Wireless Networks manuscript No.
(will be inserted by the editor)

Neural Network based Instant Parameter Prediction for Wireless Sensor Network Optimization Models

Ayhan Akbas · Huseyin Ugur Yildiz · Ahmet Murat Ozbayoglu · Bulent Tavli

Received: date / Accepted: date

Abstract Optimal operation configuration of a Wireless Sensor Network (WSN) can be determined by utilizing exact mathematical programming techniques such as Mixed Integer Programming (MIP). However, computational complexities of such techniques are high. As a remedy, learning algorithms such as Neural Networks (NNs) can be utilized to predict the WSN settings with high accuracy with much lower computational cost than the MIP solutions. We focus on predicting network lifetime, transmission power level, and internode distance which are interrelated WSN parameters and are vital for optimal WSN operation. To facilitate an efficient solution

Ayhan Akbas
Department of Computer Engineering
University of Turkish Aeronautical Association
06790, Ankara, Turkey
E-mail: aakbas@thk.edu.tr

Huseyin Ugur Yildiz
Department of Electrical and Electronics Engineering
TED University
06420, Ankara, Turkey
E-mail: hugur.yildiz@tedu.edu.tr

Ahmet Murat Ozbayoglu
Department of Computer Engineering
TOBB University of Economics and Technology
06560, Ankara, Turkey
E-mail: mozbayoglu@etu.edu.tr

Bulent Tavli
Department of Electrical and Electronics Engineering
TOBB University of Economics and Technology
06560, Ankara, Turkey
E-mail: btavli@etu.edu.tr

for predicting these parameters without explicit optimizations, we built NN based models employing data obtained from an MIP model. The NN based scalable prediction model yields a maximum of 3% error for lifetime, 6% for transmission power level error, and internode distances within an accuracy of 3 meters in prediction outcomes.

Keywords Wireless sensor networks · neural networks · multi-layer perceptron · backpropagation · maximum lifetime · lifetime prediction · transmission power level · internode distance

1 Introduction

The idea of monitoring a predetermined area remotely through the use of a plurality of sensor nodes that have wireless communication capabilities is so enticing that for more than two decades Wireless Sensor Networks (WSNs) have been in the focus of the academic and industrial research. WSNs are composed of sensor node platforms which are capable of measuring physical quantities, processing the raw data, and communicating the processed data towards a base station. Although, it is possible to create a WSN topology where all sensor nodes communicate directly with the base station, typically, sensor nodes convey the data they acquire from the environment via other sensor nodes acting as relays [2].

WSN nodes, typically, operate on batteries and deployed in environments where replacing the batteries is not convenient (or in certain deployments is not feasible). Therefore, prolonging the WSN lifetime by reducing the energy dissipation as much as possible is one of the most important objectives in WSN design. Energy dissipation of a WSN node can broadly be categorized into the computation and communication energy dissipation terms. Communication energy dissipation is shown to be the dominant energy dissipation term through direct experimentation. Therefore, reducing the energy dissipation of communication is a priority for WSN lifetime maximization [23].

It is possible to abstract the data flow from sensor nodes towards the base station as a network flow problem. Indeed, by transforming the network topology into a graph it is possible to formulate the WSN network lifetime maximization problem as a mathematical programming model. In fact, Mixed Integer Programming (MIP) is a well known and widely employed technique in literature to model WSN optimization problems [71].

Once a problem is expressed as an MIP then the solution of an instance of the problem is guaranteed to be the optimal solution (*i.e.*, not an approximate solution). However, problems modeled with MIP are, generally, NP-complete problems. Therefore, MIP models cannot be solved to optimality beyond small problem instances. For the solution of large optimization problems heuristic algorithms are utilized. Meta-heuristic algorithms are a particularly useful and convenient category for the approximate solutions of optimization problems [69].

Neural Networks (NNs) are one of the most popular meta-heuristic learning algorithms. In fact, an NN, which is built by interconnecting decision making units such as radial basis functions, learns from the data. Indeed, one of the most powerful aspects of NNs is that they can learn and recognize complex relationships in the data utilized to train them [4].

In this study, we utilized an exact mathematical programming model (*i.e.*, MIP) and a meta-heuristic model (*i.e.*, NN) jointly to create a novel framework for the rapid prediction of network lifetime, transmission power level, and internode distance in WSNs. In fact, the optimal network resource allocation to maximize the WSN lifetimes is obtained by using the MIP model. NN is trained by using this data. Hence, we created a very useful framework by utilizing the best features of the exact and heuristic algorithms (*i.e.*, instantly obtained high accuracy predictions). More precisely stated, our model can predict exactly one of the parameters of network lifetime, transmission power level, or internode distance when all the other parameters are predetermined (*e.g.*, network lifetime is predicted when all the other parameters including the transmission power level and internode distance are predetermined).

As stated in the preceding paragraphs, network lifetime is one of the ultimate Quality-of-Service (QoS) performance metrics in WSNs, hence, prolonging the network lifetime is the common objective in most of the WSN research studies. Both the power level and the internode distance parameters are directly related to the network lifetime. Increasing the internode distance results in higher communication energy which in turn reduces the network lifetime. Likewise, increasing the transmission power results in higher communication energy and reduction in network lifetime. However, choosing a power level lower than the minimum required level will result in the loss of network connectivity. Nevertheless, we can come up with a prediction model (just like linear or nonlinear regression) where we use two of the parameters and estimate the third one. Hence, accurate prediction of these three parameters is of paramount importance for WSNs.

1.1 Scope and Contributions

Abstraction of WSNs through graphs and modeling the data flows through MIP is a commonly utilized approach in the literature. In this manuscript, we employ an MIP model to maximize the network lifetime of WSNs by adopting a detailed energy dissipation model and an experimentally verified path loss model. Although, the optimal solution for a given network configuration can be obtained by the solution of the MIP model, the time required for the termination of such a solution is prohibitively high. As a remedy, we proposed an NN based framework for the instant prediction of WSN parameters (network lifetime, internode distance, power level). Performance evaluations reveal that the proposed NN based approach can predict WSN parameters with high accuracy in significantly shorter computation times.

The rest of the paper is organized as follows: Section 2 overviews the related work. Section 3 presents the system model, the MIP framework, in detail, and a brief overview of NNs. Numerical analyses are provided in Section 4. We discuss various aspects of this study in Section 5. Conclusions are given in Section 6.

2 Related work

Machine learning (ML) models are used, extensively, to create prediction models for many WSN applications [3]. In [59], a learning-based self adapting medium access control (MAC) protocol is proposed which optimally selects MAC protocols by considering reliability, energy consumption, latency, and resiliency against noise. In [75], authors propose a framework called pTUNES that adapts link, topology, and traffic patterns according to the application specific requirements (*e.g.*, reliability, network lifetime, and latency) by using a predictive control model. In [35], authors use ML to predict the link quality as well as packet reception ratio (PRR). The inputs of the ML model consist of physical layer parameters of last received packets and PRR. In [30], ML is used to characterize the communication performance of the carrier-sense multiple access with collision avoidance (CSMA/CA) MAC layer and to estimate packet delivery ratio as a function of the collected measurements.

In the literature, genetic algorithms (GA), evolutionary search, and particle swarm optimization (PSO) methods are employed to improve the network lifetime [9,20,45], communication efficiency [48], and cluster head (CH) selection [9,45] for WSNs. In [20], intelligent routing protocols based on reinforcement learning, ant colony optimization, fuzzy logic, GA, and neural networks (NNs) are investigated for prolonging the WSN lifetime. In [48], an ML-based GA algorithm is proposed to adjust the transmission rate of a dynamic WSN to optimize the communication efficiency. GA is also used in [45] for CH selection which can prolong the network lifetime. In this work, the aim is to apply clustering on the WSN nodes for load balancing through optimum selection of CHs. A similar problem is also studied in [9] where a GA based distance-aware routing protocol is proposed that jointly optimize CH selection and network lifetime. In [26], a PSO based method is proposed to design an energy efficient and improved lifetime path for a mobile sink in WSNs where pareto dominance is considered to select global and local best solutions for each particle in the swarm. In [51], a hybrid evolutionary algorithm is developed to investigate the optimal trade-off between network lifetime and robustness in multi-path routing schemes.

There is a significant amount of research studies that focus on NNs in various WSN scenarios [1, 15, 64]. Most of these works focus on data sampling [5], localization [46, 61, 62], node clustering [19], and energy-aware routing [20]. We present an overview of the selected publications as follows. In [5], authors propose a multi-layer perceptron (MLP) model (a class of feed-forward artificial NNs) to achieve a smart sampling solution for reducing the time spent for transmissions, hence, reducing the energy consumed in WSNs. In [61, 62], MLP, Radial Basis Function (RBF), and Back Propagation Network (BPN) artificial NN models are utilized for localization applications. In these studies, time difference of arrival information is used to calculate the distance from anchor nodes to sensor nodes. The distance information is used to train artificial NN models. In [19], a BPN based CH selection model is developed which relates node weights directly to the decision-making predictions. The proposed solution reduces the network traffic while extending the network lifetime. In [46], a comparative analysis of conjugate gradient based feed-forward artificial NNs is performed for localization in WSNs.

There is a large and growing body of literature on the parameter prediction for WSNs by using NNs. We commonly observe that some environmental parameters regarding to the specifications of WSNs (*e.g.*, physiological parameters [41], greenhouse parameters [7], soil parameters [50], *etc.*), PRR [8, 66], localization parameters [18], transmission power [55], channel condition parameters [13,32] are estimated throughout these studies. In [41], an artificial NN model is developed to predict the time when to transfer physiological parameters (*i.e.*, arterial pressure, central venous pressure, and pulmonary artery pressure) for maintaining the consumption of energy while keeping the mean square of the estimation error at an acceptable level in wireless body area sensor networks. In [7], a fuzzy NN with proportional–integral–derivative

(PID) control method is proposed for estimating and controlling the greenhouse environmental parameter (*i.e.*, air temperature, humidity, and illumination) changes in WSNs for agricultural applications. In [50], a cluster based routing protocol using an NN data fusion method is proposed to estimate the farmland soil carbon sink parameters (*e.g.*, soil temperature, moisture, pH value, *etc.*). In [34], a distributed clustering routing protocol is proposed for energy harvesting WSNs where an NN based solar energy harvesting prediction model is exploited to ensure the energy efficiency of the proposed protocol. In [55], authors determine the transmission power required for a received signal strength indicator (RSSI) threshold to be satisfied. RSSI values are estimated by using variants of Kalman filter and the estimated RSSI values are used as the inputs to an artificial NN based power control method. In [8], authors use artificial NNs to predict packet losses in indoor and outdoor WSN applications. Similarly, in [66], a wavelet NN method is proposed for estimating the probability-guaranteed limits on the PRR. In [32], authors propose an NN based learning method to estimate the idle period distribution by using the spectrum sensing capabilities of sensor nodes. In [13], a feed-forward NN model is developed to estimate the channel conditions which can offer low latency with high throughput at any time slot.

The most related study to our work is presented in [18] where the authors estimate the distance between mobile sensor nodes and anchor nodes for both indoor and outdoor WSNs. A log-normal shadowing model and a PSO based artificial NN model are used to accurately estimate the internode distance. Although authors consider the physical layer parameters such as RSSI to estimate the internode distance, lifetime and transmission power level prediction are not considered in this work.

Extensive review of the existing literature indicates that there is no NN based study to predict network lifetime, transmission power level, and internode distance in WSNs, in the literature. To the best of our knowledge, our attempt is the first to study the prediction of these WSN parameters using NN models. Our motivation in using NN is due to the fact that as the number of nodes in the network increases, the

MIP models become unpractical due to hardness of the solution of MIP models which, generally, are NP-complete problems having extremely high computation times.

3 System Model

In this section, we present an overview of our system model then we detail the physical and link layer energy consumption models. The MIP model as well as the NN model are also elaborated in this section.

3.1 Overview

Throughout this work we consider a WSN which consists of a single base station and a plurality of sensor nodes which are deployed over a predetermined sensing area [71] in a grid fashion. Let $G(V, A)$ represents the network topology where V is the set of all nodes and A is the set of all links. The base station is node-1 and located at the center of the grid. We define $W = V - \{1\}$ to represent the set of all sensor nodes. We assume that the base station knows the network topology completely and has a sufficiently high processing capacity. A time division multiple access (TDMA)-based MAC layer is operational which prevents possible collisions. At each TDMA round, all sensed data are conveyed to the base station either by using single-hop and/or multi-hop transmission schemes. Time is organized as constant duration rounds where each round is assumed to be 60 seconds (*i.e.*, $T_{rnd} = 60$ s). Indeed, one minute sampling interval is a typical value in certain WSN applications [14]. At each round, $s_i = 1$ data packet is generated which models the sensed data. A two-way handshaking mechanism is active where data packets and acknowledgement (ACK) packets are used between each communicating node pair. The data packet size is 256 Bytes (*i.e.*, $M_P = 256$ Bytes) and ACK packet size is $M_A = 20$ Bytes [71]. The active TDMA communication slot time (T_{slot}) is considered as 115 ms [71]. On each link, transmission power levels are locally optimized as detailed in [71]. The proposed MIP model has an objective to maximize the lifetime of the network (N_{rnd}). The

Table 1: Symbols and their corresponding explanations used throughout this work.

Symbol	Explanation	Symbol	Explanation
N_{rnd}	Network lifetime (in rounds)	T_{rnd}	Round duration (60 s)
f_{ij}	Amount of data packets flowing from node-i to node-j	ϱ	Battery energy of each sensor node (10 KJ)
s_i	Number of data packets generated at node-i at each round (1 packet/round)	E_{PP}	Packet processing energy (120 μJ)
d_{ij}	Distance between node-i and node-j (m)	d_{int}	Internode distance (m)
$P_{tx}^{crc}(l)$	Transmission power at level-l (mW)	$P_{tx}^{ant}(l)$	Antenna power at power level-l (mW)
$\overline{P_{tx}^{crc}(l)}$	Transmission power at level-l (dBm)	$\overline{P_{tx}^{ant}(l)}$	Antenna power at power level-l (dBm)
P_{rx}^{crc}	Reception power (35.4 mW)	ξ	Data rate of Mica2 motes (19.2 Kbps)
E_{DA}	Data acquisition energy (600 μJ)	T_{DA}	Data acquisition time (20 ms)
M_P	Data packet size (255 Bytes)	M_A	ACK packet size (20 Bytes)
T_{slot}	Active TDMA slot time (115 ms)	P_{slp}	Sleep power (3 μW)
$T_{tx}(M_P)$	Duration of an M_P bytes of data packet transmission	$T_{tx}(M_A)$	Duration of M_A bytes of ACK packet transmission
$\overline{PL_0}$	Path loss at the reference distance (31 dB)	d_0	reference path loss distance (1 m)
$\overline{X_\sigma}$	Shadowing variable – $\mathcal{N}(0,\sigma^2)$ (dB)	σ^2	Variance of shadowing (1.42 dB)
n	Path loss exponent (3.69)	$\overline{P_n}$	Noise Floor (–115 dBm)
$\overline{\gamma_{ij}(l)}$	Signal-to-noise ratio at node-j (dBm)	$\overline{P_{rx,ij}^{ant}(l)}$	The received signal power due to a transmission at power level-l over the link (i,j) (dBm)
$p_{ij}^s(l,\varphi)$	Probability of a successful packet reception of a φ Byte packet transmitted at power level-l over the link (i,j)	$p_{ij}^f(l,\varphi)$	Probability of a failure packet reception of a φ Byte packet transmitted at power level-l over the link (i,j)
$p_{ij}^{HS,s}(l,k)$	Probability of a successful handshake	$p_{ij}^{HS,f}(l,k)$	Probability of a failed handshake
$\lambda_{ij}(l,k)$	Packet retransmission rate for the link (i,j)	$\overline{P_{sns}}$	Sensitivity level (–102 dBm)
$E_{tx}^P(l,\varphi)$	Energy for transmitting an φ-Byte packet data at power level-l	$E_{tx}^{HS}(l,M_P)$	Total energy dissipation of a transmitter in a slot
$E_{tx,ij}^D(l,k)$	Energy consumption of a transmitter for completing a handshake including retransmissions	$E_{rx,ji}^D(l,k)$	Energy consumption of a receiver for completing a handshake including retransmissions
$E_{rx}^{HS,s}(k,M_A)$	Energy dissipation of a receiver for a successful handshake	$E_{rx}^{HS,f}$	Energy dissipation of a receiver for a failed handshake due to data packet errors
$T_{act,i}$	Total active time of node-i	I_{jnlk}^i	Interference function at node-i
(l,k)	Power levels for data and ACK packets	S_L	Set of all power levels
$G=(V,A)$	Network topology	A	Set of links
V	Set of all nodes (including the base station)	W	Set of all sensor nodes

network lifetime is defined as the time elapsed until the first node consumes all of its battery energy [70]. Symbols and their corresponding explanations used throughout this paper are provided in Table 1.

3.2 Physical Layer Model

The WSN considered in this work consists of Mica2 sensor nodes which are widely used sensor node platforms in WSN research. In Table 2, we tabulate the operating characteristics of Mica2 motes such as the transceiver power consumption, $P_{tx}^{crc}(l)$, and output antenna power, $P_{tx}^{ant}(l)$, at each power level-l. S_L presents the power levels set. Reception power consumption is constant which is taken as 35.4 mW (*i.e.*, $P_{rx}^{crc} = 35.4$ mW). All the parameters of Mica2 motes' characteristics, reported in

Table 2: Transmission power consumption ($P_{tx}^{crc}(l)$ – mW) and output antenna power ($P_{tx}^{ant}(l)$ – mW) for each power level-l [71].

l	$P_{tx}^{crc}(l)$	$P_{tx}^{ant}(l)$	l	$P_{tx}^{crc}(l)$	$P_{tx}^{ant}(l)$
1	25.8	0.0100	14	32.4	0.1995
2	26.4	0.0126	15	33.3	0.2512
3	27.0	0.0158	16	41.4	0.3162
4	27.1	0.0200	17	43.5	0.3981
5	27.3	0.0251	18	43.6	0.5012
6	27.8	0.0316	19	45.3	0.6310
7	27.9	0.0398	20	47.4	0.7943
8	28.5	0.0501	21	50.4	1.0000
9	29.1	0.0631	22	51.6	1.2589
10	29.7	0.0794	23	55.5	1.5849
11	30.3	0.1000	24	57.6	1.9953
12	31.2	0.1259	25	63.9	2.5119
13	31.8	0.1585	26	76.2	3.1623

this study, are experimentally verified. We adopted all these values from [71] where the original references for each parameter can be found.

We employ a log-normal shadowing channel model as the physical layer model. The signal-to-noise ratio (*i.e.*, $\overline{\gamma_{ij}}(l)$ in dB) at the receiving node-j due to the transmission from node-i with power-l can be expressed as

$$\overline{\gamma_{ij}}(l) = \underbrace{\overline{P_{tx}^{ant}}(l) - \overline{PL_0} + 10n\log_{10}\left(\frac{d_{ij}}{d_0}\right) - \overline{X_\sigma}}_{\overline{P_{rx,ji}^{ant}(l)}} - \overline{P_n}. \tag{1}$$

In this equation, $\overline{P_{tx}^{ant}}(l)$ is transmission power of the antenna with power level-l (in dBm), $\overline{P_{rx,ji}^{ant}(l)}$ is the reception power (in dBm), $\overline{PL_0} = 31$ dB is the reference path loss value (in dB), $n = 3.69$ [39] is the path loss exponent, d_{ij} is the link distance (in meters), $d_0 = 1$ m is the reference distance (in meters), $\overline{X_\sigma}$ is a zero-mean Gaussian random variable with variance $\sigma^2 = 1.42$ dB [39] to model the shadowing (in dB), and $\overline{P_n}$ is the noise floor value which is taken as –115 dBm for Mica2 platforms [71]. The success probability of a φ-Byte packet reception which is transmitted over the link-(i, j) at power level-l when using a non-coherent frequency shift keying modulation and non-return-to-zero encoding (default modulation and encoding schemes for

Mica2 motes) is

$$p_{ij}^{s}(l,\varphi) = \left(1 - \frac{1}{2}\exp\left(\frac{-\gamma_{ij}(l)}{2} \times \frac{1}{0.64}\right)\right)^{8\varphi}. \quad (2)$$

Note that, $\gamma_{ij}(l)$ is the signal-to-noise ratio given in ordinary form. Similarly, packet failure probability is calculated as $p_{ij}^{f}(l,\varphi) = 1 - p_{ij}^{s}(l,\varphi)$. Successful handshake probability when the data packet is transmitted over the link-(i,j) with power level-l and ACKed with power level-k is $p_{ij}^{HS,s}(l,k) = p_{ij}^{s}(l,M_P) \times p_{ji}^{s}(k,M_A)$ as long as $\overline{P_{rx,ij}^{ant}(l)} \geq \overline{P_{sns}}$ and $\overline{P_{rx,ji}^{ant}(k)} \geq \overline{P_{sns}}$. In this notation, the reception sensitivity of the Mica2 motes is $\overline{P_{sns}} = -\ 102$ dBm. Failed handshake's probability will be $p_{ij}^{HS,f}(l,k) = 1 - p_{ij}^{HS,s}(l,k)$. Considering an infinite retransmission mechanism, each data packet, on the average, has to be retransmitted $\lambda_{ij}(l,k) = 1/p_{ij}^{HS,s}(l,k)$ times.

3.3 Link Layer Energy Consumption Model

3.3.1 Energy Consumption of the Transmitting Node

The energy required to transmit M_P Bytes of data from node-i to node-j by using power level-l is $E_{tx}^{P}(l,M_P) = P_{tx}^{crc}(l) \times T_{tx}(M_P)$. In this equation, $T_{tx}(M_P) = M_P/\xi$ is the time to transmit an M_P-Byte data packet where $\xi = 19.2$ kbps is the data rate of the Mica2 mote platforms [71]. In the rest of the slot time, $T_{slot} - T_{tx}(M_P)$, the transmitter node will stay in reception mode where $P_{rx}^{crc} \times (T_{slot} - T_{tx}(M_P))$ of energy is dissipated. Hence, the total energy dissipation of a transmitter in a single successful handshake is expressed as $E_{tx}^{HS}(l,M_P) = E_{tx}^{P}(l,M_P) + P_{rx}^{crc} \times (T_{slot} - T_{tx}(M_P))$. Considering retransmissions and packet processing costs (*i.e.*, $E_{PP} = 120\ \mu$J [71]), the transmitter node's energy dissipation can be expressed as

$$E_{tx,ij}^{D}(l,k) = E_{PP} + \lambda_{ij}(l,k) \times E_{tx}^{HS}(l,M_P). \quad (3)$$

3.3.2 Energy Consumption of the Receiving Node

In a successful handshake during a single slot time, $P_{rx}^{crc} \times (T_{slot} - T_{tx}(M_A))$ of energy is consumed to receive an M_P-Byte data packet. Note that, $T_{tx}(M_A) = M_A/\xi$ is the time to transmit an M_A-Byte ACK packet. As soon as the data packet is received, the ACK packet is transmitted with power level-k which costs $E_{tx}^{P}(k, M_A) = P_{tx}^{crc}(k) \times T_{tx}(M_A)$ of energy. The total energy dissipation in this case when considering the retransmissions can be expressed as $E_{rx}^{HS,s}(k, M_A) = P_{rx}^{crc} \times (T_{slot} - T_{tx}(M_A)) + E_{tx}^{P}(k, M_A)$. If the handshake fails due to the ACK packet errors in the reverse communication channel, the handshaking process must be repeated where $E_{rx}^{HS,s}(k, M_A)$ should be multiplied by the factor of $\lambda_{ij}(l, k)p_{ij}^{s}(l, M_P)p_{ji}^{f}(k, M_A)$ including the retransmissions. However, if the handshake fails due to the received data packet errors, then the receiving node will only wait in the reception mode which costs $E_{rx}^{HS,f} = P_{rx}^{crc} \times T_{slot}$ of energy. To sum up, the energy dissipation of the receiver including retransmissions and packet processing costs can be expressed as

$$\begin{aligned} E_{rx,ji}^{D}(l,k) = E_{rx}^{HS,s}(k, M_A) + E_{PP} \\ + \lambda_{ij}(l,k)p_{ij}^{s}(l, M_P)p_{ji}^{f}(k, M_A)E_{rx}^{HS,s}(k, M_A) \\ + \lambda_{ij}(l,k)p_{ij}^{f}(l, M_P)E_{rx}^{HS,f}. \end{aligned} \tag{4}$$

3.4 MIP Framework for WSN Lifetime Maximization

$$\text{Maximize } N_{rnd} \tag{5a}$$

s.t.

$$\sum_{(i,j)\in A} f_{ij} - \sum_{(j,i)\in A} f_{ji} = N_{rnd} \times s_i, \ \forall i \in W \tag{5b}$$

$$T_{act,i} = T_{slot}\left[\sum_{(i,j)\in A} \lambda_{ij}(l,k)f_{ij} + \sum_{(j,i)\in A} \lambda_{ji}(l,k)f_{ji}\right] + N_{rnd} \times T_{DA}, \ \forall i \in W \tag{5c}$$

$$\underbrace{\sum_{(i,j)\in A} E^D_{tx,ij}(l,k) f_{ij}}_{\textbf{transmission}} + \underbrace{P_{slp}(N_{rnd} \times T_{rnd} - T_{act,i})}_{\textbf{sleep}}$$
$$+ \underbrace{\sum_{(j,i)\in A} E^D_{rx,ji}(l,k) f_{ji}}_{\textbf{reception}} + \underbrace{N_{rnd} \times E_{DA}}_{\textbf{acquisiton}} \leq \varrho,\ \forall i \in W \tag{5d}$$

$$T_{slot}\left[\sum_{(i,j)\in A} \lambda_{ij}(l,k) f_{ij} + \sum_{(j,i)\in A} \lambda_{ji}(l,k) f_{ji} + \sum_{(j,n)\in A} \lambda_{jn}(l,k) f_{jn} I^i_{jn}(l,k)\right]$$
$$\leq N_{rnd} \times T_{rnd},\ \forall i \in V \tag{5e}$$

$$f_{ij} \geq 0,\ \forall (i,j) \in A \tag{5f}$$

The MIP model that maximizes the network lifetime is presented in (5). The MIP model is, actually, a network flow formulation with side constraints and it is based on the MIP model introduced in [71]. The objective function of the MIP framework is to maximize the lifetime of the network which is defined as N_{rnd} (in rounds). The network lifetime in terms of seconds can be defined as $N_{rnd} \times T_{rnd}$. The decision variables of the MIP model are the amount of data packets flowing from node-i to node-j which are represented by the integer variables f_{ij}. The constraints of the MIP model are expressed by Eqs. (5b)–(5f).

Constraint (5b) is the *flow balancing constraint* which states that at each sensor node (*i.e.*, $\forall i \in W$), the data flowing into node-i (*i.e.*, $\sum_{(j,i)\in A} f_{ji}$) plus the data generated by node-i throughout the lifetime (*i.e.*, $N_{rnd} \times s_i$) must be equal to the data flowing out of node-i (*i.e.*, $\sum_{(i,j)\in A} f_{ij}$). Constraint (5c) is used to calculate the active time of node-i (*i.e.*, $T_{act,i}$). In this constraint, $T_{DA} = 20$ ms is the data acquisition time [71]. Constraint (5d) is the *energy balancing constraint* which states that the total energy dissipation of a sensor node includes the energy costs of transmission, reception, sleep, and acquisition is bounded by the energy stored in batteries ($\varrho = 10$ KJ). In this constraint, sleep power is taken as $P_{slp} = 30$ μW and data acquisition energy dissipation at each round is taken as $E_{DA} = 600$ μJ [71]. Network bandwidth is bounded by the network lifetime in terms of aggregate durations of flows in constraint (5e). In this constraint, $I^i_{jn}(l,k)$ is defined as the interference function which takes value of 1 if node-i is in the interference region at power level-l for the data transmission from node-j to node-n, or at power level-k for the ACK transmission from node-n to node-j. Mathematically speaking, $I^i_{jn}(l,k) = 1$ if $\overline{P^{ant}_{rx,ji}(l)} \geq \overline{P_{sns}}$ or $\overline{P^{ant}_{rx,ni}(k)} \geq \overline{P_{sns}}$. Otherwise it is zero (*i.e.*, $I^i_{jn}(l,k) = 0$). Finally, constraint (5f) states that flows should be non-negative.

3.5 Neural Networks

NNs are extensively used for various pattern recognition and data analysis problems [21]. The scope of the problems covered by NNs has a wide range, such as handwritten character recognition [11], stock market forecasting [49, 58], traffic sign detection [63], aviation [25], manufacturing [57], fingerprint identification [10], speech recognition [60], automatic vehicle classification [56], sentiment analysis [12], flow pattern classification [44], license plate recognition [27], *etc.* One major reason why the researchers choose to use NNs is the ability of NNs to generalize and learn from complex patterns where traditional approaches such as linear/nonlinear regression and/or function approximation methodologies are not adequate to address the complexity of the problem. Most of these aforementioned problems have complex feature representations which cannot be easily modeled by classical techniques.

NNs (more generally ML) have also been attracting a lot of interest in wireless next-generation networks including cognitive radios, massive multiple-input and multiple-output (MIMO) systems, smart grid, heterogeneous networks, energy harvesting, and so on [24]. In general, ML methods can be categorized as supervised and unsupervised learning. The emerging reinforcement learning is also included as a new category for ML. Supervised learning is used to build a system model by using predefined inputs and known outputs. Development of an algorithm that is trained with the signal propagation characteristics to estimate the location of a node is considered as an example of supervised learning [53]. Recent works show that spectrum sensing and channel estimation can be performed by using supervised learning for cognitive radios [74]. Moreover, MAC protocol development, link quality estimation, localization, *etc.* can also be performed with supervised learning [28]. In contrast, unsupervised learning is used to find similarities in the input data where the learner is provided with inputs and unknown outputs. Existing research show that unsupervised learning can be used in clustering [73] and load balancing in heterogeneous networks as well as data aggregation [72] and event detection in wireless networks [37]. Finally, reinforcement learning is used for dynamic iterative learning and decision-making process. Distributed resource allocation in small-cell networks [17] and development of energy harvesting models can be performed by using reinforcement learning [43].

The applications of Deep Learning (DL) in different layers of wireless networks are overviewed in [38]. Note that, DL is a branch of ML such that ML and DL problems are solved with NNs. It is shown that DL provides much improvements in wireless network applications when compared to classical ML algorithms due to its high prediction accuracy and no pre-processing complexity of the input data [38]. DL can be used for interference alignment [22], spectrum sensing [29, 52], and classification of modulation schemes at physical layer [47]. Moreover, traffic prediction [42] and channel resource allocation [33, 65] can be performed via DL at the data link layer. Optimal routing planning [68] and traffic load balancing are also two possible network layer applications of DL. Furthermore, DL is an effective method for identification of the traffic and applications at upper layers [38].

As exemplified in the preceding paragraphs, NN models have been used in a wide variety of implementations in WSN literature. The difference of our paper is to provide a fast solution to the network lifetime optimization process using a feedforward NN model. LP, MIP, or metaheuristics approaches exist for this particular problem in the literature. But these solutions can be prohibitively costly with increasing network size. Meanwhile, a feedforward NN, by its nature, is not an optimization technique, instead, it is used in classification or regression problems. However, once trained, calculating the result from given inputs is extremely fast which is our main motivation (*i.e.*, to come up with fast network lifetime optimization for a given network). To best of our knowledge, such an approach has never been adapted in the literature.

We preferred to use MLP with backpropagation learning algorithm due to its simplicity in implementation and wide usage in literature for various application areas. MLP is not the best performing NN model and deep learning models generally outperform traditional NN models like MLP, however, they require more data. Since our study is among the first of its kind, we opted to have a very general model for the proof-of-work. For future implementations, we will adapt different NN models, deep feedforward MLPs and autoencoders for different WSN topologies.

Even though there are several different NN models and topologies, one particular model stands out as the most commonly used NN, which is the MLP using BP Learning Algorithm [54]. MLP model is generally preferred due to its relatively simple implementation, superior performance over a wide range of problems and well-studied working structure [36]. MLP is a commonly used feedforward NN model [6]. As a result, in most of the studies, when the researchers mention they use NNs in their analysis without explicitly specifying which model, they actually refer to MLP, in most case [31].

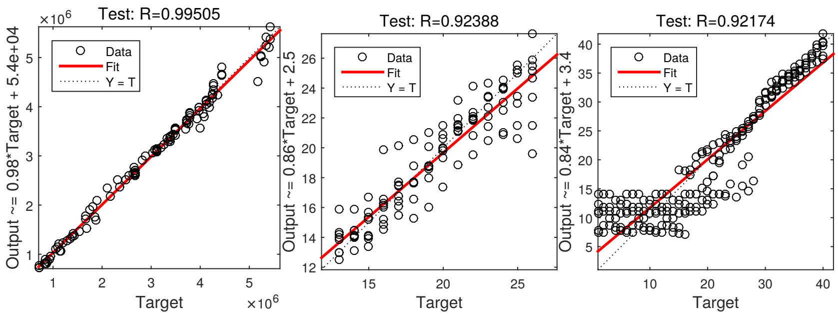
Since MLP is extensively studied and covered in literature, we will only provide a very basic representation of the model. MLP and all NN models use neurons as their basic building block. Each neuron is connected to every neuron in the preceding and following layer. Such a representation is called a fully-connected network. MLP is fully connected, but there are some other NN types that are not fully connected. In MLP, data is represented as features to the model through the input layer. The number of neurons in the input layer will be the same as the feature vector size. A number of hidden layers (mostly one or two) exist. Each layer has a number of hidden neurons depending on the size of the data set and the complexity of the problem. The number of neurons in the output layer is equal to the number of outputs in the data representation. In our particular case, it is just one (network lifetime). Generally speaking, in regression problems (like network lifetime prediction), there is a single neuron in the output layer, whereas in classification problems (like character recognition), the number of neurons in the output layer equals to the number of different classes defined in the data set.

4 Experiments

We investigate the performance of the MIP framework by using General Algebraic Modeling System (GAMS) with CPLEX solver [16] and NN solution by using MATLAB Neural Network (NN) toolbox [67]. To collect WSN data for NN prediction, we utilize a 25-node grid network topology, assuming internode distances (*i.e.*, d_{int}) are ranging from 1 to 40 meters. 26 different transmission power levels (presented in Table 2) are utilized. Note that we explore WSNs with varying number of nodes (up to 121 sensor nodes) in the later parts of this Section.

In our analysis, we utilize 100 random scenarios in which path loss values are regenerated randomly at each run. We present the mean of the results to simulate the effects of log-normal shadowing on each run. A data set of 560 values which contain lifetime, power levels, and distance values is used. More specifically stated, we acquired the data set to be used to train the NN by running the MIP model in subsection 3.4 for each network configuration. We use three separate prediction models. In the first model, distance values and power levels are presented as inputs and network lifetime is predicted (see Figure 1a). In the second model, lifetime values and distance values are presented and power level is predicted (see Figure 1b). The last model use network lifetime values and power levels as inputs and internode distance is predicted (see Figure 1c). One hidden layer with ten neurons is chosen as the MLP topology for all selected models. The hidden neurons use a sigmoid activation function, whereas the output neuron has a linear activation function. 60% of the data is used for training, 20% is used for Cross Validation (CV), and the remaining 20% is spared for testing purposes in all prediction models. A 5-fold CV and testing is implemented in order to test all available data. As the stopping criteria for the network training process, the early stopping with CV error increase criterion is adapted in order to avoid overfitting. All models stop their training process within 20 epochs.

The ground truth in our analysis is the results obtained by the MIP model presented subsection 3.4. Since we already have the optimum results through the utilization of the MIP model which, in fact, gives exactly the same results with the MIP model provided in [71], we, explicitly, provide comparisons with [71]. However, by definition, the results obtained by our NN-based approach cannot surpass the optimum for any given configuration, yet, our results are within close neighborhoods of the optimal results with significantly shorter computation times.



(a) Lifetime prediction (b) Power level prediction (c) Distance prediction

Fig. 1: NN test correlation results

After completion of the prediction tests, results are checked for the statistical significance by paired T-tests. T-test p-values along with other statistical metrics (*e.g.*, root-mean-square-error – RMSE, mean-percent-error – MPE, mean-absolute-percent-error – MAPE, and correlation coefficient – R) are tabulated in Table 3. T-test p-values above 0.05 indicate that difference between the actual values and the predicted ones are statistically insignificant. Overall regression tests for lifetime, power level, and distance show that network lifetime prediction, power level prediction, and internode distance prediction correlations are 0.995, 0.924, and 0.922, respectively.

Table 3: Prediction performance of NN models.

Model	RMSE	MPE	MAPE	R	T-test p-value
Lifetime	119924	0.116	3.354	0.99577	0.959
Powerlevel	1.788	–0.9145	6.2275	0.92388	0.734
Distance	4.507	-46.105	63.04	0.92174	0.921

Table 4: Prediction performance of linear regression models.

Model	RMSE	MPE	MAPE	R	T-test p-value
Lifetime	729379	11.92	29.71	0.8254	0.863
Powerlevel	4.0198	4.64	19.155	0.0784	0.985
Distance	6.5296	58.765	81.75	0.8246	0.814

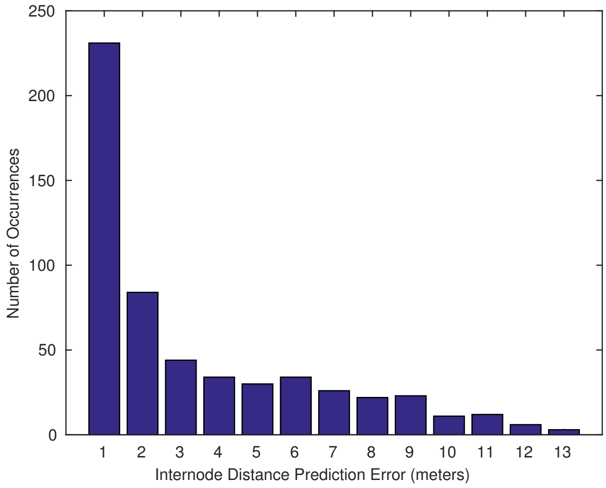


Fig. 2: Internode distance prediction performance histogram

We also compare the NN results against a Linear Regression (LR) Model for each prediction case (*i.e.*, lifetime prediction, power level prediction, and internode distance prediction). The prediction performance statistics for the LR Model are tabulated in Table 4, whereas the NN prediction performance statistics are provided in Table 3. The results indicate that the NN model outperformed the LR Model in all three cases (*i.e.*, lifetime and power level prediction performances of NN models are more accurate than the LR models). The overall prediction performance of the internode distance is considerably worse than the other two prediction models, mostly due to the difficulties in estimating smaller node distances that are closer to each other. This phenomena can easily be observed from Figure 1c where the prediction values for smaller distances at the lower left part of the graphs are far from the ideal prediction line, whereas larger distances (at the upper right corner) show much better prediction performance. However, even in that case, NN model still is much better than the LR model as observed from Tables 3 and 4.

Figure 2 shows the distribution of error in distance predictions in meters. The figure reveals that the actual error distance of NN model is insignificant from RF communication point of view as most of the errors are concentrated within the range of 3 meters. Since we studied internode distances ranging from 1 to 40 meters, small distance values tend to have higher percent errors even though actual error is negligible (*i.e.*, ±1 meter of prediction error for 1 meter internode distance results in 100% prediction error however, such small difference values of distances are not taken into account by RF practitioners.

Table 5: Runtime values in terms of seconds for various grid topologies and internode distances (d_{int}) with their standard deviations (included after ± sign).

d_{int} (m)	Topology				
	Grid: 3×3	Grid: 5×5	Grid: 7×7	Grid: 9×9	Grid: 11×11
10	36±18	228±31	713±99	1633±174	3500±400
20	27±3	155±21	505±267	1294±150	3438±1217
30	20±3	133±18	407±41	1199±152	3424±1174
40	17±4	116±14	408±54	1192±325	2783±499
50	15±3	109±9	403±51	1126±114	2549±294
60	14±3	112±15	367±43	1099±123	2462±262
70	15±4	113±16	415±327	1153±329	2396±192
80	14±3	125±37	406±59	1083±106	2357±195
90	14±4	195±32	397±55	1252±761	2238±77
100	14±3	198±40	368±45	1032±336	2151±56

Meanwhile, our main goal in this study is to provide a model to predict these values without going into deep, time consuming computations of the optimization models with huge number of nodes. To show how computational complexity changes

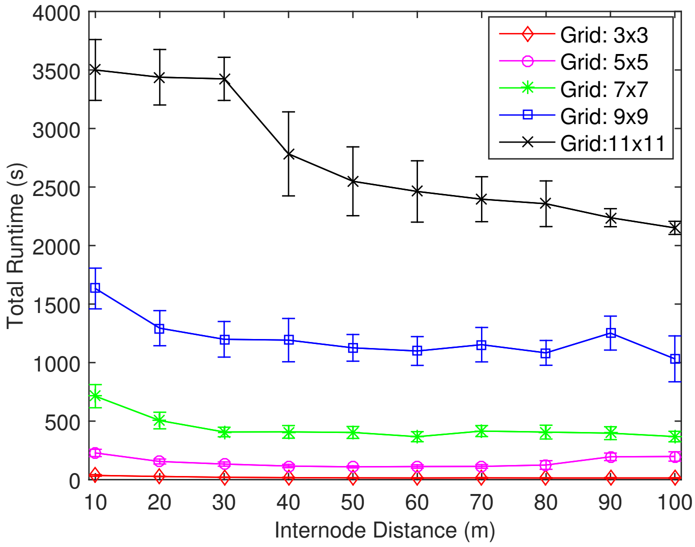


Fig. 3: Runtime values in seconds for various grid topologies and internode distances with their standard deviations (as error bars).

as number of nodes and internode distances increases, we consider the grid topologies of 9 nodes (*i.e.*, Grid: 3×3), 25 nodes (*i.e.*, Grid: 5×5), 49 nodes (*i.e.*, Grid: 7×7), 81 nodes (*i.e.*, Grid: 9×9), and 121 nodes (*i.e.*, Grid: 11×11) as internode distance ranging from 10 to 100 meters with 10 meters of increments.

Total runtime values in seconds for various grid topologies and internode distances with their standard deviations (included after ± sign) are presented in Table 5. As the number of nodes in the network increases, the computing time increases exponentially. As for the internode distance, large internode distances require less time to be computed due to the increasing sparsity which restricts MIP model's feasible solution region into a narrower domain, resulting in less candidate solutions to be searched. To illustrate the sharp increase in the computation time, we plot the values given in Table 5 and in Figure 3. The standard deviations are presented as error bars in this figure. Computation time increases drastically more than 250 fold for 121-node network with 10 meters internode distance compared to 9 nodes network with 100 meters internode distance, resulting in more than 58 minutes of a runtime to calculate the lifetime of a single problem instance on an 8-core Intel i7 machine with 8 GB RAM whereas it is trained in 3-5 seconds with NNs and tested (predicted) within half a second.

5 Discussion

During our implementation, we, actually, tried various NN topologies other than the one we present in this study with different parameter settings such as various numbers of layers and neurons, yet, we opted to use the model which had the best performance and simplest configuration (*i.e.*, the NN topology we present in this paper). The MLP NN used in the network is a 1 hidden layer, 2 inputs (*e.g.*, transmission power level and internode distance for the lifetime prediction model) and 1 output (lifetime for the chosen example) model. The initial values are selected randomly. We do not provide the other topology choices, however, the performance results are very similar (*i.e.*, we could not gain any significant improvement in the overall performance through introducing more layers and increasing the number of neurons). Furthermore, the particular choice for the activation functions (sigmoid in the hidden layers and linear activation at the output) is a generally adapted methodology, especially for regression problems, which is the case in our implementation. We tried using the hyper tangent function, but could not improve the performance, so we decided to use the aforementioned activation functions. We also implemented statistical significance tests to make sure the proposed model is general enough, robust, and the results are consistent. The results indicate there is no significant difference in our data set selection. In other words, the model is general enough such that any given data subset with the same statistical distribution will provide similar results.

We would like to stress the fact that computational complexity of NNs is non-negligible. In other words, NNs are computationally expensive models and are challenging to implement on constrained WSN devices. In fact, there are specially adapted libraries to be used for limited capability computing platforms in practice [40]. However, aforementioned high computational cost is generally only observed during training. Once the training is completed, testing phase (or actual prediction with a given input) is a straightforward process and generally provides the expected results instantly with low computational cost. This is also valid in our particular use case.

In this study, we opted to employ NNs, specifically, as opposed to other continuous machine learning techniques such as regression analysis and polynomial regression because NN models, generally, outperform linear or nonlinear regression models from the estimation performance perspective. At the same time, NNs tend to provide a more generalized solution to the prediction problems, especially when appropriate training and cross validation techniques are used. As a matter of fact, linear regression is a single neuron NN with logistic regression output, yet, if a polynomial or sigmoid kernel is utilized as the transfer function then a nonlinear regression model can be generated accordingly. Nevertheless, using multiple layers and/or multiple neurons provide a more flexible model, hence, more complex mappings can be implemented. This is a preliminary study for us to explore the capabilities of an NN model in WSN parameter prediction. Encouraged by the success of the NN approach we built in this study, we will be extending our framework to create more generalized NN models that can be applied to a wider set of parameters with high prediction accuracy.

We create NN models for up to 121-node WSNs. Beyond 121 nodes, the MIP model cannot create results in a reasonable amount of time. That was the reason that we utilize these relatively small sized WSN choices. However, we observe that increasing the number of nodes within the WSN do not jeopardize the prediction performance of the NN model at the output. Hence, we conjecture that it might be possible to assume that the parameters of larger networks can also be predicted similarly. Nevertheless, we need to change our optimization technique to be able to explore larger topologies which we leave as one of the future research avenues.

This study, in fact, is a feasibility study where we utilized a centralized approach. For example, we assume that the base station has the complete network topology information. As such, we could determine the performance limits of the proposed NN model. However, for practical applicability, our solution should be operating as a distributed algorithm. Hence, one of the future research avenues is transforming the NN model in this study to a distributed algorithm.

6 Conclusion

In WSNs, rapid and accurate prediction of network lifetime, internode distance, and power level are important. In fact, these three parameters are closely related, therefore, once two of them are fixed the optimal value of the third one can be predicted with high accuracy. In WSN literature, network lifetime optimization problems, commonly, are treated as MIP problems, however, MIP problems, generally, are NP complete, therefore, the solution times are prohibitively large. Furthermore, the inverse problem of designing a WSN (*i.e.*, determining network parameters) given a certain network lifetime is much more complicated and time consuming. As a remedy for this problem, we propose an NN based estimation scheme. We utilize an experimentally verified link layer model and empirically determined energy dissipation characteristics of widely used Mica2 mote platform. Due to the computational complexity arise from the MIP model, we employe the proposed NN technique as our heuristic algorithm. WSN lifetime calculations by using the MIP model which takes over 58 minutes on a Intel 8-core i7 machine to be solved optimally drops drastically to half a second with around 3% of error for lifetime and 6% for transmission power level when computed using the proposed NN approach. As for internode distance, 64% of the predictions have an error of less than 3 meters which is considered insignificant in RF communications. Predictions are statistically tested with T-tests and results obtained show that differences between calculated parameters and predicted parameters are statistically insignificant. As such, our model is very useful in instant estimation of crowded and/or complex network topologies where the estimation of network parameters via a mathematical programming framework (as in this study) can be very costly.

References

1. Ahad, N., Qadir, J., Ahsan, N.: Neural networks in wireless networks: Techniques, applications and guidelines. Journal of Network and Computer Applications **68**, 1–27 (2016)
2. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. Ad Hoc Networks **3**(3), 325 – 349 (2005)
3. Alsheikh, M.A., Lin, S., Niyato, D., Tan, H.P.: Machine learning in wireless sensor networks: Algorithms, strategies, and applications. IEEE Communications Surveys Tutorials **16**(4), 1996–2018 (2014)
4. Alwadi, M.A.: Energy efficient wireless sensor networks based on machine learning. Ph.D. thesis, University of Canberra (2015)
5. Aram, S., Mesin, L., Pasero, E.: Improving lifetime in wireless sensor networks using neural data prediction. In: Proc. World Symposium on Computer Applications & Research (WSCAR), pp. 4–6 (2014)
6. Baesens, B., Gestel, T.V., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society **54**(6), 627–635 (2003)
7. Bai, X., Liu, L., Cao, M., Panneerselvam, J., Sun, Q., Wang, H.: Collaborative actuation of wireless sensor and actuator networks for the agriculture industry. IEEE Access **5**, 13,286–13,296 (2017)
8. Barve, Y.D.: Neural network approach to the prediction of percentage data packet loss for wireless sensor networks. In: Proc. Southeastern Symposium on System Theory, pp. 103–106 (2009)
9. Bhatia, T., Kansal, S., Goel, S., Verma, A.: A genetic algorithm based distance-aware routing protocol for wireless sensor networks. Computers & Electrical Engineering **56**, 441–455 (2016)
10. Blue, J., Candela, G., Grother, P., Chellappa, R., Wilson, C.: Evaluation of pattern classifiers for fingerprint and ocr applications. Pattern Recognition **27**(4), 485–501 (1994)
11. Cao, J., Ahmadi, M., Shridhar, M.: A hierarchical neural network architecture for handwritten numeral recognition. Pattern Recognition **30**(2), 289–294 (1997)
12. Chen, L.S., Liu, C.H., Chiu, H.J.: A neural network based approach for sentiment classification in the blogosphere. Journal of Informetrics **5**(2), 313–322 (2011)
13. Chincoli, M., den Boef, P., Liotta, A.: Cognitive channel selection for wireless sensor communications. In: Proc. IEEE Int. Conf. Networking, Sensing and Control (ICNSC), pp. 795–800 (2017)
14. El-Hoiydi, A., Decotignie, J.D.: WiseMAC: An ultra low power MAC protocol for multihop wireless sensor networks. In: S.E. Nikoletseas, J.D.P. Rolim (eds.) Algorithmic Aspects of Wireless Sensor Networks, pp. 18–31 (2004)

15. Enami, N., Moghadam, R.A., Haghighat, A.: A survey on application of neural networks in energy conservation of wireless sensor networks. In: Proc. Int. Conf. Recent Trends in Wireless and Mobile Networks (WiMo), pp. 283–294 (2010)
16. GAMS Development Corporation: General Algebraic Modeling System (GAMS) Release 24.2.1. Washington, DC, USA (2013). URL http://www.gams.com/
17. Gao, T., Chen, M., Gu, H., Yin, C.: Reinforcement learning based resource allocation in cache-enabled small cell networks with mobile users. In: Proc. IEEE/CIC Int. Conf. Communications in China (ICCC), pp. 1–6 (2017)
18. Gharghan, S.K., Nordin, R., Ismail, M., Ali, J.A.: Accurate wireless sensor localization technique based on hybrid PSO-ANN algorithm for indoor and outdoor track cycling. IEEE Sensors Journal **16**(2), 529–541 (2016)
19. Guo, L., Chen, F., Dai, Z., Liu, Z.: WSN cluster head selection algorithm based on neural network. In: Proc. Int. Conf. on Machine Vision and Human-machine Interface, pp. 258–260 (2010)
20. Guo, W., Zhang, W.: A survey on intelligent routing protocols in wireless sensor networks. Journal of Network and Computer Applications **38**, 185–201 (2014)
21. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd Edition. Prentice Hall PTR (1998)
22. He, Y., Zhang, Z., Yu, F.R., Zhao, N., Yin, H., Leung, V.C.M., Zhang, Y.: Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks. IEEE Transactions on Vehicular Technology **66**(11), 10,433–10,445 (2017)
23. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications **1**(4), 660–670 (2002)
24. Jiang, C., Zhang, H., Ren, Y., Han, Z., Chen, K.C., Hanzo, L.: Machine learning paradigms for next-generation wireless networks. IEEE Wireless Communications **24**(2), 98–105 (2017)
25. Karakaya, M., Sevinç, E.: An efficient genetic algorithm for routing multiple uavs under flight range and service time window constraints **10**, 105 (2017)
26. Kaswan, A., Singh, V., Jana, P.K.: A multi-objective and PSO based energy efficient path design for mobile sink in wireless sensor networks. Pervasive and Mobile Computing **46**, 122–136 (2018)
27. Kocer, H.E., Cevik, K.K.: Artificial neural networks based vehicle license plate recognition. In: Procedia Computer Science, vol. 3, pp. 1033–1037 (2011)
28. Kulin, M., Fortuna, C., De Poorter, E., Deschrijver, D., Moerman, I.: Data-driven design of intelligent wireless networks: An overview and tutorial. Sensors **16**(6:790) (2016)

29. Kulin, M., Kazaz, T., Moerman, I., Poorter, E.D.: End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications. IEEE Access **6**, 18,484–18,501 (2018)
30. Kulin, M., de Poorter, E., Kazaz, T., Moerman, I.: Poster: Towards a cognitive MAC layer: Predicting the MAC-level performance in dynamic WSN using machine learning. In: Proc. Int. Conf. Embedded Wireless Systems and Networks, pp. 214–215 (2017)
31. Kuzborskij, I., Gijsberts, A., Caputo, B.: On the challenge of classifying 52 hand movements from surface electromyography. In: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4931–4937 (2012)
32. Laganà, M., Glaropoulos, I., Fodor, V., Petrioli, C.: Modeling and estimation of partially observed wlan activity for cognitive WSNs. In: Proc. IEEE Wireless Communications and Networking Conference (WCNC), pp. 1526–1531 (2012)
33. Li, H., Gao, H., Lv, T., Lu, Y.: Deep q-learning based dynamic resource allocation for self-powered ultra-dense networks. In: Proc. IEEE Int. Conf. Communications Workshops (ICC Workshops), pp. 1–6 (2018)
34. Li, J., Liu, D.: An energy aware distributed clustering routing protocol for energy harvesting wireless sensor networks. In: Proc. IEEE/CIC Int. Conf. Communications in China (ICCC), pp. 1–6 (2016)
35. Liu, T., Cerpa, A.E.: Data-driven link quality prediction using link features. ACM Trans. Sen. Netw. **10**(2), 37:1–37:35 (2014)
36. Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B.: A review of classification algorithms for eeg-based brain–computer interfaces. Journal of Neural Engineering **4**(2), R1 (2007)
37. Ma, Y., Peng, M., Xue, W., Ji, X.: A dynamic affinity propagation clustering algorithm for cell outage detection in self-healing networks. In: Proc. IEEE Wireless Communications and Networking Conference (WCNC), pp. 2266–2270 (2013)
38. Mao, Q., Hu, F., Hao, Q.: Deep learning for intelligent wireless networks: A comprehensive survey. IEEE Communications Surveys Tutorials pp. 1–1 (2018)
39. Martinez-Sala, A.S., Pardo, J.M.M.G., Egea-Lopez, E., Vales-Alonso, J., Juan-Llacer, L., Garcia-Haro, J.: An accurate radio channel model for wireless sensor networks simulation. Journal of Communications and Networks **7**(4), 401–407 (2005)
40. McDanel, B., Teerapittayanon, S., Kung, H.: Embedded binarized neural networks. In: Proc. Int. Conf. on Embedded Wireless Systems and Networks (EWSN), pp. 168–173. Junction Publishing (2017)
41. Mishra, A., Chakraborty, S., Li, H., Agrawal, D.P.: Error minimization and energy conservation by predicting data in wireless body sensor networks using artificial neural network

and analysis of error. In: Proc. Consumer Communications and Networking Conf. (CCNC), pp. 177–182 (2014)
42. Nie, L., Jiang, D., Yu, S., Song, H.: Network traffic prediction based on deep belief network in wireless mesh backbone networks. In: Proc. IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–5 (2017)
43. Ortiz, A., Al-Shatri, H., Li, X., Weber, T., Klein, A.: Reinforcement learning for energy harvesting point-to-point communications. In: Proc. IEEE Int. Conf. Communications (ICC), pp. 1–6 (2016)
44. Ozbayoglu, A.M., Yuksel, H.E.: Analysis of gas–liquid behavior in eccentric horizontal annuli with image processing and artificial intelligence techniques. Journal of Petroleum Science and Engineering **81**, 31–40 (2012)
45. Pal, V., Singh, Y.G., Yadav, R.: Cluster head selection optimization based on genetic algorithm to prolong lifetime of wireless sensor networks. Procedia Computer Science **57**, 1417–1423 (2015)
46. Payal, A., Rai, C.S., Reddy, B.V.R.: Artificial neural networks for developing localization framework in wireless sensor networks. In: Proc. Int. Conf. Data Mining and Intelligent Computing (ICDMIC), pp. 1–6 (2014)
47. Peng, S., Jiang, H., Wang, H., Alwageed, H., Yao, Y.D.: Modulation classification using convolutional neural network based deep learning model. In: Proc. Wireless and Optical Communication Conference (WOCC), pp. 1–5 (2017)
48. Pinto, A., Montez, C., Araújo, G., Vasques, F., Portugal, P.: An approach to implement data fusion techniques in wireless sensor networks using genetic machine learning algorithms. Information Fusion **15**, 90–101 (2014)
49. Qiu, M., Song, Y., Akagi, F.: Application of artificial neural network for the prediction of stock market returns: The case of the japanese stock market. Chaos, Solitons & Fractals **85**, 1–7 (2016)
50. Qiulan, W., Xia, G., Yong, L., Hongli, X.: Research on multi-sensor data fusion algorithm of soil carbon sink factors based on neural network. In: Proc. Int. Conf. Mechatronic Sciences, Electric Engineering and Computer (MEC), pp. 637–641 (2013)
51. Rahat, A.A., Everson, R.M., Fieldsend, J.E.: Evolutionary multi-path routing for network lifetime and robustness in wireless sensor networks. Ad Hoc Networks **52**, 130–145 (2016)
52. Rajendran, S., Meert, W., Giustiniano, D., Lenders, V., Pollin, S.: Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. IEEE Transactions on Cognitive Communications and Networking pp. 1–1 (2018)
53. Rana, S.P., Dey, M., Siddiqui, H.U., Tiberi, G., Ghavami, M., Dudley, S.: UWB localization employing supervised learning method. In: Proc. IEEE Int. Conf. Ubiquitous Wireless Broadband (ICUWB), pp. 1–5 (2017)

54. Rumelhart, D., Hinton, G., Williams, R.: Learning representations of back-propagation errors. Nature **323**, 533–536 (1986)
55. Sabitha, R., Bhuma, K., Thyagarajan, T.: Transmission power control using state estimation-based received signal strength prediction for energy efficiency in wireless sensor networks. Turk J Elec Eng & Comp Sci **25**(1), 591–604 (2017)
56. Sarikan, S., Ozbayoglu, A., Zilci, O.: Automated vehicle classification with image processing and computational intelligence. In: Procedia Computer Science, vol. 114, pp. 515–522 (2017)
57. Serin, G., Gudelek, U., Ozbayoglu, A., Unver, H.: Estimation of parameters for the free-form machining with deep neural network. In: Proc. IEEE International Conference on Big Data, pp. 2102–2111. IEEE (2017)
58. Sezer, O., Ozbayoglu, A., Dogdu, E.: An artificial neural network-based stock trading system using technical analysis and big data framework. In: ACM Southeast Conference, pp. 223–226. ACM (2017)
59. Sha, M., Dor, R., Hackmann, G., Lu, C., Kim, T.S., Park, T.: Self-adapting MAC layer for wireless sensor networks. In: Proc. IEEE Real-Time Systems Symposium, pp. 192–201 (2013)
60. Shahamiri, S.R., Salim, S.S.B.: Real-time frequency-based noise-robust automatic speech recognition using multi-nets artificial neural networks: A multi-views multi-learners approach. Neurocomputing **129**, 199–207 (2014)
61. Singh, P., Agrawal, S.: Node localization in wireless sensor networks using the M5P tree and SMOreg algorithms. In: Proc. Int. Conf. on Computational Intelligence and Communication Networks, pp. 104–104 (2013)
62. Singh, P., Agrawal, S.: TDOA based node localization in WSN using neural networks. In: Proc. Int Conf. Communication Systems and Network Technologies, pp. 400–404 (2013)
63. Stallkamp, J., Schlipsing, M., J. Salmen, C.I.: Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural Networks **32**, 323–332 (2012)
64. Subha, C.P., Malarkan, S., Vaithinathan, K.: A survey on energy efficient neural network based clustering models in wireless sensor networks. In: Proc. Int. Conf. Emerg. Trends VLSI, Embedded Systems, Nano Electr. and Telecomm. Systems (ICEVENT), pp. 1–6 (2013)
65. Sun, H., Chen, X., Shi, Q., Hong, M., Fu, X., Sidiropoulos, N.D.: Learning to optimize: Training deep neural networks for wireless resource management. In: Proc. IEEE Int. Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1–6 (2017)
66. Sun, W., Lu, W., Li, Q., Chen, L., Mu, D., Yuan, X.: WNN-LQE: Wavelet-neural-network-based link quality estimation for smart grid WSNs. IEEE Access **5**, 12,788–12,797 (2017)

67. The MathWorks Inc.: MATLAB Neural Networks Toolbox Release 2017b. Natick, Massachusetts, United States (2017). URL https://www.mathworks.com/products/neural-network.html
68. Valadarsky, A., Schapira, M., Shahaf, D., Tamar, A.: A machine learning approach to routing. CoRR **abs/1708.03074** (2017)
69. Wolsey, L.: Integer Programming. Wiley Interscience Publication (1998)
70. Yetgin, H., Cheung, K.T.K., El-Hajjar, M., Hanzo, L.H.: A survey of network lifetime maximization techniques in wireless sensor networks. IEEE Communications Surveys Tutorials **19**(2), 828–854 (2017)
71. Yildiz, H.U., Tavli, B., Yanikomeroglu, H.: Transmission power control for link-level handshaking in wireless sensor networks. IEEE Sensors Journal **16**, 561–576 (2015)
72. Yoon, S., Shahabi, C.: The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks. ACM Trans. Sen. Netw. **3**(1:3) (2007)
73. Yue, H., Xuedong, G.: Clustering on heterogeneous networks. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **4**(3), 213–233 (2014)
74. Zhang, R.: On active learning and supervised transmission of spectrum sharing based cognitive radios by exploiting hidden primary radio feedback. IEEE Transactions on Communications **58**(10), 2960–2970 (2010)
75. Zimmerling, M., Ferrari, F., Mottola, L., Voigt, T., Thiele, L.: pTUNES: Runtime parameter adaptation for low-power MAC protocols. In: Proc. ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN), pp. 173–184 (2012)