A New Perceptron-Based Neural-Network Algorithm to Enhance the Scheduling Performance of Safety-Critical WSNs of Increased Dependability

Issam Alnader[1], Aboubaker Lasebae[2], Rand Raheem[3]

Middlesex University, The Burroughs, London NW4 4BT, UK
I.Al-Nader@mdx.ac.uk; A.Lasebae@mdx.ac.uk; R.H.Raheem@mdx.ac.uk

Abstract. Wireless Sensor Networks (WSNs) are embedded systems consisting of multiple distributed Sensor Nodes and usually one or more Base Stations, placed within an area of interest, to monitor and detect given behaviours and changes. Nowadays, WSNs are widely used in Safety-Critical systems where their dependability requirements are determined by the correct operations of the three primary properties: Connectivity, Coverage, and Lifetime of the network. These properties have been mostly considered independently of each other due to the complexity of addressing them simultaneously. This paper proposes a Perceptron-based Artificial Neural Network (ANN) analyser to analyse the performance of the Scheduling algorithms (e.g., where nodes alternate between awake (ON) and sleep (OFF) states) in WSNs using a MATLAB simulation environment. This approach uses a neural network to learn, train, and test the performance of such algorithms, to better the overall dependability of the network. The simulation results show possible ways to improve the lifetime of nodes using a more dynamic connectivity /coverage strategy. In particular, nodes that are switched ON more than four times were identified and classified. This has the benefit of improving network lifetime and hence its service availability and reliability attributes (dependability) by balancing the workload or the sleep schedule of those nodes, the network's lifetime increases by avoiding unnecessarily depleting nodes' energy.

Keywords: Internet of Things (IoT), WSN, dependability, WSN optimisation, WSN duty cycle, (Artificial Intelligence) AI performance analysis

1 Introduction

In recent years, the use of WSNs in the field of Safety-Critical systems has increased considerably. Owning to their easy deployment and ability to operate in a sensitive, harsh, or dangerous environment seamlessly. Safety-Critical systems are concerned with the safety of human beings and their assets [1]. In particular, their primary objectives are to avoid human injuries and deaths, economic loss, or environmental damage. A recent example, a WSN system has been deployed in the Fukushima nuclear plant in Japan, to help monitor the level of radiation. Another example is the use of WSNs in the outbreak of fire at Grenfell Tower in London, UK [2]. Therefore, ensuring dependability, during design time, and operability, during run-time w.r.t. the WSN objectives of such networks are important factors.

As mentioned earlier, the WSN consists of multiple Sensor Nodes and a Base Station as illustrated in Figure 1. Typically, a WSN is deployed in a target area of interest, to maintain the correct functional properties; monitor and detect certain events, behaviours and changes, and report the detected events to the Base Station Therefore, these functional properties influence the three crucial factors of a dependable WSN; connectivity, coverage, and lifetime. Those three factors have not been addressed widely in the literature as part of a global multi-objective optimisation problem. They are often addressed separately due to their conflicting trade-off complexity. Therefore, this paper will examine and analyse them as a single optimisation problem w.r.t. achieving their quality of service (QoS) requirements [3][4]. While system dependability has been defined as the system's ability to provide a service that can justifiably be trusted [5]. The term "justifiably be trusted" is expressed in the context of security and/or system QoS verification and validation failures. On the one hand, a System failure is related to the system/devices, while a service failure is related to the application's specific requirements. System failures can happen without forcefully causing a service failure, but this is not always the case. When a WSN communication link fails, this leads a node to use a different path such that data can still reach the Base Station. This is an example of when a system failure does not cause a service failure. On the other hand, a system failure can cause a reduction in QoS, i.e., service failure, where a link failure may cause a node to use a longer path which will cause a delay in reaching the Base Station [4].

In order to ensure WSN dependability, a number of performance challenges such as coverage, connectivity, throughput, delay, reduced data in a network, and network lifetime must be addressed first. Utilising energy efficiently under these performance challenges is still an active research area [2] [6]. A well-known optimisation problem in the WSN community is “the m-covered and connected sensor networks’ multi-objective” problem [20] [21]. Node scheduling has been proposed in the literature to address the m-covered connected WSN w.r.t the network’s lifetime, hence improving the network’s dependability too.

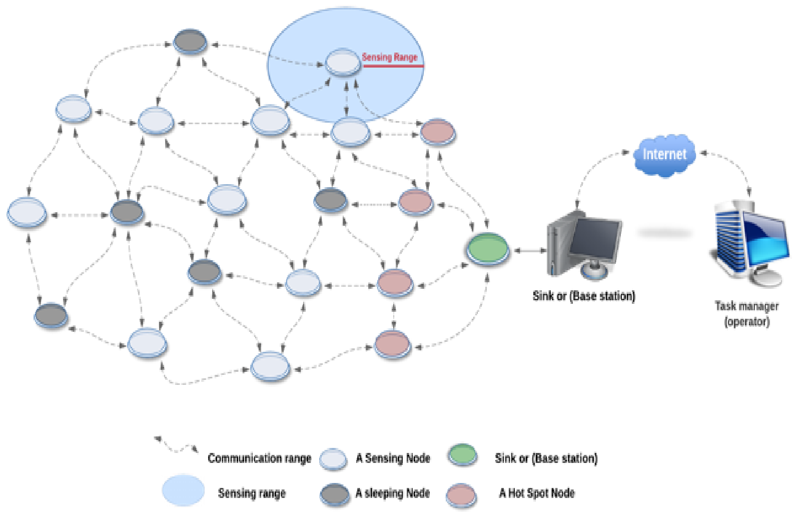

Fig. 1. The basic structure of the Wireless Sensor Network

The dependability of a system is defined in many computer system contexts. In general, a dependable system means a trustable system with its operation to achieve the required service at the time specified by the application. According to Avizienis et al. in [5], the dependability of a system is "the ability of the system to avoid service failures that are more frequent or more severe than is acceptable". [5] presents a mechanism for building a dependable system, which includes three main factors; the attributes, means, and threats to system dependability. In brief, the attributes of dependability are achieved by the means of fault avoidance, to lessen the likelihood of faults being formed within a system, whereas the means are the techniques to counter the threats that affect the dependability attributes of a system.

In principle, the threats to the dependability of the WSNs stem from the failures relating to energy efficiency [1]. Typically, it is unrealistic to assume that sensor nodes will not fail. In the WSNs, failures are expected to occur under certain conditions during the WSN run time. To succeed in deploying a dependable WSN, it is important to identify, analyse, and prioritise the different kinds of network failures based on their severity on the network performance. For example, in an application such as a fire detection system, sensor nodes must be powered on all the time to function and provide the level of service required. In this context, the most important dependability attributes are availability and reliability, given that if the system is not available during a fire, the fire will go unnoticed, and if the system is not reliable, it will not be able to detect the fire in an adequate time and manner. A relevant and recent example of how crucial these attributes are in the disaster that occurred in Grenfell Tower [2]. Had the system been more dependable, actions could have been taken faster, thus resulting in less damage to the building and more lives saved.

In order to formulate the problem of dependability testing in WSNs one must demonstrate the failure that is linked to its components:

Node failure: A node can fail due to a failure in one of its components, e.g., memory, processor, sensor, and battery. In this paper, the focus is on failure due to battery outages. This failure can cause other failures, namely communication failure and coverage failure.

Communication (link) failure. A Link between nodes is considered lossy and can fail due to many reasons, e.g., radio interference, collision, or due to a node dying as a result of fully depleting its energy.

Coverage failure: A coverage failure occurs when a node fails to collect data in its target area due to battery depletion. Coverage and communication go together. If a node covers an area but cannot communicate back to the Base Station, this is as if the area is not covered. Reciprocally, if a node can communicate back to its Base Station but there is no coverage, then detected events cannot be communicated, as if there is no communication.

Consequently, to address the node scheduling problem and improve the dependability of Safety-Critical WSN, the paper will be structured as follows: section 2 provides a review of such scheduling approaches. While in section 3 a neural network is proposed to analyse the performance and behaviour of one node-scheduling algorithm. Section 4 illustrates the achieved results via the MATLAB simulator. Finally, section 5 concludes this work.

2 Related Work

The reviewed work, confirms that the present problems within WSNs go beyond energy efficiency, other areas of focus should also include coverage and connectivity of WSNs.

Tian et al. [7]'s scheduling algorithm analyses how saving energy depends on how long a node takes to decide its eligibility to be turned off (asleep) - the faster the decision is made, the higher the energy saving. However, this approach suffers control message overheads as it requires the use of GPS for geo-location information of sensor nodes, which is convenient but expensive to utilise. As networks age in time, neither network connectivity nor coverage is preserved. When a node fails (i.e., as a result of depleted energy), the sleep nodes will remain asleep assuming failed nodes are active. As a result, the network becomes partitioned where some nodes are isolated. Tian et al.'s approach only focuses on network connectivity and energy efficiency but does not look into coverage.

Yet et al.'s robust energy-conserving protocol for long-lived sensor networks [8] PEAS tend to maintain network coverage by activating nodes that do not receive a reply to the probe message. This approach also suffers control message overheads due to broadcasting the probe messages periodically to maintain a set of active nodes. This solution also fails when a node is set to be active; it remains active till it fully depletes its energy and dies; consequently, a coverage hole appears in the network. This work does not ensure the original sensing coverage, and hence, important events may not be captured and reported to the Base Station.

An Adaptive Self-Configuring Sensor Networks Topologies (ASCENT) algorithm is presented in [9]. ASCENT has several drawbacks. First, it demands each node to be aware of the network state information resulting in control message overheads. Second, Cerpa and Estrin [9] did not consider network coverage in the design of this solution. Finally, having two states (passive and test) to alternate between active and sleep states, requires additional waiting time and data collection in each of these states; consequently; this affects ASCENT's ability to adapt to network dynamics rapidly. In addition, once nodes become active, they stay active till they deplete energy, this results in portioning the WSNs.

Viera et al. devised in [10] a scheduling algorithm based on the Voronoi diagram. The algorithm extends the network lifetime by exploiting node redundancy. Once a node is determined active and responsible for monitoring an area of interest it stays active until its energy is depleted. The work in [10] does not suggest when the turned-off nodes will be on again and it is not mentioned how well the coverage is maintained. Even if nodes in a smaller area are labelled to be turned off the issue of the network coverage is not adequately addressed.

Xu et al. proposed in [11] a scheduling solution called GAF (Geographic Adaptive Fidelity). GAF saves energy by assuming sensor nodes always maintain a good level of connectivity. This assumption is a drawback because nodes' communication links can be vulnerable to many factors that affect the performance of the scheduling algorithm. This renders their proposed solution inapplicable for safety-critical WSNs. Furthermore, GAF doesn't consider network coverage. This is mainly because nodes are turned off (scheduled to sleep) based on turning off their communication unit, causing a blind spot.

In [12], Ha proposed a sleep scheduling algorithm based on spanning tree topology to manage energy consumption and extend the network lifetime of the WSN. The algorithm is susceptible to parent node failures. For example, the failure of a parent node would cut off the communication with its sub-tree nodes (children) affecting the end-to-end communication reliability. Such failure would partition the WSN even if it remains connected. The use of tree-based routing achieves energy efficiency but at the cost of delay because data will always be reported via a single route. This prevents promptness and timing performance of reporting data to the sink node at the appropriate desired time.

A solution called a Lightweight Integrated Protocol Suite (LIPS) was proposed by Tate et al [13]. The solution assumes that sensor nodes within a cell are able to hear each other and hence synchronise. In WSNs, the wireless links are subject to many sources of obstacles which makes it difficult to maintain connectivity. In a forest fire system scenario, as nodes are deployed in a harsh environment and surrounded by dynamic obstacles, this would have an impact on energy consumption and the time a WSN takes in order to converge to the ready stable state.

In [14], Yang addresses a sleep scheduling strategy at the application layer where a set of sensor nodes is selected and activated according to a cost function to extend the network lifetime of the WSN. Although the proposed selection strategy based on Senor Usage Index (SUI) seems to have achieved a near-optimal lifetime, the periodic broadcast of a beacon control message by the Base Station every half an hour to manage the WSN would lead to communication overhead which in turn leads to energy inefficiency.

In [15], Ambrose proposed a distributed scheduling algorithm for composite events detection in WSNs to solve multi-objective problems concerning coverage and energy consumption. A composite event detection is an event that consists of two or more simple events. Ambrose's solution tends to save energy by (1) scheduling nodes to sleep, and (2) minimising the transmission range between each node in the network while maintaining network connectivity. In [15], Ambrose assumed that node locations are known either by GPS or other computation localisation methods. This assumption is unrealistic and cost-inefficient in large-scale WSNs. The deployment environment in [15] was a simple grid area, but large-scale WSNs usually consist of complex and dynamic obstacles. The distributed scheduling algorithm implies a complex computation process, especially when maintaining network connectivity, which causes an energy overhead. In a busy WSN, such a computation is not acceptable, hence the need for a simple computation algorithm is highly sought after in resource-constrained WSNs.

Chipara [16] proposed ESSAT algorithm which introduces a sleep schedule that considers both energy consumption and the application deadline. ESSAT consists of two main components: a traffic shaper and a sleep schedule. ESSAT addressed a challenging optimisation problem, but it depends on the prediction of the time properties of the workload information that is obtained from earlier message exchanges amongst sensor nodes. ESSAT is in turn vulnerable and dependent on time. However, an advantage is that the sleep schedule requires no time synchronisation algorithm and hence a node can sleep and wake up according to information made available by message communication.

Cardei et al. [17] proposed a scheduling algorithm that organises the network's sensor nodes network into sets. Each set covers the target area at different time intervals. Finding the maximum number of sets can provide good network coverage and extend the network lifetime. Cardei 's approach has limitations; First, it requires the geo-location of each sensor node which is very expensive. Many applications do not require the use of the GPS system. In addition, the solution in [17] focuses on providing good sensing coverage while omitting network connectivity. Thus, important events may be detected but not reported back to the Base Station for processing, which is meaningless.

Wu et al. [19] proposed lightweight deployment-aware scheduling for wireless sensor networks. In [19] Wu et al. assumes a sensor node maintains three states: on-duty, ready-to-off, and off-duty to improve the network lifetime as part of its scheduling algorithm. LDAS has managed to extend the network lifetime, but it didn't consider network coverage.

He et al proposed a distributed scheduling algorithm in [18]. It suffers from a control message overhead. Another drawback is in its reactive approach where sensor nodes may drift in time. As a result, sentry (master) nodes may not know for certain which nodes are asleep and which are awake. Hence, the non-sentry nodes will keep broadcasting multiple wake-up beacon messages to all non-sentries in the network.

In summary, most scheduling algorithms in the literature do not address the three requirements of the WSN altogether, i.e., coverage, connectivity, and network lifetime. Most works address each requirement in isolation and do not consider network coverage. Other works such as [17] and [7] consider network coverage but without ensuring a good level of network connectivity; and [7] uses a very simple WSN scenario lacking analysis and tests on realistic and complex WSNs. There are two published works that consider addressing the three objectives collectively: Randomised Coverage-based Scheduling (RCS) algorithm by C. Liu et al. [21]; and a Clique-based node scheduling by Lei Wang [20]. C. Liu et al. [21] have shown that partitions can occur further away from the

Base Station as the death of subsequent nodes can result in live nodes attempting to compensate for the drop-in coverage and connectivity in the system. More details of C. Liu et al. [21] solution is presented in section 4. The Clique-based approach [20] relies on a complex node localization and clustering mechanism. While the Clique-based algorithm in [20] guarantees maximum connectivity and coverage ratio, this comes at the expense of heavy computation, thus making the verification of the Clique-based algorithm more difficult.

3 Proposed Framework

This section presents the proposed solution and the simulation of the RCS algorithm [21] in MATLAB. The elements of the simulation are the RCS scheduling algorithm, the Perceptron ANN-based analysis tool, and a generic WSN application. Figure 2 below illustrates the basic framework of the analysis of the Scheduling Algorithm:

1. Initialise the Sleep/Wake up Schedule Algorithm via MATLAB, with randomised Node deployment.

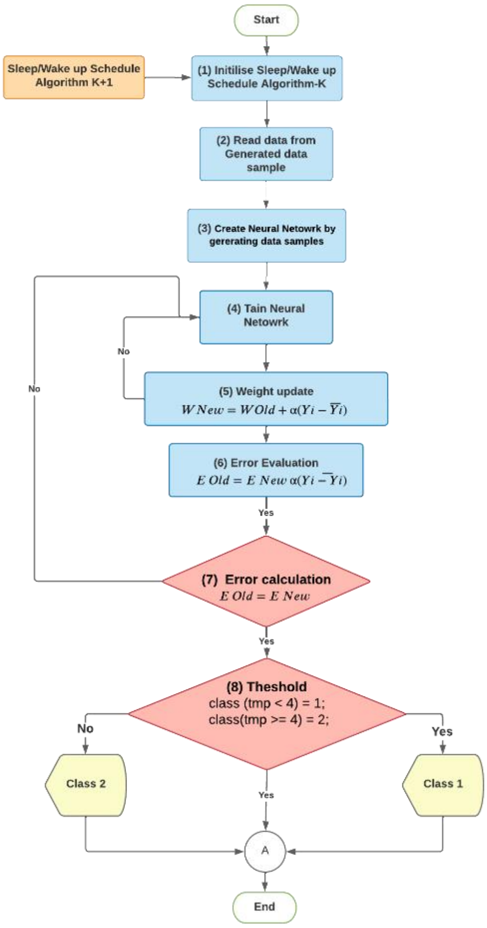

Fig. 2. Proposed Framework

2. Apply a Neural Network Schedule Analyser to predict which Nodes are activated more often, given the number K of groups.

3. Evaluate the data collected to ensure that the error rate (accuracy of classification) is withheld to a decent percentile: if the error is not within an acceptable range, another scenario is commenced to increase the sample size of data, as continuous testing allows the neural network analyser to predict with a higher level of accuracy.

The initial experiment is to observe how a WSN would perform under classical configurations, which will then be compared to a similar experiment run with the inclusion of a scheduling algorithm, as displayed in Figure 3 below. During the test-based portion of the experiment, a neural network will be supplemented into the system.

3.1 problem formulation

Perceptron supervised learning algorithm is a simple version of artificial neural networks. The basic process is to train the given input X with the output Y mapping and then test it with some data samples. It is a supervised algorithm as it supervises the outputs mapping data Y. The main goal is to convert the data into two classes; class A and class B. Hence, a perceptron algorithm is known as a binary classifier as shown in equations (4) and (5). The following equation represents the data structure of the perceptron algorithm in the WSN scheduling algorithm.

$$R = i1nXiWi + b \qquad (1)$$

Where R is the results, sigma is the summation of all intended X features and the associated weight W. n represents the number of features and b is the bias value. In the main problem of concern domain X represents the position of each sensor node in the network to the Base Station. Simplifying the above equation will give us the following result:

$$R = X \,.\, W + b \qquad (2)$$

The following function called the activation function, takes the input of equation (2), R, and produces the binary classification.

$$\boldsymbol{F(x)} = \frac{1}{1+e^{-x}} \qquad (3)$$

The output of the activation function is of either 1 or 0. In the assumed scenario it is called class 1 or class 2.

$$(X \,.\, W + b) > 4 \; = \; class\ 1 \qquad (4)$$

$$(X \,.\, W + b) < 4 = \; class\ 2 \qquad (5)$$

The above process of training goes through multiple iterations in order to reach zero error of the smallest possible error value.

$$W\mathrm{New} = W\mathrm{Old} + \propto (Yi - \overline{Yi}) \qquad (6)$$

The update function adds the new weight (Wnew) to, the learning rate, multiplied by the difference of the right Yi and Yi wrong output which is the error rate. The weight is updated until the desired solution is achieved. Note, the value of the error rate which influences the learning time.

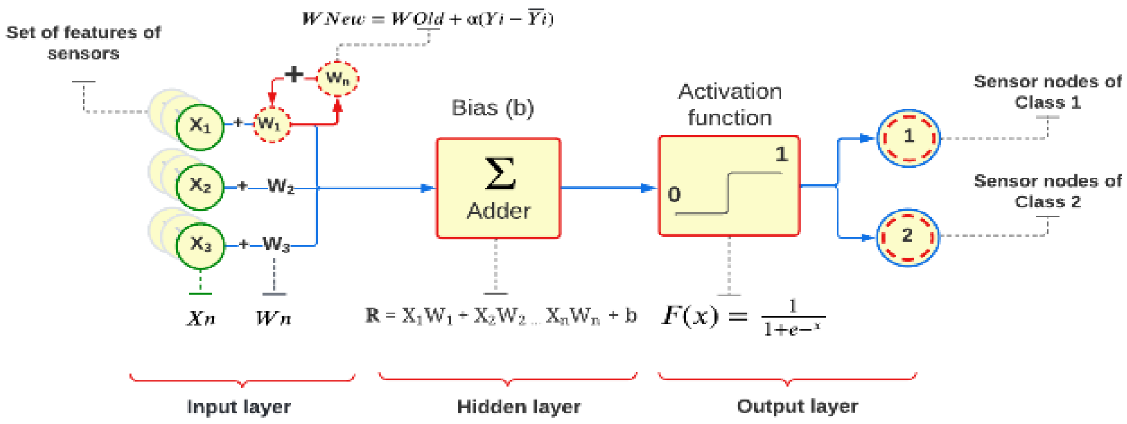

Fig. 3. Perceptron-based Neural-Network Algorithm

The following pseudocode summarises the proposed solution:

Algorithm: Perceptron-based Neural-Network Algorithm

```
Input: Set of sample data of Sensor Nodes
Output: Set of sample Sensor Nodes of Class 1 or Class 2

Initialize:  Sleep Awake Cycle (k to k+1);
Read data from the data sample as ~x
Train Neural Network ()
Choose an initial weight vector ~W
Initialize the minimization approach
 While the error did not converge do
 For all (~x, ~d) € D do
     apply ~x to the network and calculate the network output
     calculate error (~x)
 end For
  calculate error (D)
 For all weights summing over all training patterns
  perform one update step of the minimization approach
 end while
if Threshold > 4 then
     Class 2
 Else
     Class 1
end if
```

4 Simulation Experiments

The result of the analysis is populated from both the classical configuration and the configuration with the scheduling algorithm's addition to compare the rate of improvement in efficiency between the two scenarios; before and after applying the neural network analyser, as displayed in Figures 4 to 10.

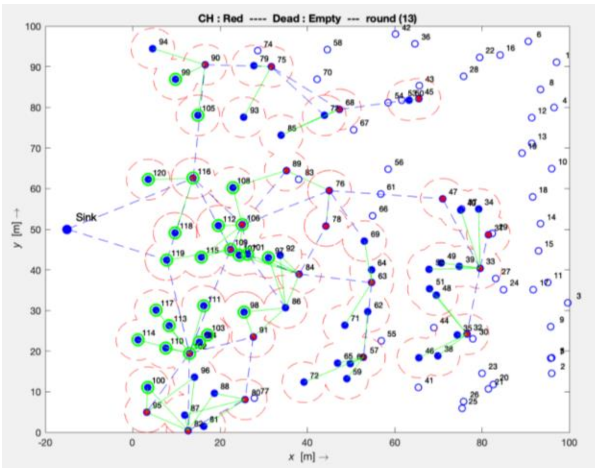

Fig. 4. WSNs without the use of scheduling-based Neural Networks

As illustrated in Figure 4, a WSN consists of 1 sink node and 150 nodes, where each node has a transmission range (R) of 3.5 meters on an area of interest of x = 100, y = 100, the initial energy consumption is 0.25 joules per node, whilst the energy required to receive and to transmit one bit is 1.0000e-09 joules per node, this is also the value of the data aggregation. The desired packet size of a cluster head per round is 3000 bits; all the data mentioned can be seen in the table below. As the simulation runs, the network's performance can be observed using the standard cluster routing algorithm, without the scheduling algorithm. Additional important parameters were considered in the simulated environment as represented in Table 1.

Table 1. Parameters set within the simulation of the WSN

Simulation Parameter	Description	Value
Coverage Parameters	Distance from sink to sensor area	5 meters
Node Parameters	Energy of free space model amplifier	5.00e-08 joules
	Maximum Range of transmission per node (R)	15 meters
Scheduling Algorithm Parameters	Desired Percentage of Cluster Heads	5/100
	Packet Size for normal nodes per round	80 bits
Simulation Parameters	Number of Rounds	2000

4.1 Experiment 1: WSN without a scheduling algorithm

The motivation behind this experiment is to observe the behaviour of the WSN without any supporting scheduling algorithm i.e., having the nodes "turned on" throughout the experiment time period. This will allow us to evaluate how nodes consume energy and behave in the deployed environment.

In the simulation, a Base Station is considered and 150 stationary homogeneous sensor nodes are distributed randomly in a 100 by 100 m2 field as the area to be monitored. The experimentation of this work is based on executing multiple simulation rounds. The number of rounds is set to 2000 as the upper bound. The simulation will halt execution once 95% of the nodes reach complete energy depletion i.e., nodes break point. The sensing range of each sensor node is set to 5 m. The maximum communication range per sensor is set to 15 m. The network lifetime is calculated by the consumption of communication (transmitting and receiving), and sensing units of each node. See Table 1 for a tabulated form of the aforementioned parameters.

At the beginning of the simulation, each node starts its operation with an initial energy of 3 joules. Every sensor node consumes $1.0000e^{-09}$ joules of its initial energy when transmitting or receiving one bit.

In addition, amplifying the signals i.e., digitising them for transmitting to/from nodes in free space consumes 5.00e-08 joules of energy. The factoring in this specific energy consumption element has been absent in the literature. This adds more novelty and contribution to this paper in the WSN community. As the simulation progresses, there will be more energy consumption by the nodes which will have effects on network lifetime, connectivity, and coverage.

4.2 Experiment 2: WSN with RCS scheduling algorithm

The motivation behind this experiment is to evaluate the behaviour of nodes when a scheduling algorithm is implemented to turn nodes ON and OFF w.r.t. optimisation objective(s). In this experiment, albeit using the same configurations of Experiment 1 above, a new parameter is added to the simulation, K, by the RCS algorithm. K is equal to the number of node subsets i.e., the division of the nodes into a number of K subsets of nodes. In this experiment, K is set to 2.

On one hand, figure 5 represents the energy consumption of each node which is dramatically higher without the scheduling algorithm (Experiment 1), even managing to cease activity at 640 seconds. In contrast, when RCS scheduling is used (Experiment 2), each node uses substantially less energy and remains active for 1180 seconds. This shows that the WSN application is less energy-efficient in the absence of the RCS scheduling algorithm.

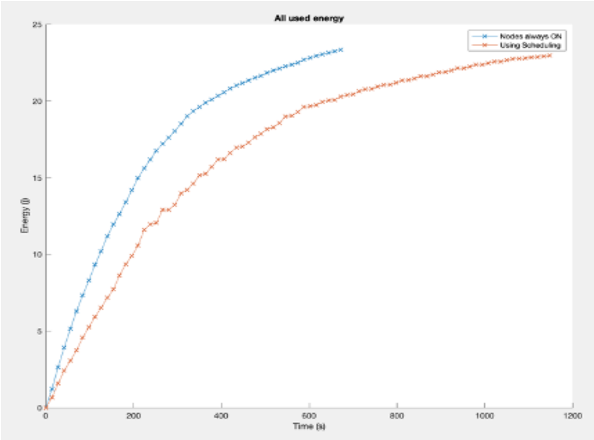

Fig. 5. Energy Consumption Level

On the other hand, figure 6 presents the number of nodes alive at any given round. The number of active nodes starts dropping at round 5, 50% of nodes have ceased to be alive at round 23, and 100% are dead at round 49. In contrast, in Experiment 2, 50% of nodes died at round 34, and 100% at round 85.

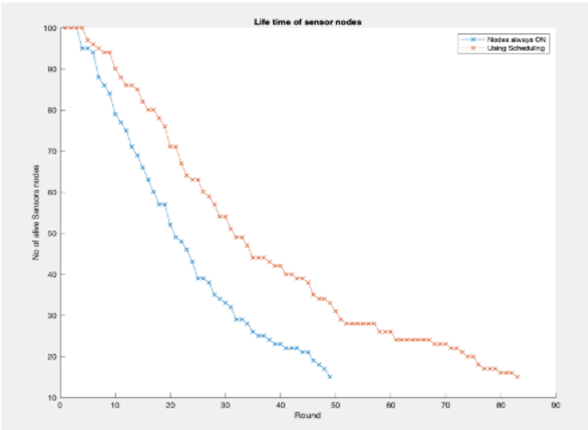

Fig. 6. Lifetime of Sensor Nodes

The throughput of the data passed through 100 sensor nodes over 1200 seconds. Figure 7 shows that the throughput is higher for nodes activated in the absence of a scheduling algorithm.

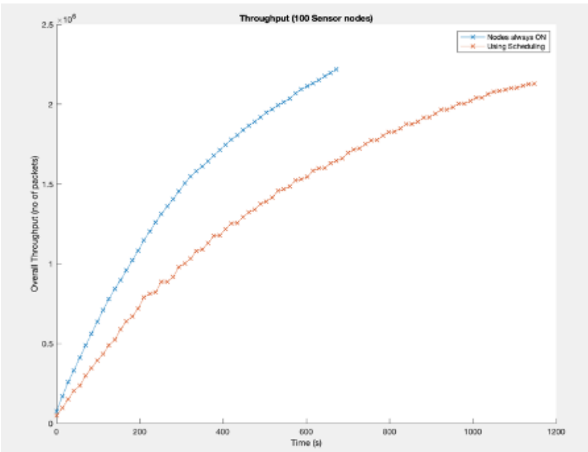

Fig. 7. Throughput

Figure 8 depicts the ratio of coverability for a sensor node to monitor, using a scheduling algorithm (experiment 2) and in its absence (experiment 1). In conjunction with Figure 9, once the nodes cease to be connected, their coverage will drop off without a scheduling algorithm

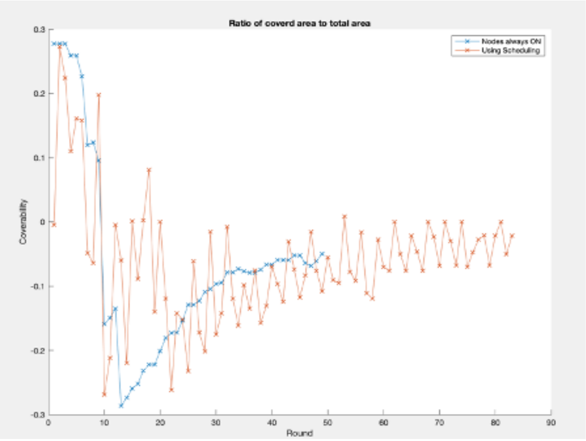

Fig. 8. Coverability throughout simulation

During 90 rounds, the nodes' connectivity was tested separately using a scheduling algorithm (experiment 2), and without using one (experiment 1). The results show that connectivity efficiency is dramatically reduced when the system is not supplemented with a scheduling algorithm, as the nodes cease to run after 50 rounds. The data provided in the graphic shows service availability is maintained for a longer duration when a scheduling algorithm is present in a system.

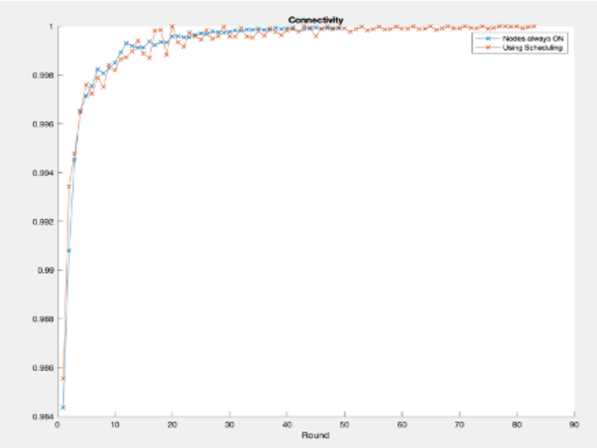

Fig. 9. Connectivity

4.3 Experiment 3: WSN with RCS scheduling algorithm supported by ANN

The motivation behind this experiment is to support the existing RCS scheduling algorithm [20] with the ANN algorithm proposed in this paper, by predicting the overworked (switched ON) nodes. Such nodes, highlighted in green circles in Figure 10, will be forced to be switched ON by the RCS algorithm more than four times.

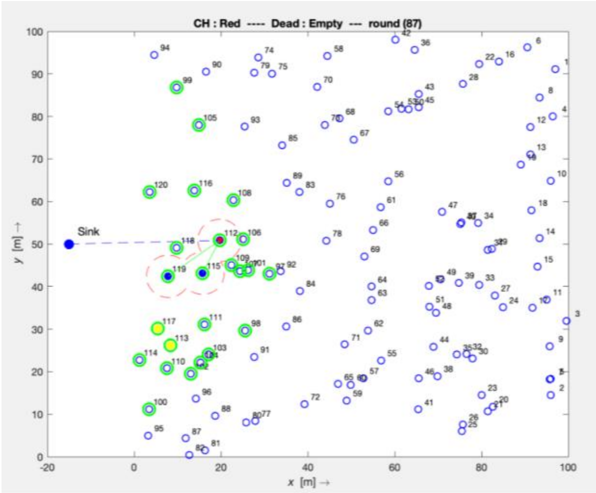

Fig. 10. Scheduling algorithm for WSNS with the use of neural networks

Perceptron Artificial Neural Network Analyser. The purpose of the ANN is to analyse the performance of the scheduling algorithm by returning the number of nodes that are overworked. The ANN returns for each node if it is overworked (class B) or not (class A). The data collected from the ANN is used to further analyse and enhance the dependability of the WSN. The following are the new ANN parameters.

Table 2. Parameter of the Perceptron Artificial Neural Network Analyser

Neural Network Parameters	Value
NumToCreate (# of data sampled)	1000-2000
R (Maximum Range of Transmission)	15
S X(X-Axis Coordinates of Sink Node)	-15
S Y(Y-Axis Coordinates of Sink Node)	50
Number of Input Variables	2
Number of Functions to fit	1
Ratio of Data used for creating model	0.85
Error Rate @ 200th Iteration	4.990006e-01%
Accuracy of Classification @ 200th Iteration	86.1302

Training Progression. Figure 11 depicts the neural network's training progression, and the error rate changes across every consecutive iteration of the training, ending at 200. The error rate reaches an acceptable range during the 149th iteration, holding a value of 4.998903e-01%. The final stabilised error rate at the 200th iteration is 4.990006e-01%. The accuracy of classification for the training is 86.1302, meaning the neural network could detect which nodes will be 'overworked', kept ON, in four or more occurrences throughout the 200 iterations of testing taking place in the training segment of the experiment.

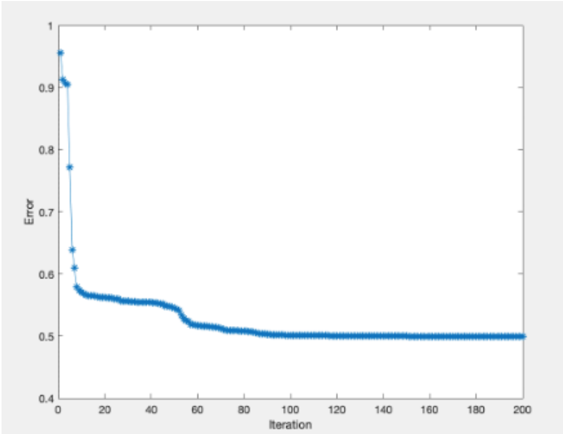

Fig. 11. Training Progression

Neural Network Analysis. The experiment was undertaken to emphasise the scheduling algorithm's strengths; a simulation was performed using a pre-existing replicated RCS within MATLAB, which was built from the ground up as there is no open-source framework for this algorithm and one simulation in the absence of the RCS. The algorithm in question was chosen due to its ability to consider coverage, connectivity, and a lifetime of a network regarding wireless sensor nodes. Coverage, connectivity, and a lifetime of a network were used to measure the wireless sensor network's performance, as shown in the figures presented above as matrices. It was clear that the RCS's presence substantially increased a system's performance in the parameters mentioned earlier throughout every matrix.

Evaluation of the ANN analysis. Following our experiment, the ANN algorithm returns 19 Nodes that died earlier than any other node in the network (Overused). Figure 10 shows all 19 Overused Nodes (Highlighted in green colour), which are situated in the 1-Hop area. This will create a partition in the network whereby events will be detected in the subsequent k-Hop areas ($k > 1$) but will not be reported to the Base station, thus rendering the whole implementation of the WSN useless and undependable. The ANN analysis hence confirms the initial hypothesis that nodes that are closer to the Base station are more likely to undertake more workload. However, the ANN analyser is limited to discovering the number of Overused nodes and where partitioning is more likely to occur in the network but does not provide a way to improve the existing algorithm in question (e.g. RCS in this case) or come up with a new suggestion to revive the 1-Hop area, e.g., by utilising nodes in the 2-Hop area instead.

Given the result of the ANN analyser, it would be interesting to synthesize a new scheduling that would improve on the scheduling from the analysed algorithm (here RCS) or would return a verdict indicating that no further improvement is possible. We leave this question to future work. Alternatively, we could try and propose a new scheduling "manually", however, approaches different from that used by RCS exist which can provide better results such as the Hidden Markov Model-based algorithm proposed by the authors in Alnader et al., 2023 [1].

5 Conclusion

This paper addressed the lack of work done on the m-covered and connected sensor networks' multi-objective optimisation problem in WSNs. While C. Liu had proposed an RCS algorithm to solve it, it was not clear how good the RCS solution is. To answer this, we have proposed a perceptron-based Neural Network Analyser which analyses the performance of RCS scheduling and returns the expected lifetime of every node. Applied to a network of 150 nodes, the ANN analyser detected that the first 19 nodes to die were all from the 1-Hop area. This suggests that RCS can be improved, although the ANN analyser does not say how RCS can be improved. In order to answer the question of improving C. Liu's [21] work, the authors are exploring new techniques, e.g., the AI-based Hidden Markov Model, bio-inspired Bat model, Self-Organising Map Model, and Long-Short-Term Memory(LSTM) Model. Thanks to this, it might be possible to determine in the near future, which approach provides the better guarantees to obtain an optimal solution to the scheduling problem in WSNs.

References

1. I. Alnader, A. Lasebae and R. Raheem "Using Hidden Markov Chain for Improving the Dependability of Safety-Critical WSNs", the 37-th International Conference on Advanced Information Networking and Applications (AINA-2023), 2023.
2. J. Preston, Grenfell tower: preparedness, race and disaster capitalism. Springer, 2018.
3. F. Sailhan, T. Delot, A. Pathak, A. Puech, and M. Roy, 'Dependable wireless sensor networks', in 3rd workshop Gestion des Données dans les Systèmes d'Information Pervasifs (GEDSIP) in cunjunction with INFORSID, 2009, pp. 1–16, Accessed: Jun. 21, 2015. [Online]. Available: http://cedric.cnam.fr/~sailhanf/publications/gedsip.pdf.
4. J. E. Mbowe and G. S. Oreku, 'Quality of service in wireless sensor networks', Wirel. Sens. Netw., vol. 6, no. 2, p. 19, 2014, Accessed: Dec. 13, 2016. [Online]. Available: http://search.proquest.com/openview/5fff55ea50299f7ca559e98320247861/1?pq-origsite=gscholar.
5. A. Avizienis, J.-C. Laprie, B. Randell, and others, Fundamental concepts of dependability. University of Newcastle upon Tyne, Computing Science, 2001.
6. Y. E. Aslan, I. Korpeoglu, and Ö. Ulusoy, 'A framework for use of wireless sensor networks in forest fire detection and monitoring', Comput. Environ. Urban Syst., vol. 36, no. 6, pp. 614–625, 2012, Accessed: Nov. 05, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0198971512000300.
7. D. Tian and N. D. Georganas, 'A Coverage-preserving Node Scheduling Scheme for Large Wireless Sensor Networks', in Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, New York, NY, USA, 2002, pp. 32–41, doi: 10.1145/570738.570744.
8. F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, 'PEAS: A robust energy conserving protocol for long-lived sensor networks', in Distributed computing systems, 2003. Proceedings. 23rd international conference on, 2003, pp. 28–37, Accessed: Sep. 26, 2015. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1203449.
9. A. Cerpa and D. Estrin, 'ASCENT: Adaptive self-configuring sensor networks topologies', Mob. Comput. IEEE Trans. On, vol. 3, no. 3, pp. 272–285, 2004, Accessed: Oct. 09, 2015. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1318596.
10. M. A. M. Viera, L. F. M. Viera, L. B. Ruiz, A. A. Loureiro, A. O. Fernandes, and J. M. S. Nogueira, 'Scheduling nodes in wireless sensor networks: A Voronoi approach', in Local Computer Networks, 2003. LCN'03. Proceedings. 28th Annual IEEE International Conference on, 2003, pp. 423–429, Accessed: Aug. 04, 2015. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1243168.
11. Y. Xu, J. Heidemann, and D. Estrin, 'Geography-informed energy conservation for ad hoc routing', in Proceedings of the 7th annual international conference on Mobile computing and networking, 2001, pp. 70–84, Accessed: Oct. 11, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=381685.
12. R. W. K. Ha, 'A Sleep-Scheduling-Based Cross-Layer Design Approach for Application-Specific Wireless Sensor Networks', 2006, Accessed: Jun. 21, 2015. [Online]. Available: https://uwspace.uwaterloo.ca/handle/10012/918.
13. J. Tate and I. Bate, 'LIPS: A Protocol Suite for Homeostatic Sensornet Management', in Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on, 2011, pp. 263–272, Accessed: Jun. 26, 2015. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5773400.
14. O. Yang, Sleeping strategies for wireless sensor networks. University of Rochester, 2011.
15. A. I. Ambrose, 'Scheduling for Composite Event Detection in Wireless Sensor', Master Thesis Fla. Atl. Univ., 2008, Accessed: Aug. 06, 2015. [Online]. Available: http://fau.digital.flvc.org/islandora/object/fau%3A2881/datastream/OBJ/view/Scheduling_for_composite_event_detection_in_wireless_sensor_networks.pdf.
16. O. Chipara, 'Towards real-time wireless sensor networks', 2010, Accessed: Jun. 21, 2015. [Online]. Available: http://openscholarship.wustl.edu/etd/62/.
17. M. Cardei, M. T. Thai, Y. Li, and W. Wu, 'Energy-efficient target coverage in wireless sensor networks', in INFOCOM 2005. 24th annual joint conference of the ieee computer and communications societies. proceedings ieee, 2005, vol. 3, pp. 1976–1984, Accessed: Feb. 09, 2016. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1498475.
18. T. He et al., 'Energy-efficient surveillance system using wireless sensor networks', in Proceedings of the 2nd international conference on Mobile systems, applications, and services, 2004, pp. 270–283, Accessed: Jul. 14, 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=990096.
19. K. Wu, Y. Gao, F. Li, and Y. Xiao, 'Lightweight deployment-aware scheduling for wireless sensor networks', Mob. Netw. Appl., vol. 10, no. 6, pp. 837–852, 2005, Accessed: Feb. 27, 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=1160129.
20. 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.
21. C. Liu, K. Wu, Y. Xiao, and B. Sun, 'Random coverage with guaranteed connectivity: joint scheduling for wireless sensor networks', IEEE Trans. Parallel Distrib. Syst., no. 6, pp. 562–575, 2006, Accessed: Mar. 07, 2016. [Online]. Available: http://www.computer.org/csdl/trans/td/2006/06/l0562-abs.html.