	0.00
Environmental noise	Moderate interference (wind, heat)	0.30
Required coverage	Very high (area-wide detection)	0.95
Required connectivity	Very high (always-on communication required)	0.90
Sensor lifetime goal	Moderate-to-high (multi-season)	0.70

These engineered features capture the essential operational demands of the scenario and serve as the input vector for the next step in the framework. The engineered feature vector for this case is: [0.80, 0.75, 0.40, 0.20, 1.00, 0.00, 0.30, 0.95, 0.90, 0.70]

Step 2: Input to Neural Network

The input vector obtained from Table 1, $X = [0.80, 0.75, 0.40, 0.20, 1.00, 0.00, 0.30, 0.95, 0.90, 0.70]$ is fed into a trained neural network that has learned how various feature combinations impact algorithm performance. The network processes the input through multiple layers of neurons using learned weights and activation functions enabling it to model complex relationships between features and algorithm suitability.

The neural network outputs a probability distribution over possible algorithm classes using the neural network model layers as shown in Step 3:

Step 3: Network Output

Table 2. Neural Network Output of the Top Three Recommendation.

Algorithm (Ranks)	Confidence Value	Output Probability
LSTM (Rank 1)	24.3%	✓
HMM (Rank 2)	20.6%	✗
Bird Flocking (Rank 3)	20.1%	✗

This result confirms LSTM’s suitability for scenarios demanding fast response, wide coverage, and stable connectivity—while also indicating that HMM and Bird Flocking may serve as viable secondary or hybrid options.

4. Experimental Setup

4.1. Scenarios Description

The experimental evaluation covers five representative Wireless Sensor Network (WSN) application scenarios, each with unique requirements and operational characteristics:

- **Healthcare Monitoring:** Continuous patient vital signs tracking requiring low latency, high reliability, and extended network lifetime in dynamic environments.
- **Military Operations:** Tactical surveillance with stringent security, connectivity, and rapid adaptability to changing battlefield conditions.
- **Industrial IoT:** Monitoring of manufacturing processes demanding high coverage, fault tolerance, and low energy consumption for prolonged operations.
- **Forest Fire Detection:** Wide-area environmental sensing that prioritizes extensive coverage, stable connectivity, and long network lifetime in remote, energy-constrained conditions.
- **Disaster Recovery:** Emergency response networks requiring quick deployment, robust connectivity, and real-time data transmission under harsh, unpredictable conditions.

These scenarios represent safety-critical use cases where the correct node scheduling algorithm profoundly impacts network performance and application success.

4.2. Simulation Environment

The performance of node scheduling algorithms was analyzed using MATLAB R2018b, a widely used simulation environment offering flexible modeling and visualization capabilities suitable for WSN research. The simulation parameters and datasets were designed to reflect realistic deployments for each scenario, with focus on the following aspects:

- **Network Parameters:** Number of nodes, node density, deployment area, initial energy per node, traffic load, and communication range.
- **Node Behavior:** Scheduling algorithm logic, energy consumption models, and node state transitions.
- **Environmental Factors:** Dynamic changes in network conditions, node failures, and interference patterns modeled according to scenario-specific challenges.
- **Datasets:** Synthetic traffic and environmental data were generated based on typical application behaviors (e.g., temperature and smoke levels for forest fire detection, vital signs for healthcare).

Each algorithm was tested across 30 simulation runs to ensure statistical reliability. The simulation was parameterized to capture performance trends under varying energy budgets, coverage requirements, and network dynamics.

4.3. Performance Metrics

The evaluation focused on critical metrics reflecting the suitability of node scheduling algorithms for the diverse scenarios:

- **Average Connectivity:** Measures the proportion of nodes capable of communicating with each other, indicating network robustness.
- **Maximum and Minimum Coverage:** Represents the best- and worst-case coverage performance, reflecting algorithm behavior at peak and threshold energy levels.
- **Average Lifetime:** The average number of nodes alive per simulation round, signifying energy efficiency and network longevity.
- **Total Number of Zones/Hops:** Represents how nodes are managed spatially; fewer zones imply better connectivity and coordinated coverage.

These metrics collectively characterize the trade-offs between energy consumption, coverage quality, and network reliability.

5. Results and Discussion

This section presents the evaluation outcomes of the proposed AI-driven decision-support framework for selecting optimal node scheduling algorithms in WSNs. Implemented in MATLAB, the framework assesses multiple algorithms—HMM, BAT, SOFM, Bird Flocking, and LSTM—across key performance metrics: network lifetime, coverage, connectivity, and latency.

The decision process combines feature similarity scoring with a trained feedforward neural network, enabling the model to match scenario-specific requirements to algorithmic strengths. The evaluation spans five safety-critical domains: forest fire detection, industrial IoT monitoring, healthcare systems, military operations, and disaster recovery. By integrating metric-based ranking with machine learning-driven predictions, the framework provides data-driven, transparent, and adaptive recommendations suitable for both predefined and novel operational contexts.

5.1. Forest Fire Detection

In this scenario, rapid detection, wide coverage, and stable connectivity are essential for monitoring remote areas. The model evaluated inputs reflecting early fire identification, minimal latency, and prolonged network lifetime. Using a dataset, the framework identified LSTM as the top-performing algorithm (confidence score: 24.3%), surpassing HMM (20.6%) and Bird Flocking (20.1) (Figure 5).

The radar chart highlights the scenario’s priorities: coverage (~0.85), low-hop relay (~0.9), lifetime (~0.75), and connectivity (~0.7). LSTM’s ability to capture temporal dependencies in sensor data enables early anomaly detection and dynamic scheduling, yielding superior energy efficiency and reduced latency—critical for dependable wildfire monitoring across vast, infrastructure-limited areas.

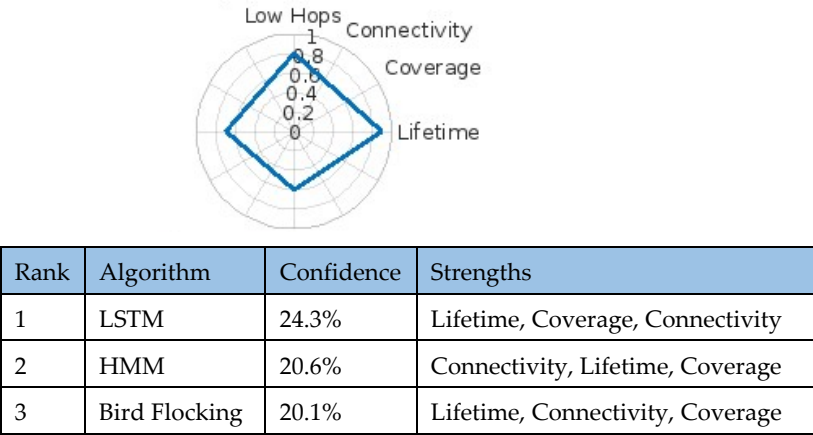


Figure 5. Forest fire Detection System Scenario Results with a Rader Chart.

5.2. Industrial IoT Monitoring

Here, long-term uptime, broad coverage across facilities, and fault-tolerant connectivity are key. The model processed priorities including battery life optimization, reliable throughput, and resilience in challenging industrial conditions. LSTM again ranked highest (23.4%), ahead of HMM (21.8%) and Bird Flocking (20.5) (Figure 6).

Radar chart analysis shows emphasis on lifetime (0.9), coverage (0.85), and connectivity (0.75). LSTM’s predictive modeling of time-series data supports proactive maintenance scheduling and adaptive load balancing, making it well-suited for continuous industrial monitoring.

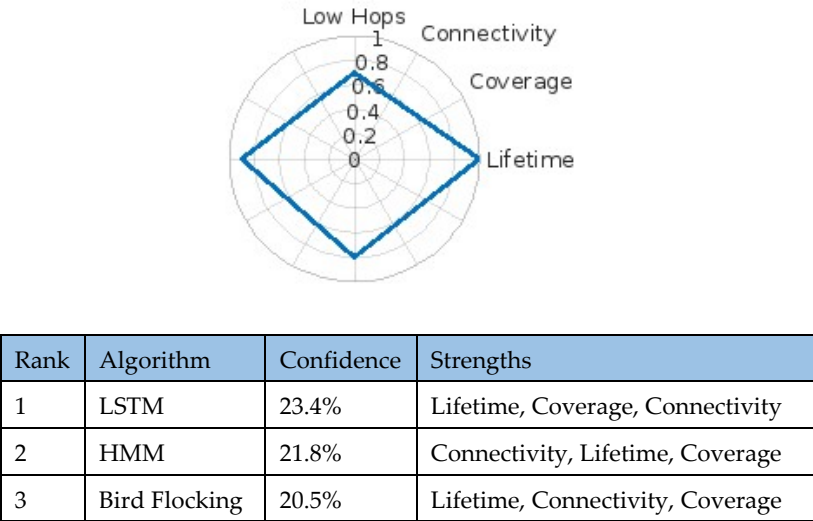


Figure 6. Industrial IoT Scenario Results with a Rader Chart.

5.3. Healthcare Monitoring

For healthcare systems—where low-latency transmission, prolonged device operation, and resilient connectivity directly affect patient safety—the framework prioritized low-hop routing (~0.9), extended lifetime (~0.8), and strong connectivity (~0.7). LSTM led with 23.6%, outperforming HMM (21.4%) and Bird Flocking (20.1) (Figure 7).

Its temporal modeling capability enables proactive communication load management, adaptive scheduling during patient movement, and uninterrupted monitoring for applications like vital-sign tracking and arrhythmia detection.

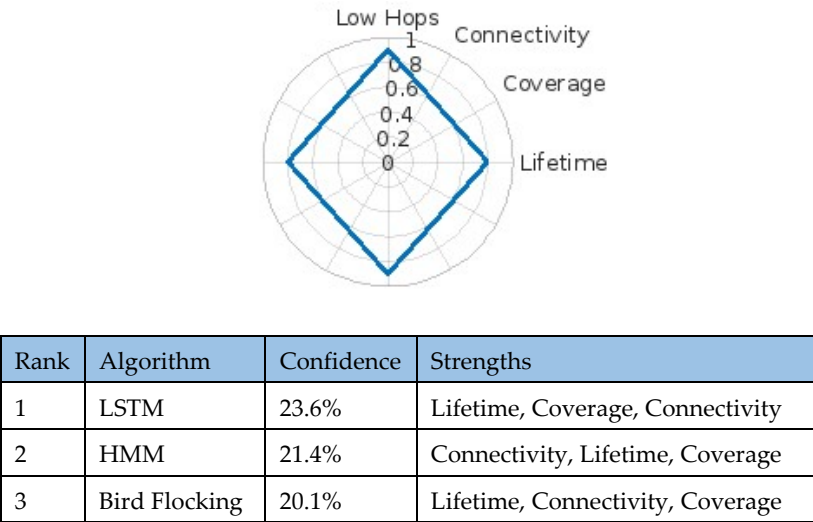


Figure 7. Healthcare Scenario Results with a Rader Chart.

5.4. Military Operations

Military environments demand robust connectivity, minimal latency, and adaptability to dynamic topologies. Priorities included low-hop routing (~0.8), connectivity (~1.0), and lifetime (~0.75). LSTM achieved the top score (23.4%), followed by HMM (21.8%) and Bird Flocking (20.5) (Figure 8).

LSTM’s strengths in temporal modelling enhance real-time routing, resilient communications under jamming, and adaptive scheduling during topology shifts—critical for sustained operational readiness.

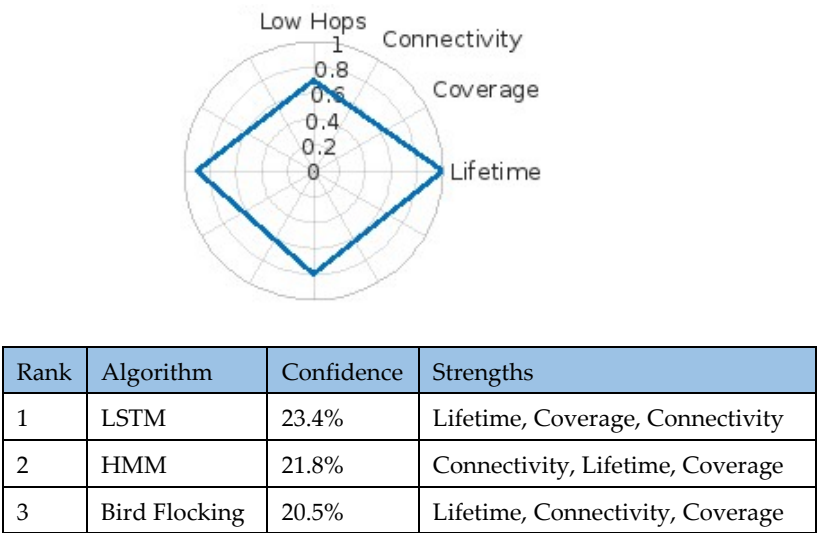


Figure 8. Military Scenario Results with a Rader Chart.

5.5. Disaster Recovery

In disaster response, extended lifetime, wide coverage, and low-hop routing (~0.8) are vital for supporting dispersed teams in damaged environments. LSTM scored 22.8%, outperforming Bird Flocking (21.7%) and HMM (21.6%) (Figure 9).

Its sequence modelling predicts node failures, reroutes data dynamically, and sustains coverage under infrastructure loss, ensuring timely coordination for first responders.

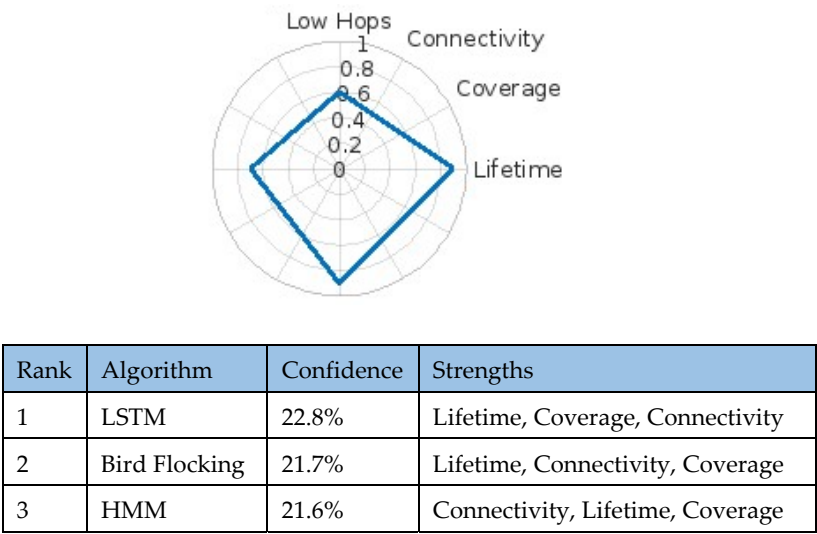


Figure 9. Disaster recovery Scenario Results with a Rader Chart.

5.6. Cross-Scenario Insights

Across all five domains, LSTM consistently delivered the highest or near-highest confidence scores, demonstrating strong adaptability, latency reduction, and network lifetime extension. While HMM excelled in real-time connectivity and Bird Flocking in operational resilience, LSTM’s temporal modeling offered the most balanced performance.

The AI-driven framework—validated through cross-validation to prevent overfitting—proved effective in generalizing to diverse, evolving conditions. By systematically aligning algorithm choice with scenario-specific priorities, it enhances resilience, reduces latency, and ensures sustained operational uptime.

Ultimately, this approach advances autonomous WSN scheduling, supporting rapid decision-making and mission success in complex, high-stakes environments.

6. Conclusions

This study presents a novel AI-driven framework designed to optimize node scheduling algorithm selection for diverse WSN applications. Recognizing that no single algorithm is universally optimal, the framework evaluates key requirements—such as coverage, connectivity, and network lifetime—to intelligently match scenarios with the most suitable scheduling strategies. Validated across mission-critical domains including forest fire detection, industrial IoT monitoring, healthcare monitoring, military operations, and disaster recovery, the LSTM algorithm consistently achieved near-optimal performance. Its dynamic modelling of temporal dependencies led to notable gains in latency, energy efficiency, and fault tolerance. Complementary algorithms like HMM, Bird Flocking, BAT, and SOFM contributed strengths in network lifetime, coverage, and connectivity. The framework's strength lies in combining accuracy with automation, leveraging a trained neural network enhanced by rigorous cross-validation to ensure robust generalization and prevent overfitting. By integrating a diverse algorithm set—from biologically inspired models to recurrent networks—this data-driven approach enhances WSN scheduling dependability, reliability, and scalability. Ultimately, the framework enables intelligent, efficient, and context-aware decision-making, with important implications for the resilience and performance of mission-critical networks.

Future research will focus on expanding the framework's capabilities by integrating additional scheduling algorithms to capture a wider range of operational contexts. Incorporating more scenarios and new selection metrics—such as fault tolerance, and security features, will enable a more holistic evaluation of algorithm suitability, critical for increasingly complex and hostile environments. Furthermore, automating decision thresholds through dynamic, data-driven functions will enhance real-time adaptability and responsiveness. Together, these advances will move the framework closer to fully autonomous, self-optimizing WSN deployments, capable of sustaining resilient, efficient, and secure communications in diverse and evolving scenarios.

Acknowledgment: The authors would like to thank all those who contributed to the completion and success of this work. Special thanks to the Faculty of Science and Technology at Middlesex University who played a significant role in backing this work at all stages of the study. Extended thanks go to Dana Elnader, Faculty of Studio Art, University of Guelph.

References

1. O. Chipara, 'Towards real-time wireless sensor networks', 2010, Accessed: June 21, 2015. [Online]. Available: <http://openscholarship.wustl.edu/etd/62/>
2. M. Rausand, *Reliability of safety-critical systems: theory and applications*. John Wiley & Sons, 2014.
3. A. Meier, 'Safety-Critical Wireless Sensor Networks', SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, 2009. Accessed: June 27, 2015. [Online]. Available: <ftp://ftp.tik.ee.ethz.ch/pub/people/andreama/papers/meier2009diss.pdf>
4. K. Sohraby, D. Minoli, and T. Znati, *Wireless sensor networks: technology, protocols, and applications*. John Wiley & sons, 2007.
5. I. Al-Nader, A. Lasebae, R. Raheem, and G. Ekembe Ngondi, 'A Novel Bio-Inspired Bat Node Scheduling Algorithm for Dependable Safety-Critical Wireless Sensor Network Systems', *Sensors*, vol. 24, no. 6, p. 1928, 2024, Accessed: May 21, 2024. [Online]. Available: <https://www.mdpi.com/1424-8220/24/6/1928>
6. D. Masti, D. Bernardini, and A. Bemporad, 'Learning virtual sensors for estimating the scheduling signal of parameter-varying systems', in *2019 27th Mediterranean Conference on Control and Automation (MED)*, IEEE, 2019, pp. 232–237.
7. Gunjan, 'A Review on Multi-objective Optimization in Wireless Sensor Networks Using Nature Inspired Meta-heuristic Algorithms', *Neural Process. Lett.*, vol. 55, no. 3, pp. 2587–2611, June 2023, doi: 10.1007/s11063-022-10851-4.
8. I. Al-Nader, A. Lasebae, R. Raheem, and A. Khoshkholghi, 'A Novel Scheduling Algorithm for Improved Performance of Multi-Objective Safety-Critical Wireless Sensor Networks Using Long Short-Term

- Memory', *Electronics*, vol. 12, no. 23, p. 4766, 2023, Accessed: Nov. 28, 2023. [Online]. Available: <https://www.mdpi.com/2079-9292/12/23/4766>
9. L. Wang, R. Wei, Y. Lin, and B. Wang, 'A clique base node scheduling method for wireless sensor networks', *J. Netw. Comput. Appl.*, vol. 33, no. 4, pp. 383–396, 2010.
 10. I. Alnader, A. Lasebae, and R. Raheem, 'Using Hidden Markov Chain for Improving the Dependability of Safety-Critical WSNs', in *International Conference on Advanced Information Networking and Applications*, Springer, 2023, pp. 460–472.
 11. I. Al-Nader, R. Raheem, and A. Lasebae, 'A Novel Bio-Inspired Bird Flocking Node Scheduling Algorithm for Dependable Safety-Critical Wireless Sensor Network Systems', *J*, vol. 8, no. 2, p. 19, 2025, Accessed: Aug. 14, 2025. [Online]. Available: <https://www.mdpi.com/2571-8800/8/2/19>
 12. I. Al-Nader, A. Lasebae, and R. Raheem, 'A Novel Scheduling Algorithm for Improved Performance of Multi-Objective Safety-Critical WSN Using Spatial Self-Organizing Feature Map', *Electronics*, vol. 13, no. 1, p. 19, 2023, Accessed: Dec. 30, 2023. [Online]. Available: <https://www.mdpi.com/2079-9292/13/1/19>
 13. I. Al-Nader, A. Lasebae, and R. Raheem, 'A New Perceptron-Based Neural-Network Algorithm to Enhance the Scheduling Performance of Safety-Critical WSNs of Increased Dependability', in *Information Systems for Intelligent Systems*, vol. 379, C. So In, N. D. Londhe, N. Bhatt, and M. Kitsing, Eds, in Smart Innovation, Systems and Technologies, vol. 379. , Singapore: Springer Nature Singapore, 2024, pp. 347–366. doi: 10.1007/978-981-99-8612-5_28.
 14. A. P. Fishman and C. P. LYMAN, 'Hibernation in mammals', *Circulation*, vol. 24, no. 2, pp. 434–445, 1961, Accessed: July 18, 2016. [Online]. Available: <http://circ.ahajournals.org/content/24/2/434.short>
 15. F. Sailhan, T. Delot, A. Pathak, A. Puech, and M. Roy, 'Dependable wireless sensor networks', in *3th workshop Gestion des Données dans les Systèmes d'Information Pervasifs (GEDSIP) in conjunction with INFORSID*, 2009, pp. 1–16. Accessed: June 21, 2015. [Online]. Available: <http://cedric.cnam.fr/~sailhanf/publications/gedsip.pdf>
 16. S. Lai, 'Duty-Cycled Wireless Sensor Networks: Wakeup Scheduling, Routing, and Broadcasting', Virginia Polytechnic Institute and State University, 2010. Accessed: Sept. 23, 2015. [Online]. Available: <http://scholar.lib.vt.edu/theses/available/etd-04262010-160152/>
 17. V. Narayan, A. K. Daniel, and P. Chaturvedi, 'E-FEERP: Enhanced Fuzzy Based Energy Efficient Routing Protocol for Wireless Sensor Network', *Wirel. Pers. Commun.*, vol. 131, no. 1, pp. 371–398, July 2023, doi: 10.1007/s11277-023-10434-z.
 18. S.-G. Jung, S. Yeom, M. H. Shon, D. S. Kim, and H. Choo, 'Clustering Wireless Sensor Networks Based on Bird Flocking Behavior', in *Computational Science and Its Applications -- ICCSA 2015*, vol. 9158, O. Gervasi, B. Murgante, S. Misra, M. L. Gavrilova, A. M. A. C. Rocha, C. Torre, D. Taniar, and B. O. Aduhan, Eds, in Lecture Notes in Computer Science, vol. 9158. , Cham: Springer International Publishing, 2015, pp. 128–137. doi: 10.1007/978-3-319-21410-8_10.
 19. H. Cheng, Z. Xie, L. Wu, Z. Yu, and R. Li, 'Data prediction model in wireless sensor networks based on bidirectional LSTM', *EURASIP J. Wirel. Commun. Netw.*, vol. 2019, no. 1, p. 203, Aug. 2019, doi: 10.1186/s13638-019-1511-4.
 20. S. Jeyalakshmi and R. M. Ganesh, 'Adaptive Duty-Cycle Scheduling using Bi-Directional Long Short-Term Memory (BiLSTM) for Next Generation IoT Applications', in *2022 International Conference on Computing, Communication and Power Technology (IC3P)*, IEEE, 2022, pp. 75–80. Accessed: Oct. 05, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9793496/>
 21. S. Salmi and L. Oughdir, 'Cnn-lstm based approach for dos attacks detection in wireless sensor networks', *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, 2022.
 22. A. K. Dohare, 'A CNN and LSTM-based Data Prediction Model for WSN', in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, IEEE, 2021, pp. 1327–1330.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s)

disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.