electronics

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

Faculty of Science and Technology, Department of Computer Science, Hendon Campus, Middlesex University, The Burroughs, London NW4 4BT, UK; ia287@live.mdx.ac.uk (I.A.-N.); a.lasebae@mdx.ac.uk (A.L.)
* Correspondence: r.h.raheem@mdx.ac.uk

Abstract

Wireless Sensor Networks (WSNs) are vital to a wide range of applications, spanning from environmental monitoring to safety-critical systems. Ensuring dependable operation in these networks critically depends on selecting an optimal node scheduling algorithm; however, this remains a major challenge since no single approach performs best under all conditions. To address this issue, this paper proposes an AI-driven framework that evaluates scenario-specific functional requirements—such as coverage, connectivity, and network lifetime—to identify the optimal node scheduling algorithm from a pool that includes Hidden Markov Models (HMMs), BAT, Bird Flocking, Self-Organizing Maps (SOFMs), and Long Short-Term Memory (LSTM) networks. The framework was evaluated using a neural network trained on simulated data and tested across five real-world scenarios: healthcare monitoring, military operations, industrial IoT, forest fire detection, and disaster recovery. The results clearly demonstrate the effectiveness of the proposed framework in identifying the most suitable algorithm for each scenario. Notably, the LSTM algorithm frequently achieved near-optimal performance, excelling in critical objectives such as network lifetime, connectivity, and coverage. The framework also revealed the complementary strengths of other algorithms—HMM proved superior for maintaining connectivity, while Bird Flocking excelled in extending network lifetime. Consequently, this work validates that a scenario-aware selection strategy is essential for maximizing WSN dependability, as it leverages the unique advantages of diverse algorithms.

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

check for updates

Academic Editors: Rajan Kadel, Assefa Kassa Teshome and Samar Shailendra

Received: 15 August 2025
Revised: 14 October 2025
Accepted: 20 October 2025
Published: 27 October 2025

Citation: Al-Nader, I.; Raheem, R.; Lasebae, A. A Novel Adaptive AI-Based Framework for Node Scheduling Algorithm Selection in Safety-Critical Wireless Sensor Networks. *Electronics* **2025**, *14*, 4198. https://doi.org/10.3390/electronics14214198

1. Introduction

WSNs have emerged as a fundamental infrastructure for IoT technology, enabling diverse applications from environmental monitoring to critical systems in healthcare, defence, and industrial automation [1,2]. In these systems, optimizing key performance metrics—including energy consumption, network lifetime, coverage, and connectivity—becomes paramount for ensuring dependable operation, e.g., service availability and reliability [3,4]. Among various optimization techniques, sleep/wake scheduling has proven particularly effective for energy conservation, allowing nodes to alternate between active and low-power states while maintaining essential network functionalities [5].

Selecting the most suitable node 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 selection solution.

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 solution that can dynamically select the most suitable node scheduling algorithm based on the dynamic conditions of the network [8].

The main contributions of this study are summarized as follows:

- AI-Based Framework for Dynamic Algorithm Selection: This work proposes a novel AI-driven framework that dynamically selects the most suitable node scheduling algorithm—chosen from a pool including HMM [9], BAT Algorithm [4], Bird Flocking [10], SOFM [11], and LSTM [7]—in real time, based on scenario-specific conditions.
- Neural Network–Based Decision Mechanism: A dedicated neural network model is developed to serve as the core decision engine [12]. The model is trained to evaluate complex scenario requirements—balancing coverage, connectivity, and network lifetime—and to intelligently recommend the optimal algorithm, thereby enhancing overall system dependability.
- Comprehensive Validation Across Safety-Critical Domains: The proposed framework is rigorously validated across five application scenarios: healthcare monitoring, military operations, industrial IoT, forest fire detection, and disaster recovery. The results demonstrate the framework's generality, robustness, and effectiveness in meeting diverse reliability and performance requirements.

The proposed AI-based framework uses feature engineering to capture scenario-specific requirements, such as low latency for emergency alerts or extended lifetime for remote environmental monitoring. The features are mapped to the performance profiles of various candidate algorithms—such as HMM, SOFM, LSTM, and other bio-inspired approaches [13]—allowing the trained neural network to identify and recommend the optimal option. Implemented and validated in MATLAB (version R2025b), 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.

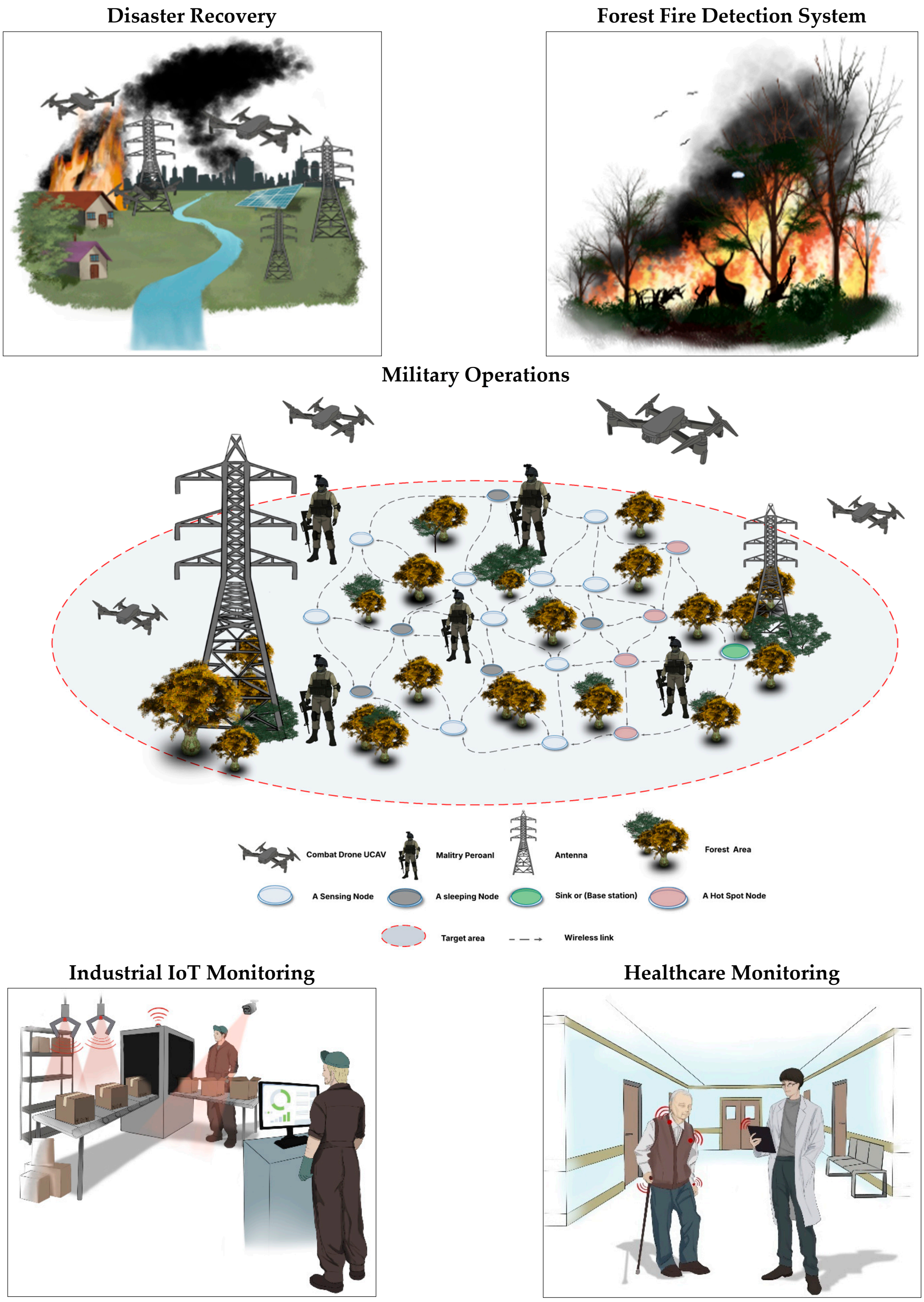


Figure 1. Representative safety-critical scenarios where WSNs play a vital role in ensuring reliable monitoring and control.

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 [14]. 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 [15], routing optimization [16], and clustering [17]. 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 introduced 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 [18]. 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 [1].

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.

Summary and Comparative Analysis

To strengthen the contextualization of this study, Table 1 summarizes the main existing approaches to node scheduling in WSNs, classified by their core optimization strategy, adaptability, target application domain, and primary limitations. The analysis highlights that most prior works are application-specific, static, or single-objective, lacking the adaptability required for dependable performance in safety-critical and dynamic environments. In contrast, the present study introduces a generalized, AI-driven adaptive framework capable of selecting the most suitable scheduling algorithm across diverse operational contexts.

Table 1. Comparative summary of representative node scheduling algorithms and frameworks in WSNs.

Reference	Technique/Model	Optimization Focus	Adaptability	Target Application	Main Limitations
[8] Wang et al., 2010	Clique-based node scheduling	Coverage, network lifetime	Static	General WSN	Limited scalability; lacks adaptive intelligence
[9] Al-Nader et al., 2023	HMM	Connectivity, reliability	Moderate	Industrial IoT	Limited temporal learning; fixed transition thresholds
[4] Al-Nader et al., 2024	BAT Algorithm (bio-inspired)	Energy efficiency, coverage	Low	Safety-critical WSN	Slower convergence; risk of local optima
[10] Al-Nader et al., 2025	Bird Flocking (swarm-based)	Lifetime, connectivity	Moderate	Military, disaster response	Sensitive to topology changes; parameter tuning required
[11] Al-Nader et al., 2023	SOFM	Spatial optimization, data clustering	Low	Environmental monitoring	Unsupervised; lacks temporal awareness
[1] Jeyalaksshmi et al., 2022	BiLSTM deep learning	Temporal prediction, adaptive duty cycling	High	Smart healthcare	Scenario-specific; limited generalization
[2] Haseeb et al., 2020	AI-based IoT framework	Energy efficiency, security	Moderate	Smart agriculture	Fixed architecture; not algorithm-agnostic
[6] Gunjan, 2023	Metaheuristic optimization	Multi-objective (coverage, energy)	Moderate	General WSN	Static optimization; no real-time learning
Proposed Framework (This Work)	**Neural Network–based algorithm selector**	**Coverage, lifetime, connectivity (multi-objective)**	**High (real-time, scenario-aware)**	**Cross-domain (healthcare, IoT, military, disaster recovery)**	**Bridges multiple algorithms; fully adaptive and generalizable**

This comparative overview positions the proposed framework as a unique contribution that integrates AI-driven decision-making to autonomously identify the optimal node scheduling algorithm for a given scenario. By bridging diverse algorithmic paradigms—

from probabilistic to bio-inspired and deep learning models, the framework achieves high adaptability and performance consistency across multiple safety-critical domains.

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. Accurately capturing and translating these requirements—such as energy constraints, node density, traffic load, and latency tolerance—into actionable input facilitates intelligent decision-making within the proposed solution. The core architecture is built upon a neural network model as shown in Figure 2 and integrated into a comprehensive framework, with its functional modules thoroughly depicted in Figure 3:

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 [19]. 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 presents the neural network architecture, consisting of an input layer with 6 feature vectors, two hidden layers (64 and 32 neurons using ReLU activation) [18], and a predictive output layer, is designed to learn the complex relationships within WSN data. Building directly upon this, Figure 3 outlines the comprehensive, structured framework for selecting optimal node scheduling algorithms. In this overarching process, the neural network from Figure 2 constitutes the critical "Neural Network Model" block. The framework begins by defining the network's specification requirements, which are processed through a feature engineering step to create the input vectors for the model. The trained neural network then processes these features, and its performance is rigorously evaluated using key WSN-specific metrics (e.g., Energy Consumption, Coverage). Finally, the validated model's predictions are used to inform the selection of the most suitable algorithm from a predefined set, thereby completing the end-to-end intelligent selection system.

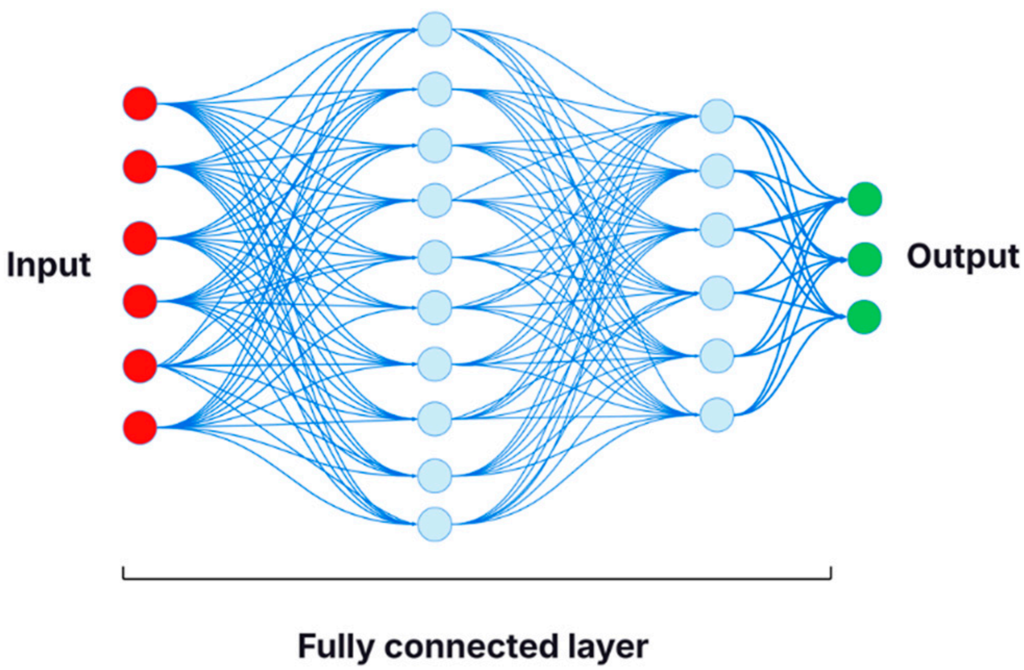


Figure 2. Structure of the neural network model forming the core of the proposed AI-based adaptive framework. Red node: Input layer; light blue nodes: Hidden neurons (fully connected layers); green nodes: Output layer.

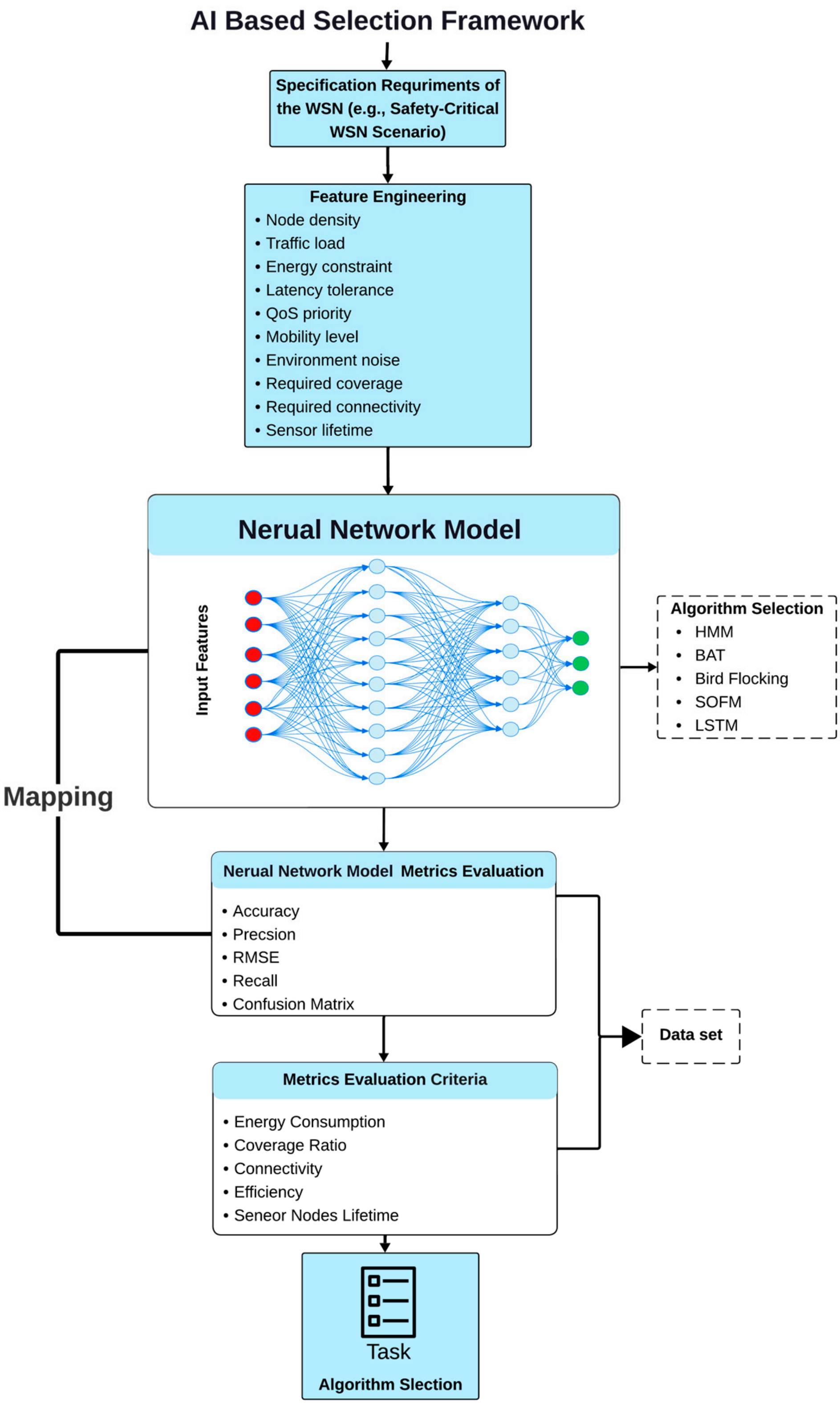


Figure 3. The Proposed AI-Based Framework.

3.5. Selection Process-Pseudocode Summary

This logical flow, defined by the framework in Figure 3 and powered by the neural network architecture from Figure 2, is translated into an executable sequence of operations. The pseudocode below outlines the step-by-step logic for extracting features, running inference, and selecting the best-fit algorithm (Algorithm 1).

Algorithm 1. Pseudocode. AI Based Framework Using a Neural Network

```
Input: DeploymentScenario
Output: SelectedNodeSchedulingAlgorithm
Begin

  // Step 1: Feature extraction function
  Function ExtractFeatures(DeploymentScenario):
    features = {}
    // General specification requirements
    features['node_density'] = MeasureNodeDensity(DeploymentScenario)
    features['traffic_load'] = MeasureTrafficLoad(DeploymentScenario)
    features['energy_constraint'] = MeasureEnergyLevel(DeploymentScenario)
    features['latency_tolerance'] = MeasureLatencyRequirement(DeploymentScenario
    features['qos_priority'] = EvaluateQoSPriority(DeploymentScenario)
    features['mobility_level'] = AssessMobility(DeploymentScenario)
    features['environment_noise'] = AssessInterferenceLevel(DeploymentScenario)
    features['required_coverage'] = RequiredCoverage(DeploymentScenario)
    features['required_connectivity'] = RequiredConnectivity(DeploymentScenario)
    features['sensor_lifetime'] = DesiredSensorLifetime(DeploymentScenario)
    return Normalize(features)

  // Step 2: Load pre-trained neural network model
  model = LoadTrainedNeuralNetwork()

  // Step 3: Extract features from the given scenario
  input_features = ExtractFeatures(DeploymentScenario)

  // Step 4: Predict best-fit node scheduling algorithm
  prediction_probabilities = model.Predict(input_features)

   // Step 5: Select algorithm with highest probability
  max_index = ArgMax(prediction_probabilities)
  algorithm_list = ['HMM', 'BAT', 'Bird Flocking', 'SOFM', 'LSTM']
  SelectedNodeSchedulingAlgorithm = algorithm_list[max_index]

  // Step 6: Return the selected algorithm (output top recommendation and
ranked alternatives if needed)
  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 WSN 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 $f_\theta : R^n \to [0,1]^{|A|}$, where θ are the learned weights, n is the number of features, and:

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

for all $x \in R^x$.

Step 1: Feature Extraction Function

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

$$x = \varnothing(S) = [d, \tau, e, \iota, q, m, \eta, c_r, k_r, L]^T \tag{2}$$

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. *$\varnothing$ is deterministic and well-defined for all$S \in S$, given measurable metrics.*

Step 2: Neural Network Decision Rule

The model computes:

$$p = f_\theta(x) = \left[p_1,\ p_2,\ \ldots, p_{|A|}\right]^T \tag{3}$$

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

By construction of the SoftMax output layer:

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

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

Step 3: Optimal Selection Criterion

The framework selects:

$$A^* = arg \max_{A_i \in A} p_i \tag{5}$$

Lemma 1 (MAP Decision Rule). *If p_i are posterior probabilities $P(A_i \mid x)$, the Maximum A Posteriori (MAP) estimate that minimizes the 0–1 loss is:*

$$\hat{A} = arg \max_{A_i} p_i\ (A_i \mid x) \tag{6}$$

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

Step 4: Correctness Proof

Theorem 1 (Framework Correctness)**.**

Given:

1. The neural network$f_\theta$is trained to approximate $P\ (A_i \mid x)$ with sufficient accuracy.

2. Feature extraction ∅ *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 → x = ∅ (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, ∅ can be extended to ∅′ : S → 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 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 2000 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 min
- Low power consumption (~6.5 W)
- High detection accuracy (>95% recall, >90% precision)
- Wide area coverage with reliable connectivity

Figure 4 shows the process for selecting the optimal node scheduling solution for the forest fire detection scenario which can be summarized as follows:

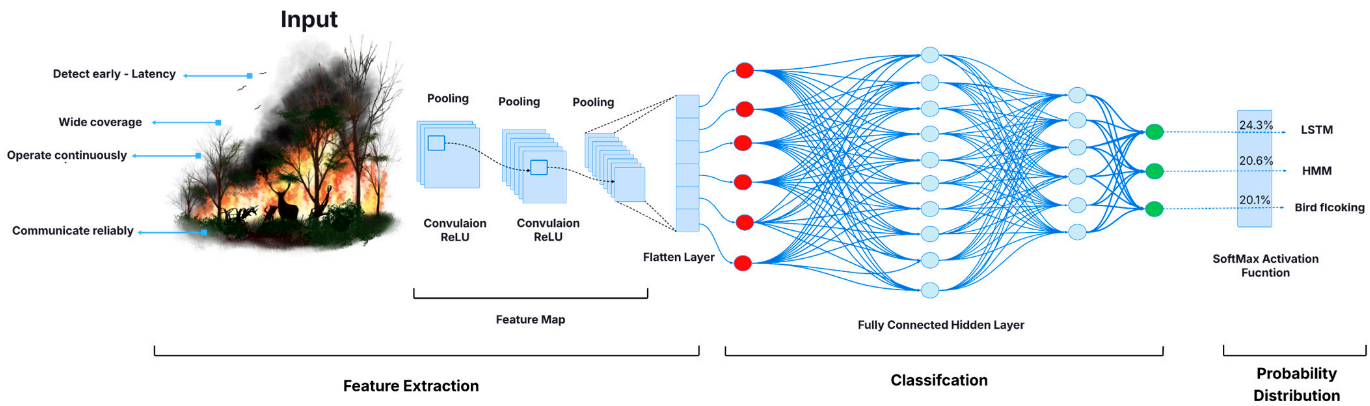


Figure 4. The Use of the Neural Network in Forest Fire Detection systems.

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 2 below.

Table 2. 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 2, 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

In the forest fire detection use case, the AI-based framework identified the LSTM node scheduling algorithm as the optimal selection, achieving superior performance as presented in Table 3.

Table 3. 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%	✖

The first symbol (✔) indicates the optimal match, representing the algorithm selected for the forest fire detection system scenario. Conversely, the cross symbol (✖) denotes algorithms that were excluded due to their lower confidence values.

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 and Simulation Environment

The experimental evaluation of the proposed AI-based framework was conducted in a MATLAB-based simulation environment to ensure controlled, repeatable, and realistic testing conditions for WSNs. The framework integrates five previously published node scheduling algorithms developed by the authors—HMM, BAT, SOFM, Bird Flocking, and LSTM [4,7,9–11]—each designed to optimize distinct dependability objectives. It is worth mentioning that 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. Statistical analysis using ANOVA confirmed significant performance differences among algorithms ($p < 0.05$). To improve model generalization and simulate realistic performance fluctuations, 1000 synthetic training samples were generated by introducing Gaussian noise (mean = 0, $\sigma = 0.1$) within the normalized [0, 1] range.

The neural network model used in this study consists of six input nodes, corresponding to the selected WSN metrics, two hidden layers (64 and 32 neurons with ReLU activation), and a three-node SoftMax output layer representing the algorithm classes. Training was performed using the Adam optimizer over 50 epochs, with a mini-batch size of 64 and 5-fold cross-validation to ensure robust generalization. The study does not introduce new datasets but leverages validated simulation data from previous works to identify the most suitable node scheduling algorithm for specific WSN deployment scenarios.

Scenarios Description

The experimental evaluation covers five representative WSN application scenarios as shown in Figure 1, 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.

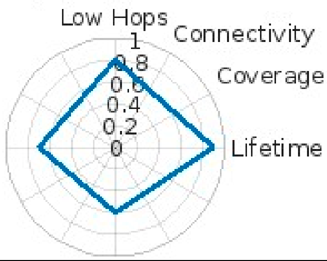
5. Results and Discussion

The AI-based framework evaluates five algorithms—HMM, BAT, SOFM, Bird Flocking, and LSTM—across three key performance metrics: network lifetime, coverage, and connectivity.

The decision process combines feature-based similarity scoring with a trained feedforward neural network, allowing 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 emerging 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 parameters reflecting early fire identification, minimal latency, and prolonged network lifetime. The framework identified LSTM as the top-performing algorithm (confidence score: 24.3%), followed by HMM (20.6%) and Bird Flocking (20.1%) (Figure 5).



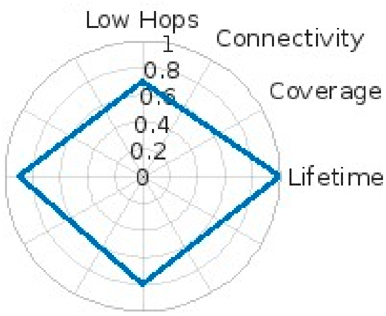
Rank	Algorithm	Confidence	Strengths
1	LSTM	24.3%	Lifetime, Coverage, Connectivity
2	HMM	20.6%	Connectivity, Lifetime, Coverage
3	Bird Flocking	20.1%	Lifetime, Connectivity, Coverage

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

The radar chart highlights key 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, achieving superior energy efficiency and reduced latency—critical for dependable wildfire monitoring in large, infrastructure-limited regions.

5.2. Industrial IoT Monitoring

In industrial IoT environments, long-term uptime, broad coverage, and fault-tolerant connectivity are critical. The model emphasized parameters such as battery life optimization, throughput reliability, and resilience under fluctuating network loads. LSTM again ranked highest (23.4%), outperforming HMM (21.8%) and Bird Flocking (20.5%) (Figure 6).



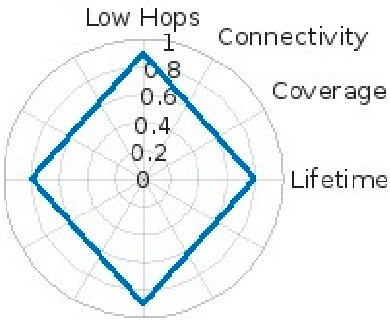
Rank	Algorithm	Confidence	Strengths
1	LSTM	23.4%	Lifetime, Coverage, Connectivity
2	HMM	21.8%	Connectivity, Lifetime, Coverage
3	Bird Flocking	20.5%	Lifetime, Connectivity, Coverage

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

Radar chart analysis emphasizes 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 particularly effective for continuous industrial monitoring.

5.3. Healthcare Monitoring

For healthcare applications—where low latency, extended sensor lifetime, and reliable connectivity directly impact patient safety—the framework prioritized low-hop routing (~0.9), long lifetime (~0.8), and strong connectivity (~0.7). LSTM achieved the highest confidence score (23.6%), outperforming HMM (21.4%) and Bird Flocking (20.1%) (Figure 7).



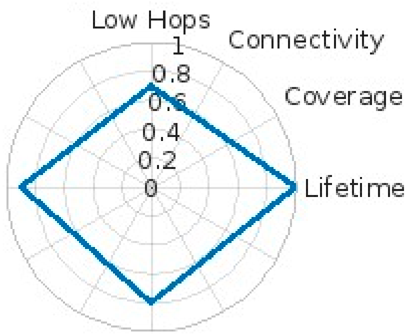
Rank	Algorithm	Confidence	Strengths
1	LSTM	23.6%	Lifetime, Coverage, Connectivity
2	HMM	21.4%	Connectivity, Lifetime, Coverage
3	Bird Flocking	20.1%	Lifetime, Connectivity, Coverage

Figure 7. Healthcare Scenario Results with a Radar Chart.

LSTM's temporal modeling enables proactive communication load management, adaptive scheduling during patient mobility, and uninterrupted monitoring—ideal for applications such as continuous vital-sign tracking and arrhythmia detection.

5.4. Military Operations

Military applications demand robust connectivity, minimal latency, and adaptability to dynamic topologies. The model prioritized low-hop routing (~0.8), full connectivity (~1.0), and extended lifetime (~0.75). LSTM achieved the top score (23.9%), followed by HMM (22.5%) and BAT (19.0%) (Figure 8).



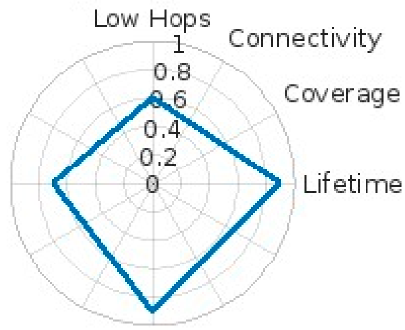
Rank	Algorithm	Confidence	Strengths
1	LSTM	23.9%	Lifetime, Coverage, Connectivity
2	HMM	22.5%	Connectivity, Lifetime, Coverage
3	BAT	19.0%	Connectivity, Lifetime, Coverage

Figure 8. Military Scenario Results with a Radar Chart.

LSTM's temporal modeling enhances real-time routing, maintains resilient communication under jamming, and supports adaptive scheduling during topology shifts—critical for sustained operational readiness in tactical environments.

5.5. Disaster Recovery

In disaster response networks, extended lifetime, wide coverage, and efficient low-hop routing (~0.8) are essential to support dispersed teams in degraded environments. LSTM again achieved the highest score (22.8%), surpassing Bird Flocking (21.7%) and HMM (21.6%) (Figure 9).



Rank	Algorithm	Confidence	Strengths
1	LSTM	22.8%	Lifetime, Coverage, Connectivity
2	Bird Flocking	21.7%	Lifetime, Connectivity, Coverage
3	HMM	21.6%	Connectivity, Lifetime, Coverage

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

Its sequence modeling capabilities enable proactive fault prediction, dynamic rerouting, and sustained connectivity despite infrastructure loss—crucial for timely coordination in emergency response operations.

Across all five scenarios, the LSTM algorithm consistently achieved the highest or near-highest confidence scores, demonstrating exceptional adaptability, latency reduction, and network lifetime enhancement. While HMM excelled in real-time connectivity and Bird Flocking showed superior resilience under dynamic conditions, LSTM's temporal modeling and results offered the most balanced and dependable performance overall.

The AI-driven framework proved effective in generalizing across diverse and evolving operational contexts. By systematically aligning algorithm selection with scenario-specific priorities, it enhances resilience, reduces latency, and ensures sustained network dependability. Ultimately, this framework advances autonomous and intelligent WSN scheduling, supporting rapid decision-making and mission success in complex, safety-critical environments.

6. Conclusions

In conclusion, 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 connectivity, coverage, and network lifetime. Complementary algorithms like HMM, Bird Flocking, BAT, and SOFM contributed strengths in network lifetime with variation in 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. 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.

Author Contributions: Conceptualization, I.A.-N.; Methodology, I.A.-N.; Software, Issam, I.A.-N.; Validation, I.A.-N.; Formal analysis, I.A.-N. and R.R.; Investigation, I.A.-N.; Resources, I.A.-N.; Writing—original draft, I.A.-N.; Writing—review and editing, R.R.; Supervision, R.R. and A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: 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.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jeyalaksshmi, S.; Ganesh, R.M. Adaptive Duty-Cycle Scheduling using Bi-Directional Long Short-Term Memory (BiLSTM) for Next Generation IoT Applications. In Proceedings of the 2022 International Conference on Computing, Communication and Power Technology (IC3P), Visakhapatnam, India, 7–8 January 2022; IEEE: New York, NY, USA, 2022; pp. 75–80. Available online: https://ieeexplore.ieee.org/abstract/document/9793496/ (accessed on 5 October 2023).
2. Haseeb, K.; Ud Din, I.; Almogren, A.; Islam, N. An energy efficient and secure IoT-based WSN framework: An application to smart agriculture. *Sensors* **2020**, *20*, 2081. [CrossRef] [PubMed]
3. Sohraby, K.; Minoli, D.; Znati, T. *Wireless Sensor Networks: Technology, Protocols, and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
4. Al-Nader, I.; Lasebae, A.; Raheem, R.; Ekembe Ngondi, G. A Novel Bio-Inspired Bat Node Scheduling Algorithm for Dependable Safety-Critical Wireless Sensor Network Systems. *Sensors* **2024**, *24*, 1928. [CrossRef] [PubMed]
5. Wang, D.; Li, X. A novel virtual sensor modeling method based on deep learning and its application in heating, ventilation, and air-conditioning systems. *Energies* **2022**, *15*, 5743. [CrossRef]
6. Gunjan. A Review on Multi-objective Optimization in Wireless Sensor Networks Using Nature Inspired Meta-heuristic Algorithms. *Neural Process. Lett.* **2023**, *55*, 2587–2611. [CrossRef]
7. Al-Nader, I.; Lasebae, A.; Raheem, R.; Khoshkholghi, A. A Novel Scheduling Algorithm for Improved Performance of Multi-Objective Safety-Critical Wireless Sensor Networks Using Long Short-Term Memory. *Electronics* **2023**, *12*, 4766. [CrossRef]
8. Wang, L.; Wei, R.; Lin, Y.; Wang, B. A clique base node scheduling method for wireless sensor networks. *J. Netw. Comput. Appl.* **2010**, *33*, 383–396. [CrossRef]
9. Alnader, I.; Lasebae, A.; Raheem, R. Using Hidden Markov Chain for Improving the Dependability of Safety-Critical WSNs. In Proceedings of the International Conference on Advanced Information Networking and Applications, Juiz de Fora, Brazil, 29–31 March 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 460–472.
10. Al-Nader, I.; Raheem, R.; Lasebae, A. A Novel Bio-Inspired Bird Flocking Node Scheduling Algorithm for Dependable Safety-Critical Wireless Sensor Network Systems. *J* **2025**, *8*, 19. [CrossRef]
11. Al-Nader, I.; Lasebae, A.; Raheem, R. A Novel Scheduling Algorithm for Improved Performance of Multi-Objective Safety-Critical WSN Using Spatial Self-Organizing Feature Map. *Electronics* **2023**, *13*, 19. [CrossRef]

12. Al-Nader, I.; Lasebae, A.; Raheem, R. 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*; So In, C., Londhe, N.D., Bhatt, N., Kitsing, M., Eds.; Smart Innovation, Systems and Technologies; Springer Nature Singapore: Singapore, 2024; Volume 379, pp. 347–366. ISBN 978-981-99-8611-8. [CrossRef]
13. Fishman, A.P.; Lyman, C.P. Hibernation in mammals. *Circulation* **1961**, *24*, 434–445. [CrossRef] [PubMed]
14. Sailhan, F.; Delot, T.; Pathak, A.; Puech, A.; Roy, M. Dependable wireless sensor networks. In Proceedings of the 3th Workshop Gestion des Données dans les Systèmes d'Information Pervasifs (GEDSIP) in Cunjunction with INFORSID, X, France, May 2010; pp. 1–15. Available online: https://hal.science/hal-01125818v1 (accessed on 19 October 2025).
15. Lai, S. *Duty-Cycled Wireless Sensor Networks: Wakeup Scheduling, Routing, and Broadcasting*; Virginia Polytechnic Institute and State University: Blacksburg, Virginia, 2010.
16. Narayan, V.; Daniel, A.K.; Chaturvedi, P. E-FEERP: Enhanced Fuzzy Based Energy Efficient Routing Protocol for Wireless Sensor Network. *Wirel. Pers. Commun.* **2023**, *131*, 371–398. [CrossRef]
17. Jung, S.-G.; Yeom, S.; Shon, M.H.; Kim, D.S.; Choo, H. Clustering Wireless Sensor Networks Based on Bird Flocking Behavior. In *Computational Science and Its Applications—ICCSA 2015*; Gervasi, O., Murgante, B., Misra, S., Gavrilova, M.L., Rocha, A.M.A.C., Torre, C., Taniar, D., Apduhan, B.O., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2015; Volume 9158, pp. 128–137. ISBN 978-3-319-21409-2.
18. Cheng, H.; Xie, Z.; Wu, L.; Yu, Z.; Li, R. Data prediction model in wireless sensor networks based on bidirectional LSTM. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 203. [CrossRef]
19. Salmi, S.; Oughdir, L. Cnn-lstm based approach for dos attacks detection in wireless sensor networks. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 835–843. [CrossRef]