BALKAN JOURNAL OF ELECTRICAL & COMPUTER ENGINEERING, Vol. 11, No. 3, July 2023

Deep Belief Network Based Wireless Sensor Network Connectivity Analysis

Ayhan Akbas and Selim Buyrukoglu

***Abstract*—Wireless sensor networks (WSNs) are widely used in various fields, and their deployment is critical to ensure area coverage and full network connectivity to achieve the maximum network lifetime. In this study, we present a mixed-integer programming (MIP) model that deeply investigates deployment parameters to optimize lifetime and analyze network connectivity. We further analyze the obtained results using Deep Belief Network (DBN) and Deep Neural Network (DNN) algorithms to achieve higher accuracy rates. Our evaluation shows that the DBN outperforms the DNN with an accuracy rate of 81.2%, precision of 81.2%, recall of 99.1%, and an F1-Score of 0.78. We also utilize two different datasets to justify the efficiency of the DBN in this research. The findings of this study emphasize the validity of our DBN algorithm and encourage further research into lifetime optimization and connectivity analysis in WSNs.**

***Index Terms*—Deep Belief Network, Wireless Sensor Network, Connectivity**

I. Introduction

WIRELESS sensor networks (WSNs) are networks of dedicated sensors distributed in space that monitor physical conditions in a given environment, and collect and transmit the data they gather to a central base station. With the increasing use of WSNs, research into the deployment of sensor nodes has become more intense in recent years. The main goals of sensor node deployment are to achieve the best possible network lifetime and the widest possible coverage area at the lowest cost possible. For that reason. in the planning phase of a WSN deployment, the aim is to maximize network lifetime while achieving planned coverage within the designated budget.

Wireless sensor networks (WSNs) rely on a sparse placement of sensor nodes on a large scale to monitor physical conditions and transmit data to a central base station. Ensuring connectivity among these nodes is crucial for accurate data collection. To address this issue, several studies have been conducted in recent years. Sheikh-Hosseini and Hashemi, for instance, studied node connectivity and different coverage types in WSN deployments [1]. They proposed a new model that optimizes node placement to cover all targets with the required number of nodes while maximizing lifetime and coverage using genetic algorithms. Another study by Senouci and Mellouk [2] investigated optimal plan strategies for WSN topologies to achieve optimal network connectivity, sensing coverage quality, reliability, and lifetime with minimal cost. They developed a probabilistic-based communication cost model and demonstrated that their deployment approach can meet the needs of real-world fusion-based WSNs with predictable performance. Aitsaadi et al. [3] also addressed the deployment problem of WSNs with the objective of achieving the best network topology with minimal deployment cost while ensuring network connectivity and optimal network lifetime. They proposed a Tabu search metaheuristic and multi-objective deployment algorithm for their solution. Sengupta et al. [4] investigated the optimal trade-offs among coverage, lifetime, energy consumption, and connectivity using a multi-objective evolutionary algorithm-based model that outperforms single-objective node deployment schemes. Sevgi and Kocyigit [5] proposed a novel framework for the optimal deployment of WSNs from the connectivity perspective, particularly for random deployments. Overall, these studies contribute to the understanding and optimization of WSN deployments for connectivity, coverage, reliability, and lifetime, opening avenues for further research in the field [6], [7].

Although several studies have investigated connectivity issues in wireless sensor networks (WSNs), these studies have mostly utilized multilayer perceptron (MLP) and backpropagation (BP) [8] to address the problem. Notably, there is a lack of research that has employed Deep Belief Networks to address WSN connectivity issues. Given the need for enhanced performance and robust classification in WSNs, exploring the potential of DBN becomes imperative. DBN's superiority lies in its efficient utilization of hidden layers, yielding substantial performance gains when compared to Multilayer perceptron models. Furthermore, DBN exhibits specific robustness in classification tasks, effortlessly handling variations in topology, internode distances, transmission power levels and other crucial channel parameters. Therefore, incorporating DBN into the realm of WSN research can significantly contribute to overcoming connectivity challenges and advancing the capabilities of wireless sensor networks.

AYHAN AKBAS is with the Department of Computer Engineering, Engineering Faculty, Abdullah Gul University, Kayseri, Turkey e-mail: (ayhan.akbas@agu.edu.tr)

https://orcid.org/0000-0002-6425-104X

SELİM BUYRUKOGLU is with the Department of Computer Engineering, Engineering Faculty, Cankiri Karatekin University, Cankiri, Turkey e-mail: (sbuyrukoglu@karatekin.edu.tr)

https://orcid.org/0000-0001-7844-3168

Manuscript received Apr 11, 2023; accepted July 03, 2023.
DOI: 10.17694/bajece.1281060

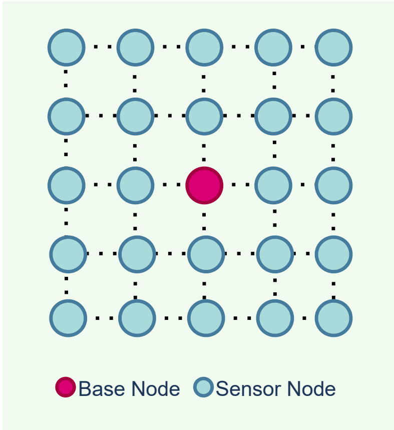

Fig. 1: Distributions of nodes in WSN grid topology

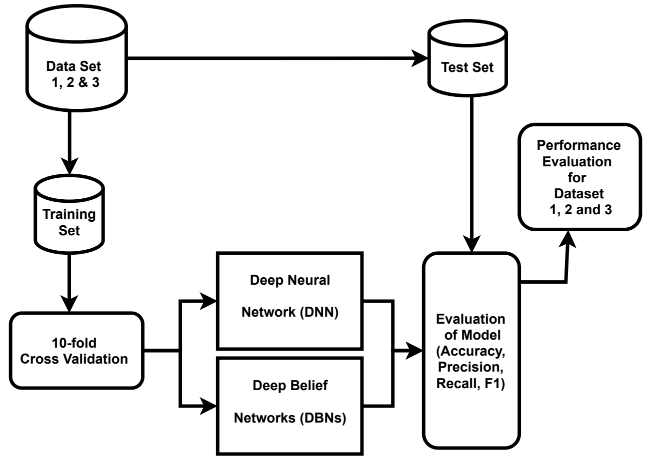

Fig. 2: Proposed Model Diagram

II. WSN SYSTEM MODEL

In this model, WSN network topology is constructed as a data flow graph and mathematically modelled using Mixed Integer Programming (MIP) in GAMS [9].

A. Wsn Model Overview

The WSN is modelled as a square topology (Fig.1) network consisting of a sink node in the center and sensor nodes around it. Sensor nodes in the network either transfer the data they collect directly to the sink node (single-hop) or relay it over other nodes (multi-hop). The 60-second round time is shared by all nodes in equal time slots, and each node transmits data in its own time slot. In each round, each node transmits its generated packet data in its own time slot. Internode transmission is performed by handshake, and every packet sent is confirmed with an ACK message, confirming that the transmission was successful. In the optimization, the maximum network lifetime is set as the objective in the linear programming. In the design of the model, it is assumed that the nodes are stationary, and all have clock synchronization with the sink node. Furthermore, we assume that the sink node's energy resources are limitless. All topology and route information is provided at the sink node. Furthermore, we assume that the data packets cannot be fragmented or aggregated.

B. Link Layer Model

In this study, the log-normal shadowing model [10] is assumed and a two-way handshaking link layer data transmission model is used, which is the extended version of the study by Akbas et. al.[11]. The MIP Model has been coded in GAMS [9] and MATLAB [12], and verified with real-world data.

C. Linear Programming Model

Each data transfer between nodes can be considered as an energy expenditure. Each bit of data sent is an energy cost that reduces the lifetime of the node. Therefore, calculating the maximum network time for the WSN can be considered

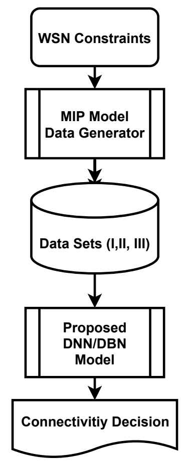

Fig. 3: WSN dataset creation

as a graph flow problem to be solved. In complex integer programming, we calculate the maximum network lifetime by defining the flow constraints on the network and modelling the flow on the WSN. The constraints we used in the model are:

- Balanced flow constraint states that for each node, the sum of the data produced in a node and the data coming from outside the node is equal to the amount of data sent from the node.
- The channel bandwidth constraint limits the channel bandwidth required to perform communication operations at each node.
- The interference matrix guarantees that the total duration of inbound data streams, outbound data streams, and interference streams is limited by the total network lifetime time.
- The non-negative constraint ensures that no data stream can be negative, i.e., all flows have to reach the sink node.

III. PROPOSED APPROACH

This section presents the proposed WSN classification approach, including four phases, which are data creation (Fig.3), data cleaning and preprocessing, deep learning models, and evaluation metrics. The proposed model diagram is depicted in Fig.2.

A. Data Creation

Three different datasets were created and used in this study to reveal the efficiency of the deep neural network (DNN) and deep belief networks (DBNs) in the use of variety of datasets. The rest of this section explains the details of three different datasets.

We have created and used 3 sets of data as given in Fig. 3, which are:

- Dataset I: The dataset has been created with the shadowing factor taken into account. The dataset has the following features: Node-count (49,81 and 121 nodes), Packetsizes (1,2,3,4,5,6 and 8), Internode distances (50 to 109 meters with 1 meter increments), Full-Connectivity (1 or 0). It has 1260 records.
- Dataset II: Exactly the same as Dataset I except for the fact that the shadowing factor has not been considered in the calculations.
- Dataset III: Dataset-III obtained through the outcome of the model, contains the following features: Shadowing (1.5 to 5.5), Path Loss exponent (2 to 4), Packet Size (1 to 8), Transmission Powerlevel (6 to 26), Node Count (9 to 121), Internode Distance (20 to 100) and the Network lifetime (0 to 5M). The size of the dataset is 8200 records.

B. Deep Learning Models

In this study, Deep Belief Networks (DBNs) and Deep Neural Network (DNN) algorithms were used in this study. The dataset was split into training (80% - 941 for train) and testset (20% - 236 for test) based on the hold out method for datasets I and II. Also, dataset III has 39375 samples, which were split into training (80% - 31500 for training) and testset (20%-7875 for testing). Ten-fold cross-validation was used in the training process of the employed DBNs and DNN while the test set (unseen data) was used to obtain the evaluation performance of the employed models. In other words, nested cross-validation was used in this study. The rest of this section covers the models' background and creation process.

1) Deep Neural Network (DNN): The structure of artificial neural networks (ANN) consists of input, hidden, and output layers. If the structure of an ANN consists of more than one hidden layer, it is considered a deep neural network (DNN) [13]. The structure of the created DNN is presented in Fig. 4. As it can be seen from Fig. 4, one input, two hidden layers, and one output layer were used in the judgement of WSNs' connectivity. Three nodes for input layers were used, while one node was used in the created DNN. Also, six and three neurons in the hidden layers were used as a result of attempting to use the various numbers of neurons in the hidden layers. Sigmoid was used as an activation function in the deep neural network model.

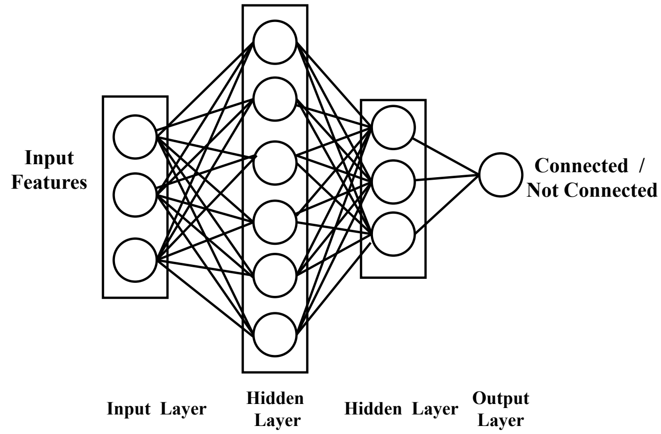

Fig. 4: Structure of the employed Deep Neural Network

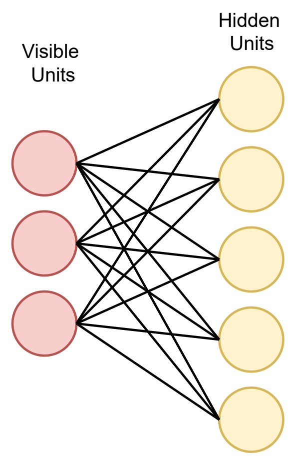

Fig. 5: RBM architecture

2) Deep Belief Networks (DBNS): The structure of a deep belief network (DBN) includes a sequence of restricted Boltzmann machines (RBMs), and they are connected sequentially [14]. Understanding RBMs is crucial for a thorough knowledge of DBNs. Thus, the rest of this section explains the RBMs and DBNs, respectively. The structure of an RBM consists of two units, which are the visible and hidden units [15]. Also, these two units are fully connected with both forward and backward connections. In an RBM structure, observable data is represented in one layer of visible units, while the hidden unit is used in terms of capturing dependencies of the visible units. Fig. 5 illustrates the structure of the employed RBMs. In an RBM, the weights related to each neuron are randomly initialized, and then the weights in a layer are updated based on the conditions of the visible and hidden units in the other layer. This is repeated until the system is sampling from its equilibrium distribution.

Deep Belief Networks (DBNs) are made up of multiple RBMs and a classifier [16]. A DBN can be considered a stacked model because it uses more than one RBM in its creation. Fig. 6 depicts the DBN architecture that was used. It

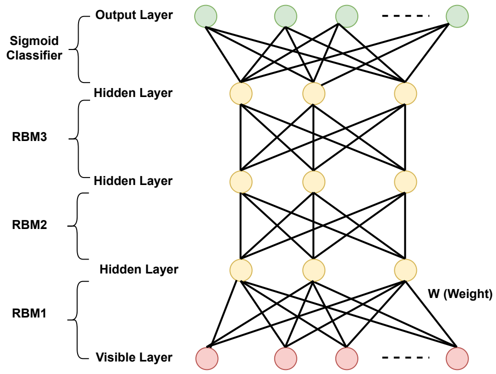

Fig. 6: The structure of the employed DBNs

TABLE I: Definition of TP, FP, TN, and FN

		Actual Class	
		Positive	Negative
Predicted Class	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

TABLE II: Evaluation results of the models

Model	Accuracy	Precision	Recall	F1-Score
DNN	80.1	81.1	98.4	0.725
DBNs	81.2	81.2	99.1	0.728

contains three RBMs and a Sigmoid classifier. In the creation of the employed DBN structure, two and four RBMs were used, except for the three RBMs to obtain the optimum number of RBMs. In the end, the best performance was obtained through the use of three RBMs.

C. Scoring Metrics

The following scoring parameters are used in the evaluation of the employed deep belief networks (DBNs) and deep neural networks (DNNs). Formulas for the used scoring parameters are presented below [17]. The definition of TP, FP, TN, and FN is presented in Table I.

- Accuracy = (TN + TP)/(TN+TP+FN+FP),
- Precision = TN/(TN + FP),
- Sensitivity (Recall) = TP/(TP + FN)
- F1-Score = 2*((precision*recall)/(precision+recall))

IV. Results And Discussion

A. Evaluation Of The Models

Table II presents the evaluation results of the employed deep neural networks (DNNs) and deep belief networks (DBNs) using Dataset I. Detailed information about the dataset was given in Sec. III-A. As it can be seen from Table II, DBNs provided better performance in terms of accuracy (81.2%), precision (81.2%), recall (100%), F1-score (72.8%), and Jaccard (81.2%). Even if the used DBNs provided better performance in the classification of fully-connectedness state analysis, the differences between the DBNs and DNN in terms of scoring metrics are not large.

TABLE III: Evaluation results of the models for different datasets

Dataset	Model	Accuracy	Precision	Recall	F1-Score
Dataset 2	DNN	81.6	84.4	92.7	89.4
	DBNs	84.4	86.4	100	91.5
Dataset 3	DNN	77.4	77.4	90.4	87.3
	DBNs	80.2	85	100	87.6

B. Evaluation Of The Models With Different Datasets

To reveal the efficiency of the employed DBNs in the prediction of whether a given WSN is fully connected, two different datasets were also created. Detailed information about these two datasets (Dataset II and III) was given in Section III-A.

The efficiency of DBNs is compared with that of DNNs in the prediction of the fully-connectedness of WSN based on datasets II and III. Table III shows the prediction performance scores of DBNs and DNNs. The results indicate that DBNs outperformed DNN if WSN is fully connected for database II and III.

In other words, DBNs have the ability to provide more accurate performance than DNNs. On the other hand, the overall prediction performance of DBNs for dataset III is considerably worse than the prediction performance of DBNs for dataset II. Moreover, the overall performance of DNN for Dataset III is also worse than the prediction performance of DNN for Dataset III.

It should be highlighted that, the number of features in datasets II and III is different. Initially, DNN was created based on Dataset II. Then, the created DNN was also used in the prediction of the connection state of the WSN network. using Dataset III. Due to the different number of features in the dataset, the number of neurons in the hidden layers was arranged to create an optimal DNN structure for Dataset III. As a result, the arranged DNN structure provides better performance than the previous DNN structure created for Database II. Table III shows the scoring performance of the arranged DNN.

C. General Discussion

The computational complexity of NNs rises exponentially with node count. Therefore, this turns into a challenging problem to resolve for constrained WSN devices and new computational methods are required to be employed and implemented. Though there are specialized libraries for use with limited capability and computing platforms [18]. Although there are specialized libraries for use with limited capabilities and computing platforms [18], the training phase of machine learning algorithms constitutes the dominant part of the high computational cost. Once the training is complete, the testing stage is a quick process that generally produces the desired results quickly and at a lower computational cost.

We've used NN models up to 121 nodes, beyond which a reasonable amount of time is not possible. For this reason, relatively small-sized WSNs were chosen. Nevertheless, increasing the number of nodes within the WSN does not endanger the forecast performance of the NN model, as the outcome. Consequently, it's possible to assume that the parameters of the more extensive networks can also be forecasted likewise. New optimization methods might be suggested to be able to analyze larger topologies in future studies. As highlighted in Section I, to the best of the authors' knowledge, there is no research that has investigated the WSN connectivity. Thus, this study revealed that DBNs can be used in the connectivity prediction for a WSN with given parameters.

V. Conclusion

A WSN needs to be fully connected so as to collect and transmit the data to the sink node. As a node fails or dies, WSN connectivity may be questioned as to whether it operates properly. In this regard, it is critical to justify a given topology in relation to WSN connectivity.In this sense, the prediction of WSN connectivity plays a key role in determining if WSN is still connected or disconnected. With the motivation that connectivity estimation can be achieved with higher accuracy and precision at a lower computational complexity, we suggested a hybrid solution based on Deep Belief Networks. Algorithms with deep architectures mostly have advantages compared to single-based algorithms in terms of robust results and higher accuracy. Therefore, Deep Belief Networks (DBNs) and Deep Neural Network (DNN) classification algorithms were employed in this study.Three different datasets were used to compare the scoring performance of these algorithms. DBNs provided better scoring values in the classification of WSN connectivity decisions than DNNs (see Table II and III) in the use of these three datasets. The findings reflect the importance of DBN compared to DNN in the classification of WSN lifetimes. As a further study, a hybrid deep neural network can be developed to predict WSN connectivity.

References

[1] M. Sheikh-Hosseini and S. R. S. Hashemi, "Connectivity and coverage constrained wireless sensor nodes deployment using steepest descent and genetic algorithms," *Expert Systems with Applications*, p. 116164, 2021.
[2] M. R. Senouci and A. Mellouk, "A robust uncertainty-aware cluster-based deployment approach for wsns: Coverage, connectivity, and lifespan," *Journal of Network and Computer Applications*, vol. 146, p. 102414, 2019.
[3] N. Aitsaadi, N. Achir, K. Boussetta, and G. Pujolle, "Artificial potential field approach in wsn deployment: Cost, qom, connectivity, and lifetime constraints," *Computer Networks*, vol. 55, no. 1, pp. 84–105, 2011.
[4] S. Sengupta, S. Das, M. Nasir, and B. K. Panigrahi, "Multi-objective node deployment in wsns: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 405–416, 2013.
[5] C. Sevgi and A. Koçyiğit, "Optimal deployment in randomly deployed heterogeneous wsns: A connected coverage approach," *Journal of Network and Computer Applications*, vol. 46, pp. 182–197, 2014.
[6] A. Akbas, H. U. Yildiz, and B. Tavli, "Data packet length optimization for wireless sensor network lifetime maximization," in *2014 10th International Conference on Communications (COMM)*. IEEE, 2014, pp. 1–6.
[7] O. G. Uyan, A. Akbas, and V. C. Gungor, "Machine learning approaches for underwater sensor network parameter prediction," *Ad Hoc Networks*, vol. 144, p. 103139, 2023.
[8] A. Akbas, H. U. Yildiz, A. M. Ozbayoglu, and B. Tavli, "Neural network based instant parameter prediction for wireless sensor network optimization models," *Wireless Networks*, vol. 25, no. 6, pp. 3405–3418, 2019.
[9] "Gams," https://www.gams.com/products/gams/gams-language/, 12 2021, (Accessed on 12/12/2021).
[10] "Log distance path loss or log normal shadowing model - gaussianwaves," https://www.gaussianwaves.com/2013/09/log-distance-path-loss-or-log-normal-shadowing-model/, 12 2021, (Accessed on 12/12/2021).
[11] A. Akbas, H. U. Yildiz, B. Tavli, and S. Uludag, "Joint optimization of transmission power level and packet size for wsn lifetime maximization," *IEEE Sensors Journal*, vol. 16, no. 12, pp. 5084–5094, 2016.
[12] "Matlab - mathworks - matlab & simulink," https://www.mathworks.com/products/matlab.html, 12 2021, (Accessed on 12/12/2021).
[13] M. Gentil, A. Galeazzi, F. Chiariotti, M. Polese, A. Zanella, and M. Zorzi, "A deep neural network approach for customized prediction of mobile devices discharging time," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.
[14] C.-H. Zhu and J. Zhang, "Developing soft sensors for polymer melt index in an industrial polymerization process using deep belief networks," *International Journal of Automation and Computing*, vol. 17, no. 1, pp. 44–54, 2020.
[15] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks," *Communications of the ACM*, vol. 54, no. 10, pp. 95–103, 2011.
[16] Y. Qin, X. Wang, and J. Zou, "The optimized deep belief networks with improved logistic sigmoid units and their application in fault diagnosis for planetary gearboxes of wind turbines," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3814–3824, 2018.
[17] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.
[18] D. Justus, J. Brennan, S. Bonner, and A. S. McGough, "Predicting the computational cost of deep learning models," in *2018 IEEE international conference on big data (Big Data)*. IEEE, 2018, pp. 3873–3882.

Biographies

Ayhan Akbas holds a Bachelor of Science and a Master of Science in Electrical and Electronics Engineering, both obtained from the Middle East Technical University in 1991 and 1995, respectively. He earned his doctoral degree from the Department of Computer Engineering at TOBB University of Economics and Technology. With over 20 years of experience in the field of Information Technology, Dr. Akbaş has occupied both technical and managerial roles at multinational corporations, including SIEMENS, Sun Microsystems, and NEC. Currently, he is pursuing his career as Asst. Professor at Abdullah Gül University. His areas of interest are Wireless Sensor Networks, IoT, 5G, Wireless Communication, and Machine Learning.

Selim Buyrukoglu is an Asst. Prof. in Computer Engineering Department at Cankiri Karatekin University. He completed his PhD at Loughborough University, UK in 2019, He received an MSc degree in Advance Computer Science in 2014 from Leicester University, UK and also a BSc degree in Computer Engineering in 2010 from European University of Lefke, TRNC. He is particularly focusing on the application of Machine Learning and Deep Learning methods on interdisciplinary subjects.