

A predictive maintenance system for wireless sensor networks: a machine learning approach

Mohammed Almazaideh, Janos Levendovszky

Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Budapest, Hungary

Article Info

Article history:

Received Jul 21, 2021

Revised Nov 24, 2021

Accepted Dec 1, 2021

Keywords:

FFNN

Machine learning

PdMs

Predictive maintenance systems

QoS of WSN

ABSTRACT

Predictive maintenance system (PdM) is a new concept that helps system operators evaluate the current status of their systems, and it also assists in predicting the future quality of these systems and scheduling maintenance action. This paper proposes a PdM model that utilizes machine learning to predict the system's operational status after M active steps based on L previous observations implemented by a feedforward neural network (FFNN). We use quantization and encoding schemes to reduce the complexity of the system. We apply the proposed model to build a PdM system for wireless sensors networks (WSNs), where our concern is to predict the state of the system as far as the quality of data transfer is concerned. The FFNN provides a forward prediction of the operational status of the network after M consecutive time steps in the future, based on the previous L readings of quality of service (QoS) requirements of WSN. We also demonstrate the relation between complexity and accuracy. We found that larger M leads to higher complexity and larger prediction error, where larger L entails higher complexity and smaller prediction error. We also investigate how quantization and encoding can reduce complexity to implement a real-time PdM system.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Mohammed Almazaideh

Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics

Budapest University of Technology and Economics

Budapest, Műegyetem rkp. 3, 1111 Hungary

Email: Almazaida@hit.bme.hu

1. INTRODUCTION

Predictive maintenance (PdM) is concerned with collecting data and estimating the operationally of the system under observation. PdM enables the users to evaluate the operating conditions and diagnose faults of the system. It also helps estimate the time of the next failure and approximate the remaining life-time of the system. PdM maximizes the system life cycle and minimizes unplanned downtime, so it also has a significant positive impact on the system's reliability under monitoring and production quality. Furthermore, PdM significantly reduces the cost of maintenance [1].

Wireless sensors networks (WSNs) and internet of things (IoT) [2] technologies are crucial tools used in the development and enhancement of PdM. They enable large-scale data acquisition from sensors distributed on machines, factories, and sites under observation. Effective PdM requires the availability of an active sensing scheme to collect the measurements to describe the working conditions of the maintained systems. The types of sensors and their numbers, distribution, and reliability play a key role in PdM's productivity and quality. The

sensing and monitoring process should be continuous, periodic, and remote to guarantee the amount and the accuracy of the data needed for precise prediction and decision [3].

Many researchers and designers of the PdM system use the WSNs and IoT as the backbone of their approaches. WSNs provide their solutions with an automatic monitoring system that does not require manual measurements in dangerous and harsh industrial environments. Moreover, wireless communications used with WSNs make it easy to deploy and configure PdM systems. Still, it may suffer from some drawbacks: limited energy resources, security, bandwidth, and limited processing capacity [4].

Besides WSNs and IoT, machine learning (ML) and deep learning (DL) [5] also are essential tools utilized in the improvement and imperfection of PdM. Neural networks are the foundation of ML/DL; they accept inputs in a two or one-dimensional form; and the output is either a categorical response in the classification model or a continuous response in the case of the regression model. Recently, many ML and DL approaches have emerged, such methods can deal with huge, multi-dimensional, and multi-variate data, and they can realize the relationships within. However, it is essential to use the appropriate approach and develop efficient prediction and classification methods to earn high performance and attain PdM's objectives [6].

This paper proposes a PdM approach consists of a prediction model and ML algorithm. The prediction model estimates the forward probability distribution of the operational status of the monitored system, the information about the monitored system is summarized in a multi-variant time series. The model estimates the probability that the system is still fully operational in the next M steps; it checks that the operability in the next M steps is guaranteed with given reliability determined by predefined parameter ϵ . The proposed model is implemented by an ML algorithm based on feedforward neural network (FFNN).

This study uses the proposed approach as a PdM for WSNs. The input of PdM is the previous L observations of the QoS parameters; the QoS parameters of the WSN include packet loss (reliability), delay, throughput, and energy consumption; they are represented as a multi-variant times series. The output is a vector that represents the status of WSN after M steps from the present time instance. We also implement quantization and special encoding schemes to reduce the complexity and memory usage of the model to make it compatible with the limited resources of WSNs.

The remainder of the paper is organized as follows: (i) In section 2, we provide a literature overview of the related work; (ii) In section 3, we present a formal presentation of the problem and the model; (iii) In section 4, we customize the model as PdM system for WSNs; (iv) In section 5, we describe the set up of the training data set; (v) In section 6, we give the numerical results of a detailed performance of the algorithm under different scenarios; and (vi) In section 7, we state some conclusions and give some commentary on the future.

2. RELATED WORK

Some researchers credit the invention of PdM to the Rio Grande Railway Company in the '40s of the 20th century [7]. The research are valuable surveys of architectures, approaches, and purposes of PdM systems; they have shown that PdM represents an essential feature of smart manufacturing systems, known as the fourth industrial revolution (industry 4.0) [1], [8], [9]. Presently, PdM is a hot research topic in the industry, covering all engineering fields ranging from civil engineering to structural engineering.

In civil engineering, the researchers proposed a PdM system in [10] to monitor railway tunnels, where the author of [11] used image processing to design a PdM system to detect and classify road distresses. PdM systems are also used in mechanical engineering, wherein [12], the researchers presented a PdM solution for metallic structure against corrosion. Also, in electrical engineering, Massaro *et al.* [13] described how to exploit various technologies to design a PdM system for energy router building equipments. Ullah *et al.* [14] used the thermal images and machine learning approach to develop a PdM system for power substation equipments.

DL and ML techniques are essential tools to ease humanitarian activities; Their applications include: natural language processing [15], self-driving cars [16], human motion detection [17], [18], health care [19], and so many other applications. There are several techniques of DL and ML utilized in designing PdM systems, most of them implemented by feedforwarded neural networks (FFNNs). Khumprom *et al.* [20] used FFNN for the prognostics of aircraft gas turbine engines and provide a data-driven model, where the complexity of the model increases with the amount of the collected data. Each piece of data is related to a different feature of the system under observation, and they reduced the complexity by cutting down on the amount of data by using an appropriate selection of the features and dimension reduction.

The PdM approach proposed in [21] is based on restricted Boltzmann machine (RBM) and support vector machine (SVM) algorithms; they used image-recognition and time series forecasting to classify the collected data as normal or abnormal. It is a fast training model because it consists of just one layer, making it unsuitable for a massive amount of data and noisy environments. Convolutional neural network (CNN) model is used in [22]; the authors modified the idea of convolution (used widely in image processing) by adding a dislocated time series (DTS). DTS discovers the relationships among the signals with different intervals in periodic mechanical signals. This technique uses shared weights to make use of neighborhoods, and the output depends on the current observations rather than the previous ones.

Tahsien *et al.* [23] presented a survey of research that implemented ML/DL techniques to improve the functionality of WSNs and IoT systems; their central aspect is network intrusion detection. Liu and Cerpa [24] used Naïve Bayes (NB) model, FFNN, and logistic regression (LR) classifier. Their approach predicts the probability of successful reception of the next packet; the inputs of the model are packet reception ratio (PRR), and physical feature of previous packets includes: signal to noise ratio (SNR), received signal strength indicator (RSSI) and link quality indicator (LQI).

Kulin *et al.* [25] proposed an ML model to predict the performance of WSNs in terms of reliability. Their model is based on regression trees, linear regression, and neural networks. The input of the model is a vector of the number of detected nodes (d), inter-packet-interval (IPI), number of received packets (RP), and number of erroneous packets/frames (errP). The output is the estimation of packet loss rate (PLR). Akbas *et al.* [26] utilized the neural network model (NN) to predict the life-time of sensors based on transmission power level and internode distance. An in-depth learning approach was proposed in [27] to estimate the energy consumption (EC) and packet delivery ratio (PDR) depending on ten input features (distance, actual transmissions, and queue size).

This paper presents a mathematical analysis of a prediction model for PdMs, and we use it with ML algorithm to build a PdM system for WSNs; most of the studies above use WSNs as the backbone and the key component of PdM [4], [28], to the best of our knowledge, there are very few studies interested in finding PdM for WSN, most of them dominating intrusion detection of IoT systems. In this study, WSN is not only a tool but also the PdM system's subject; the proposed approach takes the QoS and limited resources of WSN into account.

3. THE SYSTEM MODEL

This paper proposes a PdM approach consists of: (i) Prediction model estimates the forward probability distribution of the operational status of the monitored system, the information about the monitored system entered into the model in the form of a multi-variant time series. The model estimates the operational status of the system during the next M steps; it checks that the operability in the next M steps is guaranteed with given reliability determined by predefined parameter ϵ . (ii) ML algorithm to implement the prediction model. The proposed model is implemented by an ML algorithm based on FFNN.

3.1. Predicting the forward probability distribution

Let us assume that the information about the monitored system is summarized in times series $x(k)$, this time series can result from direct measurements or pre-processed data obtained by data fusion. Evaluation on the system state can be summarized as follows:

- If $x(k) > A$ then the system is malfunctioning and urgent maintenance action is required;
- If $x(k) \leq A$ then the system operates normally.

Based on the observations $x(k-1), x(k-2), \dots, x(k-L+1)$ the underlying challenge is to estimate the probability that the system is still fully operational in the next M steps:

$$P(x(k+M) \leq A, x(k+M-1) \leq A, \dots, x(k) \leq A | x(k-1) = i, \dots, x(k-L+1) = j) \quad (1)$$

or more precisely, to check whether operability in the next M steps is guaranteed with given reliability determined by parameter ϵ i.e. in (2).

$$M : P(x(k+M) \leq A, x(k+M-1) \leq A, \dots, x(k) \leq A | x(k-1) = i, \dots, x(k-L+1) = j) \geq 1 - \epsilon \quad (2)$$

By introducing the notations:

$$\begin{aligned} x^+(k) &:= (x(k+M), x(k+M-1), \dots, x(k)) \\ x^-(k) &:= (x(k-1), \dots, x(k-L+1)) \end{aligned} \quad (3)$$

one can write this probability in a more compact form, where set B is defined as: $B := x_i < A, \dots, x_M < A \forall 1 \leq i \leq M$.

$$M : P(x^+(k) \in B | x^-(k) = (i, \dots, j)) \geq 1 - \varepsilon \quad (4)$$

Introducing the following two vectors:

$$\begin{aligned} s^{(1)} &= (s_1^{(1)}, s_2^{(1)}) = (1, 0) \rightarrow \text{ha } x^+ \in B \\ s^{(2)} &= (s_1^{(2)}, s_2^{(2)}) = (0, 1) \rightarrow \text{ha } x^+ \notin B \end{aligned} \quad (5)$$

one can form a training set as (6).

$$\tau^{(K)} = \{(x^-(k), s(k)), k = 1, \dots, K\}, s(k) \in \{s^{(1)}, s^{(2)}\} \quad (6)$$

3.2. FFNN algorithm

We use FFNN to implement the ML algorithm. FFNNs have the most straight forward architecture. They have inputs, outputs, and numbers of hidden layers between them; as the number of hidden layers increases, the data moves in one direction from the input layer to the output layer. This study uses backpropagation (BP) as a training algorithm. It is one of the most fundamental and common training algorithms. The estimated output is calculated based on the activation function. Then, it calculates the estimation error based on the loss function. It goes backward to update the weights based on the gradient of the loss function.

The efficiency of FFNN depends on several factors such as the selection of appropriate activation function, selection of proper training algorithm, the suitable structure of hidden layers, size of the training set, and the accurate description of the problem [29]. Unfortunately, there are no standard rules for selecting, comparing, and testing the solutions; the user's satisfaction in accuracy and complexity is the primary benchmark.

The training set in (6) is used to train the corresponding FFNN, where the input-output mapping of the FFNN is $y = \text{Net}(x, w)$, where vector w refers to the weights of the network subject to learning. The weights can then be optimized by the backpropagation (BP) algorithms as:

$$w_{opt} : \min \frac{1}{K} \sum_{k=1}^K \|s(k) - \text{Net}(x^-(k), w)\|^2 \quad (7)$$

yielding:

$$\frac{1}{K} \sum_{k=1}^K \|s(k) - \text{Net}(x^-(k), w)\|^2 \rightarrow E \|s - \text{Net}(x^-, w)\|^2 \quad (8)$$

and then:

$$w_{opt} : \min_w E \|s - \text{Net}(x^-, w)\|^2 \rightarrow \text{Net}(x^-, w) = E(s|x^-) \quad (9)$$

subject to (5):

$$E(s|x^-) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P(x^+ \in B|x^-) \\ P(x^+ \in B^c|x^-) \end{pmatrix} \quad (10)$$

where:

$$\begin{aligned} E_1(s|x^-) &= P(x^+ \in B|x^-) \\ E_2(s|x^-) &= P(x^+ \in B^c|x^-) \end{aligned} \quad (11)$$

as a result, after learning, at the output of the FFNN, one can observe the estimated conditioned probabilities once the past observations are given in the input. If $P(x^+ \in B | x^-) \geq 1 - \varepsilon$ then there are at least M steps to failure. Figure 1 shows the structure of FFNN consists of three hidden layers; the input is 8 previous observations.

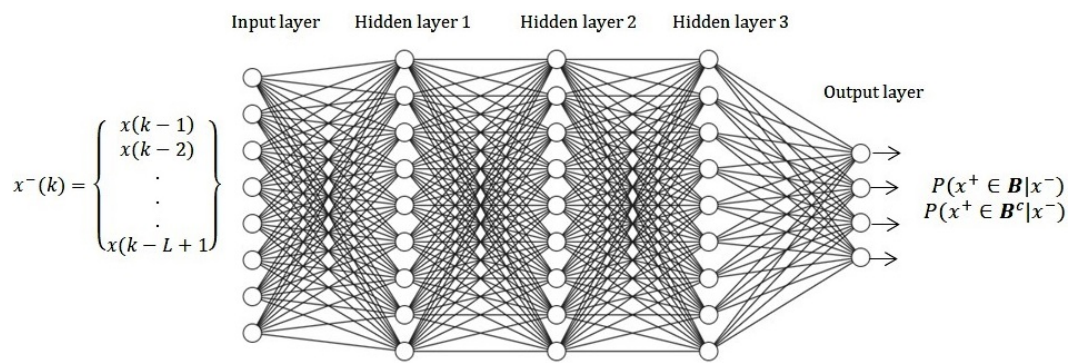


Figure 1. FFNN of three hidden layers and L=8 previous observations window

4. CUSTOMIZED PDM FOR WSNs

In this section, the proposed model is customized as a PDM for WSNs. WSN consists of some small nodes and one or more BS to form a data collection system; the nodes communicate with each other and with the BS via a wireless radio transceiver attached to them. The nodes are rigged up with application-specific sensors to measure or track a specific physical phenomenon; they have a limited-capacity central processing unit. These nodes often operate on batteries as a limited-energy source; besides that, they usually work in harsh and complex environments [30]. Designers and operators of WSN should consider their limited resources (memory and processing capabilities), limited communication bandwidth, limited energy, and other restrictions.

In the face of any limitations, any system’s performance should satisfy the minimum level of services and requirements, known as quality-of-service (QoS); in the case of WSN, QoS’s include reliability, energy efficiency, security, accuracy, delay, and-so-forth. Maintenance procedures may include selecting new heads of clusters and leaders of chains, rearrangement of clusters and chains, new sensors deployments, controlling ON/OFF schemes, and many other procedures that enhance the performance of WSNs. The limited resources of WSNs require a low complexity PDM; to reduce the complexity, we use quantization and encoding schemes.

4.1. Quantized FFNN

WSNs are limited resource systems in terms of energy, memory, and processing capabilities. To make our model compatible with such circumstances, we implement a quantization algorithm to speed up the training process and reduce the complexity of the model. Quantization enhances training speed and complexity, but it weakens the accuracy, so the user has to trade-off complexity with accuracy. Usually, variables and weights are represented as floating-point numbers; the quantization function converts them to integers, fixed-point, or integer numbers; such representations are more efficient regarding memory usage and computation speed [31], [32]. Uniform or deterministic quantization function calculates the quantization level (q) of the real values r as follows [32]:

$$q(r) = \text{sign}(r) \cdot \Delta \cdot \left\lfloor \frac{|r|}{\Delta} + \frac{1}{2} \right\rfloor \tag{12}$$

where Δ is the resolution or the quantization step.

Such functions are known as equidistant quantization. The quantization range is divided equally between quantization levels, so such functions are used in case of uniform distributions of the samples; when the distribution is not uniform, non-equidistant quantization is used; the authors of [33] used Lloyd-Max algorithm to determine the best quantization in such cases, it takes the PDF of samples distribution on account to minimize the mean square quantization error σ . Finding the optimal quantized level q_i of sample r is an iterative process where:

$$q_i(r) = \frac{\int_{c_i}^{c_{i+1}} r f(r) dr}{\int_{c_i}^{c_{i+1}} f(r) dr} \tag{13}$$

in (9), c_i and c_{i+1} are the regions of the proposed quantization level q_i , and $f(r)$ is the PDF of the samples, the goal is the minimization of (σ), which is:

$$\sigma_q^2 = \sum_{i=1}^Q \int_{c_i}^{c_{i+1}} (r - q_i)^2 f(r) dr \quad (14)$$

where Q is the numbers of the quantization levels.

4.2. Sparsity of FFNN

Memory is a crucial concern when dealing with FFNN for WSNs; many techniques have been used to improve the memory efficiency of ML/DL algorithms; some of them concern memory requirements of inference, others concern the memory requirements of training. Sparse FFNN is a common and efficient technique used widely to enhance DL/ML algorithms [34]. In sparse FFNN, the input features are represented as a sparse vector; most sparse vector elements are zeros, which need fewer computations and less memory space. Besides memory efficiency, sparsity improves the complexity and the computations of the FFNN. Unfortunately, at the same time, it degrades the accuracy of FFNN; the designer has to trade-off between the sparsity level and accuracy [35]. In this study, we use a straightforward encoding scheme used in [33]. It is compatible and complementary with the quantization algorithm, each quantization level is encoded into an orthonormal vector set:

$$q_l \rightarrow sq_l : sq_l(i) = \begin{cases} 1 & \text{if } i = l \\ 0 & \text{otherwise} \end{cases}, i = \{1, 2, \dots, Q\}$$

by the encoding (3) becomes:

$$x^+(k) := (sq_{(k+M)}, sq_{(k+M-1)}, \dots, sq_k), x^-(k) := (sq_{(k-1)}, \dots, sq_{(k-L+1)}) \quad (15)$$

5. SETUP OF THE DATASET

The dataset used for training, validation, and testing is imported from [36]. The researchers collected the data experimentally as described in their paper [37]. They used IEEE 802.15.4 link implemented on TinyOS to connect two TelosB motes, each mote uses a TI CC2420 radio transceiver with 250 kbps. The researchers trace the packet delivery performance under several pre-configured stack parameters; these parameters are related to physical, MAC, and application layers. We have generated an observations table consisting of 10000 entries. Each entry summarizes the average measured parameters of 300 packets; we have fixed the power transmission level at -19 dBm and change the other pre-configured parameters for the possible combination shown in Table 1. Besides the pre-configured parameters, the observations table has several packet delivery performance measured parameters corresponding to each combination of pre-configured parameters, as shown in Table 2. A short sample of the observations table is shown in Table 3.

Table 1. Pre-configured parameters

Parameters	Acronym	Values	Comments
Inter-Arrival Time	IAT (ms)	10, 15, 20, 25, 30, 35, 40, 50	Pre-configured
Packet Payload	PL (bytes)	20, 35, 50, 65, 80, 95, 110	Pre-configured
Maximum Queue Size	QS	1, 30, 60	Pre-configured
Maximum Transmission attempt	NMT	1, 3, 5	Pre-configured
Retry delay	DR	30, 60	Pre-configured
Power of transmission	Ptx	19	Pre-configured
Distance	D	10,20,35	Pre-configured

Table 2. Measured parameters

Parameters	Acronym	Values	Comments
Actual Queue Size	AQS	actual values (0–60)	measured
Buffer Overflow	OF	actual values (0–1)	measured
Actual Transmission attempt	N_A	actual values (0–5)	measured
Actualacknowledged transmission	ACK		measured
Received Signal Strength Indicator	RSSI		measured
Noise Floor	NF		measured
Link Quality Indicator	LQI		measured
Packet arrival time	T_{arr}		measured

Table 3. Sample of observations table

T_{arr}	125304	130758	137716	146187	156155
IAT	10	15	10	15	50
PL	20	35	65	95	110
QS	1	1	30	1	60
NMT	1	1	5	1	5
DR	30	30	30	60	60
P_{tx}	19	19	19	19	19
D	10	10	10	20	35
OF	0	0	0	0	0
Q	0.41	0.23	25.7	0.01	0.08
ACK	0.59	0.77	0.723	0.99	1
N_A	0.593	0.77	0.723	0.993	1.02
$RSSI$	-7.5167	-9.8567	-9.29	-16.31	-22.943
NF	-54.0767	-70.57	-61.0533	-88.9367	-93.71
LQI	63.08	82.3467	77.2833	106.13	106

We have used the pre-configured and measured parameters to calculate the QoS requirements of the WSN. Energy efficiency, throughput, delay, and packet loss as in [27], [37].

- Packet error rate (PER): measures the reliability of the system; it depends on the queuing characteristics (Buffering) of the nodes and the quality of the link parameters ($RSSI$, NF , and LQI)

$$PER = \frac{N_A - ACK}{N_A} \tag{16}$$

- Energy efficiency (En): determines the energy needed to transmits one beneficial bet; it depends on PER , power transmission level, the payload of the packet, length of the header, and transmission rate:

$$En = \frac{P_{tx} * (PL + PH) * Tt}{PL(1 - PER)} \tag{17}$$

PH is the length of the header/trailer, which is (11-31 bytes) in IEEE 802.15.4 [38], Tt is the transmission time which is $0.004ms$ in the case of $250kb/s$.

- Throughput (Tp) is the number of beneficial bets received per unit of time; it depends on PL , PER , and transmission service time (Ts), as:

$$Tp = \frac{PL(1 - PER)}{Ts} \tag{18}$$

where :

$$Ts = C + Tt + (N_A * D_R) \tag{19}$$

and C is a constant depends on the protocol and the specification of the radio system; it is $\approx 13.5 ms$ in the circumstances of the experiment [38].

- Delay is the time elapsed from packet generation to successful packet reception; LQI and queuing characteristics of the nodes are crucial issues when investigating delay. Researchers mostly use queuing system model to state the delay of WSNs; we use system utilization ρ as a metric to quantify the delay, where $\rho = Ts/IAT$ and as $\rho \rightarrow 1$ delay increases.

The four calculated QoS requirements (PER , En , Tp and ρ) are arranged into a $10000*4$ input feature table; each entry corresponds to an entry of the observations table. The packet arrival time (T_{arr}) is reformatted as a time series and added as a fifth column to the input features table. QoS metrics are contradictory; improving reliability decreases energy efficiency, and improving energy efficiency reduces throughput, and so on; the user should trade-off among these metrics. To define the operational status of the WSN, we define a range of each metric as follows:

$$\begin{aligned} \alpha^+ &\leq PER < \alpha^- & \beta^+ &\leq En < \beta^- \\ \gamma^+ &\leq Tp < \gamma^- & \delta^+ &\leq \rho < \delta^- \end{aligned}$$

If the four metrics are within the specified range, then the operational state of WSN is “OK” corresponding to $s^{(1)} = (1, 0)$ as defined in (5), which means that no maintenance is needed; otherwise, the operational status is “NOK” corresponding to $s^{(2)} = (0, 1)$ as defined in (5), which means that maintenance is needed. The operational status for each entry of the input features table represents an entry of the output table of the FFNN, concatenation of the input features table, and the output table forms the dataset of training, testing, and validation of the FFNN. Table 4 shows a short sample of the training set.

Table 4. Sample of the training dataset

T_{arr}	PER	En	Th	Ru	OP	
					OK	NOK
44488657	0.005618	0.084072	19.98877	3.1304	1	0
44544439	0.003333	0.08388	19.99833	1.0876	1	0
44559087	0	0.0836	20	0.87008	0	1
44597021	0	0.080343	35	1.74016	1	0
44607076	0.006667	0.080882	34.99222	1.450133	1	0

In the next stage, the entries of the training dataset are quantized by the Lloyd-Max algorithm by 8 quantization levels. Each quantized entry is encoded into an 8-bits binary vector, as described in section 4.3. The numerical numbers representing the QoS parameters at instant (t) are converted to a $1 * 4 * 8$ sparse vector. Each vector has four 1's indicate the quantization level of each QoS requirement. Table 5 shows a sample of the data set after quantization and encoding.

Table 5. Sample of the dataset after quantization and encoding

T_{arr}	PER	En	Th	Ru	OP
44488657	10000000	00000010	00100000	00010000	10
44544439	01000000	00001000	01000000	00010000	10
44559087	10000000	00100000	00010000	01000000	01
44597021	10000000	00000010	00000100	00000010	01
44607076	00100000	00000010	00000010	00000001	10

6. IMPLEMENTATION AND RESULTS

We implemented the proposed model using the deep learning toolbox of MATLAB2020b; we used the dataset explained in the previous section. In the first experiment, we investigate the effect of quantization and encoding on the accuracy and complexity of the PdM system. To get more use of the sparsity of the input vector; the FFNN deals with each binary input vector (as the sample is shown in Table 4 as a black and white pattern, where the ones appear as white points in a black line, Figure 2 shows a sample of these patterns.

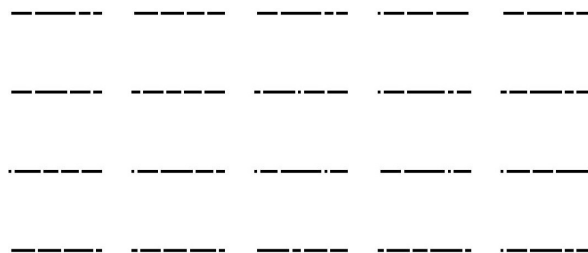


Figure 2. Samples of the input vector as black and white patterns

In this experiment, we use the accuracy as a performance metric, $Acc = R/T$ Where R is the number of correct predictions, and T is the number of the data set. Figure 3 shows the complexity of the algorithm under different numbers of hidden layers; we measure the complexity by the execution time of the training process. The figure demonstrates that the algorithm uses quantized and encoded data takes less time than the one raw data, regardless of the number of header layers. The quantized and encoded data ensures better complexity because of the sparsity enlightened in section 4.3. Both algorithms show an ascending tone of training time as the number of hidden increases. The irregularity noticed in both curves is justified by the randomness of initial values of the training process's weight and biases.

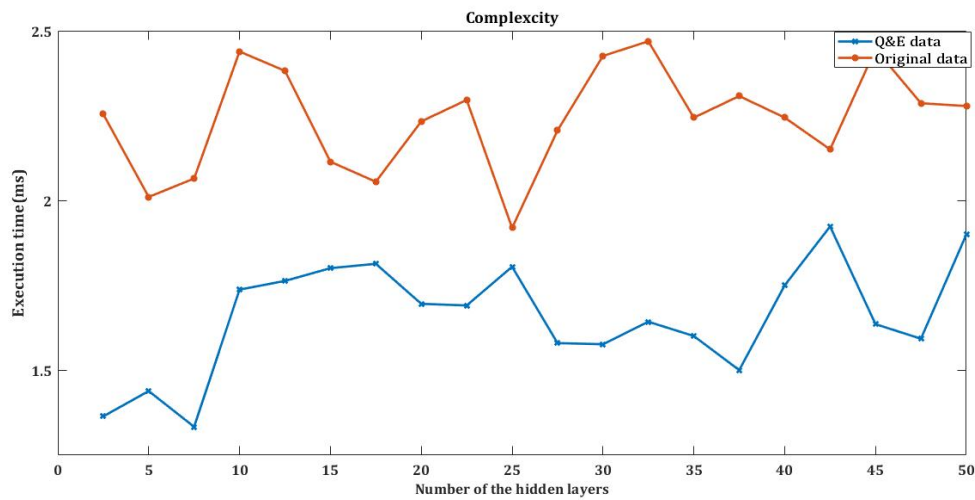


Figure 3. Complexity of original data and quantized and encoded data

In Figure 4, one notices that the raw (original) data show better accuracy than the quantized and encoded data; this happens because besides the prediction error, there is also quantization error explained in section 4.2. With quantized and encoded data, the input data appear as a lookup table, so one notices the low variance of accuracy with quantized and encoded data regardless of the number of the hidden layers. The algorithm uses the raw data exhibits better accuracy as the number of hidden layers increases

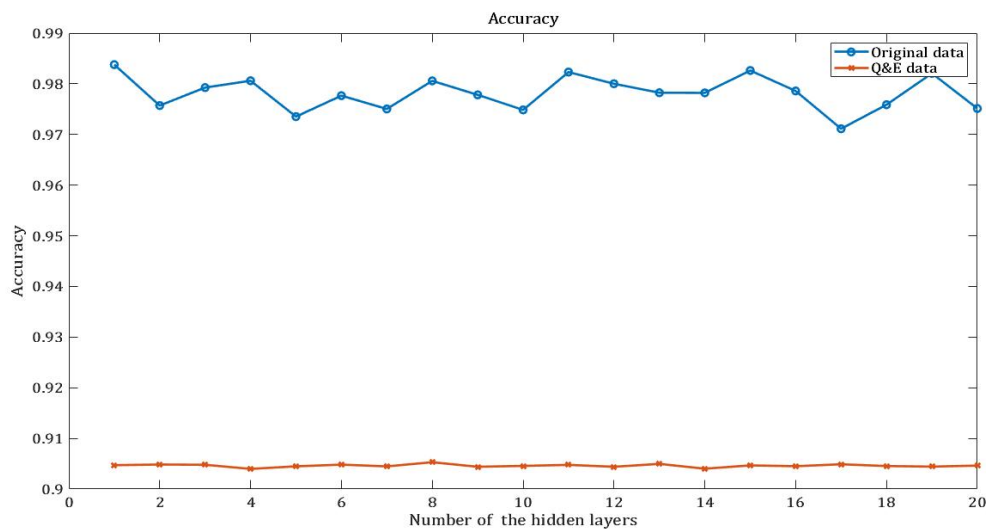


Figure 4. Accuracy of original data and quantized and encoded data

In the third experiment, we investigate the relationship between the performance and the number of future time steps M ; two metrics are used to clarify the performance; mean square error (MSE) and the execution time presenter of the complexity. The output of the FFNN is a binary vector (ops) consists of M elements, the vector (ops) states the operational status of the WSN, $ops(m) = \{m_1, m_2, \dots, m_M\}$,

$$m_i = \begin{cases} 1 & \text{The system will be OK untill step } i. \\ 0 & \text{The system will be faulty after } i \text{ steps.} \end{cases}$$

for example, if $M = 8$, then ops can be $ops = \{1, 1, 1, 1, 1, 0, 0, 0\}$, this means that the system will be faulty after five operational steps, and maintenance should take place.

Figure 5 clarifies the performance of the model under different values of $M = (1 - 10)$, where the number of hidden layers is set to ten layers, and the number of previous observations is set to 3. The left y-axis characterizes the MSE , where the right y-axis characterizes the execution time. The figure shows that as M increases, both the execution time and the MSE increase.

Figure 6 demonstrates the effect of the number of previous observations k on MSE and execution time. The number of the hidden layer is set to ten, and M is set to 5. The left y-axis represents the MSE , and the right y-axis represents the execution time; a large k means less MSE but a longer execution time.

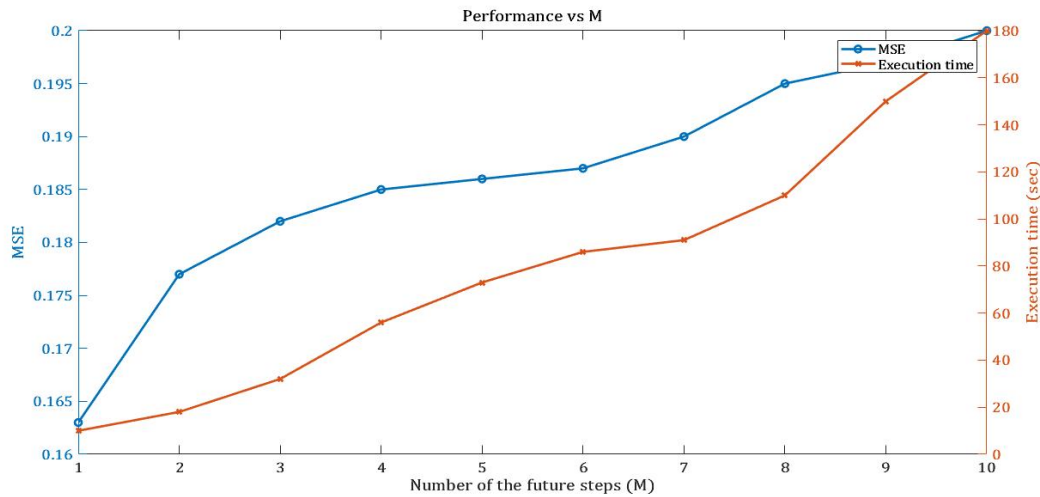


Figure 5. The relation among MSE , execution time, and M

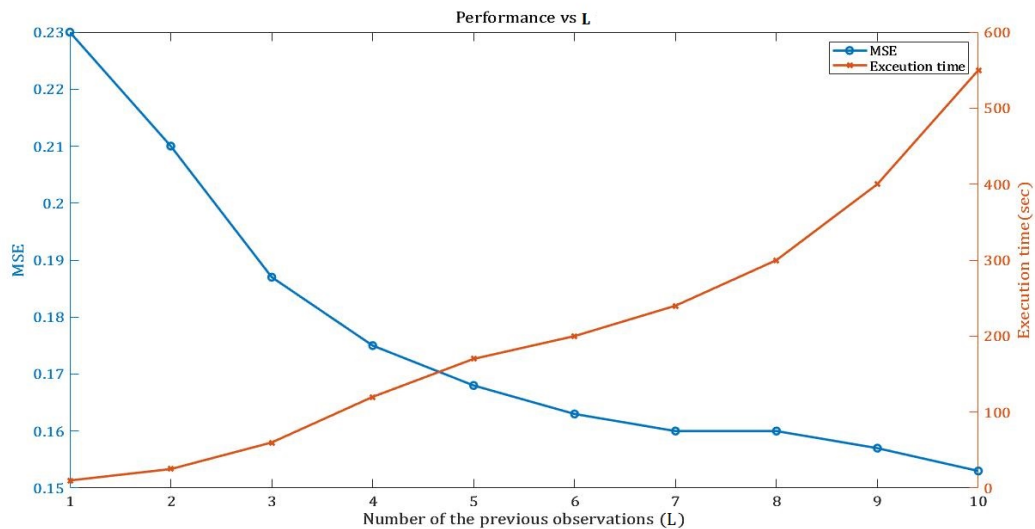


Figure 6. The relation among MSE , execution time, and L

7. CONCLUSION

In this paper, we used the FFNN machine learning model to build a PdM system for WSN. It predicts the operational status (“OK” or faulty) after M time steps based on L previous readings of QoS requirements of the WSN. We used real estate data set of one-hop WSN. We also used quantization and encoding schemes to make the system incoherent with the limited resources of the WSN. We revealed that the complexity of systems is improved by quantization, encoding, small M and small L . The accuracy is improved by using the raw (original data), small M , and large k . We will extend our approach to include multi-hop WSN and implement it by other machine and deep learning models.





REFERENCES

- [1] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A Survey of Predictive Maintenance: Systems, Purposes and Approaches," *Signal Processing*, Dec. 2019. [Online]. Available: <https://arxiv.org/abs/1912.07383>
- [2] S. T. March and G. D. Scudder, "Predictive maintenance: strategic use of IT in manufacturing organizations," *Information Systems Frontiers*, vol. 21, no. 2, pp. 327–341, 2019, doi: 10.1007/s10796-017-9749-z.
- [3] R. Isermann, *Fault-diagnosis systems: An introduction from fault detection to fault tolerance*, 2006th ed. Berlin, Germany: Springer, 2006, doi: 10.1007/3-540-30368-5.
- [4] D. Catenazzo, B. O'Flynn, and M. Walsh, "On the use of Wireless Sensor Networks in Preventative Maintenance for Industry 4.0," *2018 12th International Conference on Sensing Technology (ICST)*, 2018, pp. 256–262, doi: 10.1109/ICSensT.2018.8603669.
- [5] O. Serradilla, E. Zugasti, and U. Zurutuza, "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospect," *arXiv*, 2020.
- [6] T. Carvalho, F. Soares, R. Vita, R. Francisco, J. Basto, and S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019, doi: 10.1016/j.cie.2019.106024.
- [7] A. Prajapati, J. Bechtel, and S. Ganesan, "Condition based maintenance: a survey," *Journal of Quality in Maintenance Engineering*, vol. 18, no. 4, pp. 384–400, 2012, doi: 10.1108/13552511211281552.
- [8] H. Kang *et al.*, "Smart manufacturing: Past research, present findings, and future directions," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 1, pp. 111–128, 2016, doi: 10.1007/s40684-016-0015-5.
- [9] S. Selcuk, "Predictive maintenance, its implementation and latest trends," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2016, vol. 231, no. 9, pp. 1670–1679, doi: 10.1177/0954405415601640.
- [10] B. Farahani, F. Barros, P. Sousa, P. Tavares, and P. Moreira, "A railway tunnel structural monitoring methodology proposal for predictive maintenance," *Structural Control and Health Monitoring*, vol. 27, no. 8, 2020, doi: 10.1002/stc.2587.
- [11] T. Siriborvornratanakul, "An Automatic Road Distress Visual Inspection System Using an Onboard In-Car Camera," *Advances in Multimedia*, vol. 2018, pp. 1–10, 2018, doi: 10.1155/2018/2561953.
- [12] B. de Saporta, F. Dufour, H. Zhang, and C. Elegbede, "Optimal stopping for the predictive maintenance of a structure subject to corrosion," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 226, no. 2, pp. 169–181, 2011, doi: 10.1177/1748006x11413681.
- [13] A. Massaro, A. Galiano, G. Meuli, and S. F. Massari, "Overview and Application of Enabling Technologies Oriented on Energy Routing Monitoring, on Network Installation and on Predictive Maintenance," *International Journal of Artificial Intelligence and Applications*, vol. 9, no. 2, pp. 1–20, 2018, doi: 10.5121/ijai.2018.9201.
- [14] I. Ullah *et al.*, "Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach," *Energies*, vol. 10, no. 12, p. 1987, Dec. 2017, doi: 10.3390/en10121987.
- [15] D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.
- [16] J. Ni, Y. Chen, Y. Chen, J. Zhu, D. Ali, and W. Cao, "A Survey on Theories and Applications for Self-Driving Cars Based on Deep Learning Methods," *Applied Sciences*, vol. 10, no. 8, p. 2749, Apr. 2020, doi: 10.3390/app10082749.
- [17] M. Rungruanganukul and T. Siriborvornratanakul, "Deep Learning Based Gesture Classification for Hand Physical Therapy Interactive Program," *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Posture, Motion and Health*, vol. 12198, pp. 349–358, 2020, doi: 10.1007/978-3-030-49904-4_26.
- [18] C. Kerdivulvech, "Hybrid model of human hand motion for cybernetics application," *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 2367–2372, doi: 10.1109/SMC.2014.6974280.
- [19] A. Esteva *et al.*, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, Jan. 2019, doi: 10.1038/s41591-018-0316-z.
- [20] P. Khumprom, D. Grewell, and N. Yodo, "Deep Neural Network Feature Selection Approaches for Data-Driven Prognostic Model of Aircraft Engines," *Aerospace*, vol. 7, no. 9, p. 132, Sep. 2020, doi: 10.3390/aerospace7090132.
- [21] S. Hwang, J. Jeong, and Y. Kang, "SVM-RBM based Predictive Maintenance Scheme for IoT-enabled Smart Factory," *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, 2018, pp. 162–167, doi: 10.1109/ICDIM.2018.8847132.
- [22] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated Time Series Convolutional Neural Architecture: An Intelligent Fault Diagnosis Approach for Electric Machine," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1310–1320, Jun. 2017, doi: 10.1109/TII.2016.2645238.
- [23] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," *Journal of Network and Computer Applications*, vol. 161, p. 102630, Jul. 2020, doi: 10.1016/j.jnca.2020.102630.
- [24] T. Liu and A. E. Cerpa, "Data-driven link quality prediction using link features," *ACM Transactions on Sensor Networks*, vol. 10, no. 2, pp. 1–35, Jan. 2014, doi: 10.1145/2530535.
- [25] M. Kulin, E. de Poorter, T. Kazaz, and I. Moerman, "Towards a cognitive MAC layer: Predicting the MAC-level performance in dynamic WSN using Machine learning," in *Proc. 14th Int. Conf. Embedded Wireless Syst. Netw. (EWSN), Uppsala, Sweden*, Feb. 2017, pp. 214–215, doi: 10.5555/3108009.3108046.
- [26] 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, Aug. 2018, doi: 10.1007/s11276-018-1808-y.
- [27] M. Ateeq, F. Ishmanov, M. Afzal, and M. Naem, "Multi-Parametric Analysis of Reliability and Energy Consumption in IoT: A Deep Learning Approach," *Sensors*, vol. 19, no. 2, p. 309, Jan. 2019, doi: 10.3390/s19020309.
- [28] F. Civerchia, S. Bocchino, C. Salvadori, E. Rossi, L. Maggiani, and M. Petracca, "Industrial Internet of Things monitoring solution for advanced predictive maintenance applications," *Journal of Industrial Information Integration*, vol. 7, pp. 4–12, Sep. 2017, doi: 10.1016/j.jii.2017.02.003.
- [29] A. Laudani, G. M. Lozito, F. R. Fulginei, and A. Salvini, "On Training Efficiency and Computational Costs of a Feed Forward Neural Network: A Review," *Computational Intelligence and Neuroscience*, vol. 2015, pp. 1–13, 2015, doi: 10.1155/2015/818243.





- [30] S.-H. Yang, *Wireless Sensor Networks*. Ed. 1st, London, UK: Springer London Ltd, 2016, doi: 10.1007/978-1-4471-5505-8.
- [31] B. Jacob *et al.*, “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713, doi: 10.1109/CVPR.2018.00286.
- [32] H. Li, S. De, Z. Xu, C. Studer, H. Samet, and T. Goldstein, “Training quantized nets: a deeper understanding,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5813–5823.
- [33] A. Ceffer, N. Fogarasi, and J. Levendovszky, “Trading by estimating the quantized forward distribution,” *Applied Economics*, vol. 50, no. 59, pp. 6397–6405, Jun. 2018, doi: 10.1080/00036846.2018.1486021.
- [34] C. Louizos, M. Welling, and D. P. Kingma, “Learning sparse neural networks through L0 regularization,” *arXiv*, 2018. [Online]. Available: <http://arxiv.org/abs/1712.01312>
- [35] S. N. Sohoni, R. C. Aberger, M. Leszczynski, J. Zhang, and C. Ré, “Low-memory neural network training: A technical report,” *arXiv*, 2019. [Online]. Available: <https://arxiv.org/abs/1904.10631>
- [36] S. Fu and Y. Zhang, “CRAWDAD dataset due/packet-delivery (v. 2015-03-30),” *Community Resource for Archiving Wireless Data at Dartmouth (CRAWDAD)*, 2015. [Online]. Available: <https://crawdad.org/due/packet-delivery/20150401/>
- [37] S. Fu, Y. Zhang, Y. Jiang, C. Hu, C.-Y. Shih, and P. J. Marron, “Experimental Study for Multi-layer Parameter Configuration of WSN Links,” *2015 IEEE 35th International Conference on Distributed Computing Systems*, Jun. 2015, pp. 369-378, doi: 10.1109/ICDCS.2015.45.
- [38] I. Tomic and J. A. McCann, “A Survey of Potential Security Issues in Existing Wireless Sensor Network Protocols,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1910–1923, Dec. 2017, doi: 10.1109/jiot.2017.2749883.

BIOGRAPHIES OF AUTHORS



Mohammed Almazaideh     received B.Sc. degree in computers and digital systems engineering from Al-Balaqa Applied University, Jordan. He acquired his Master’s degree from Jordan University of science and technology, Jordan in 2011. At present, he is a Ph.D. student in factuality of electrical engineering and informatics, Department of Computer Engineering, Budapest University of Technology and Economics, Hungary. His research focus includes wireless sensors networks, IoT, and artificial intelligence. He can be contacted at email: almazaida@hit.bme.hu.



Janos Levendovszky     obtained his PhD at the Budapest University of Technology and Economics, and his DSc from the Hungarian Academy of Sciences. He is presently a full time professor at the Budapest University of Technology and Economics and also vice rector of Science and Innovation. His research area includes adaptive signal processing, networking, artificial intelligence and algebraic coding theory. He can be contacted at email: levendov@hit.bme.hu.