

RESEARCH ARTICLE

Data Prediction Based Encoder-Decoder Learning in Wireless Sensor Networks

AROUNA NDAM NJOYA¹, ALLASSAN A NKEN TCHANGMENA², ADO ADAMOU ABBA ARI^{3,4}, ABDELHAK GUEROUI⁴, CHRISTOPHER THRON⁵, BERTHINE NYUNGA MPINDA², OUSMANE THIARE⁶, AND EMMANUEL TONYE⁷

¹Department of Computer Engineering, University Institute of Technology, University of Ngaoundéré, Ngaoundéré 455, Cameroon

²African Institute for Mathematical Sciences (AIMS), Limbe 608, Cameroon

³Department of Computer Science, University of Maroua, Maroua 814, Cameroon

⁴DAVID Laboratory, Université Paris-Saclay, University of Versailles Saint-Quentin-en-Yvelines, 78035 Versailles, France

⁵Department of Mathematics, Texas A&M University–Central Texas, Killeen, TX 76549, USA

⁶Université Gaston Berger de Saint-Louis, Saint-Louis 32002, Sénégal

⁷National Advanced School of Engineering, University of Yaoundé I, Yaoundé 6880, Cameroon

Corresponding author: Arouna Ndam Njoya (ndanjoa@gmail.com)

ABSTRACT Wireless sensor networks are typically characterized by large network size and sensor nodes with low energy capacity and a limited bandwidth for data transmission. Over-activity of the sensor nodes will therefore cause many issues to the network, such as an increase in the network depletion rate and poor data transmission. Data prediction methods that exploit the inter-relationship between sensor nodes can be used to reduce data traffic across wireless networks. Several related work in data prediction do not consider the time-series distribution of the sensing data. However, exploiting sequential features of the historical observations can improve prediction accuracy, and increase the number of sensing data predicted per sequence. This work propose a sequence to sequence data prediction model, using a one dimensional layer convolutional neural network to extract spatial features from the pre-processed sensing data, and an encoder-decoder model to predict the next two outputs in the sequence by exploiting the temporal distribution of the data. The aforementioned approach has the capacity of generating more accurate information, which can reduce network traffic and energy expenditure in WSNs. Furthermore, the experimental results reveals that, based on a suitable choice of nodes, our proposed model perform accurate predictions, with reduced root mean squared error as compared to related work. We also propose an approach to regulate and control the data traffic toward the base station.

INDEX TERMS Wireless sensor network, data prediction, lifetime extension, deep learning.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are commonly used where remote monitoring and control are needed. As such, we can find their application in environmental data collection, smart transportation systems, health monitoring systems and agriculture, among others on [1], [2], [3], [4]. This is mostly because of their very low cost, abundant availability and large-scale deployment capacity. Despite this high usability property, WSNs' applications come up with many issues like data transmission delay, poor and abnormal transmission quality and very high energy consumption. Energy

consumption is indeed one of the commonly encountered problems in WSNs as it has a strong relationship with the lifetime of the nodes constituting the network [5], [6]. This is to say that, the higher the working activity of the network nodes, the faster it deteriorates and as a result the less efficient data transmission becomes. Related works to overcome this problem rely on techniques such as data compression, data aggregation and data clustering algorithms, to limit the quantity of information transmitted [7], [8], [9], [10].

Recent advancement in research in the field of Data Science and Artificial Intelligence has made it possible to develop a new technique called Data Prediction. Fortunately, data prediction has proven to be an excellent approach as it improves energy consumption and data transmission

The associate editor coordinating the review of this manuscript and approving it for publication was Gongbo Zhou.

efficiency by reconstructing data lost due to node failure [11], [12], [13].

There is a necessity in building sensor networks that takes into account the data transfer process of the sensor nodes. This unfortunately depletes the data transmission quality and lifespan of the network. Despite transmission tools such as cognitive radio modules and antennas, keeping a sensor network available for a long time requires a good data prediction strategy so as to limit data flow through the network. For WSNs dealing with a large number of sensors, several research works have been made to obtain an effective data prediction approach. Some of the works were directed toward solving the problem of energy consumption during data transmission [14], [15], [16] while other researchers were focusing on data reconstruction [17].

To improve the prediction potential, [6] proposes two main factors that have to be taken into consideration. Firstly, an investigation of a relationship between the sensing data is done, which gives further information about the spatio-temporal property of the data. The relationship between sensing data is quantified by its degree of correlation. The highest correlation is used to predict the data, which makes this factor an efficient way of detecting outliers. The second factor investigates the quality of data collection, which also helps to detect abnormal data.

Convolutional neural networks (CNN) and recurrent neural networks (RNN) are feed-forward neural networks that are generally used to extract spatial and temporal features required to improve model performance. Their broad application in spatio-temporal climate data processing, traffic flow prediction and network lifetime extension reveals CNN and RNN as suitable candidate for data prediction [18], reason why Cheng et al. [11], [19], used these neural networks to produce a multi-step data prediction model using historical data observations from different sensor nodes. As aforementioned, most data prediction studies provide single output at a time and few of them take into account the time-series distribution of the sensing data.

An important property of WSNs is their ability to collect and transfer time-series data in the form of sequences. Exploiting this characteristic will permit to forecast unavailable information over a given period of time.

In this paper, we study the strategy to design sequential prediction model, that exploits the time distribution property of the data collected by the sensors, to effectively reconstruct unobtainable data observations. Our objective is to use the correlation between the neighborhood sensors and then provide prediction of a given sensor at a time while prevent network depletion. Thus, we provide a mathematical formulation of the data prediction problem based on sequence-to-sequence prediction and use the concept of encoder-decoder to built the model. The deep-learning model developed exploits the spatio-temporal distribution of the sensing data to accurately reconstruct data loss during nodes failure. Indeed, predicting the sensing data for a set of nodes within a period has a crucial advantage that the sensor data collected at the

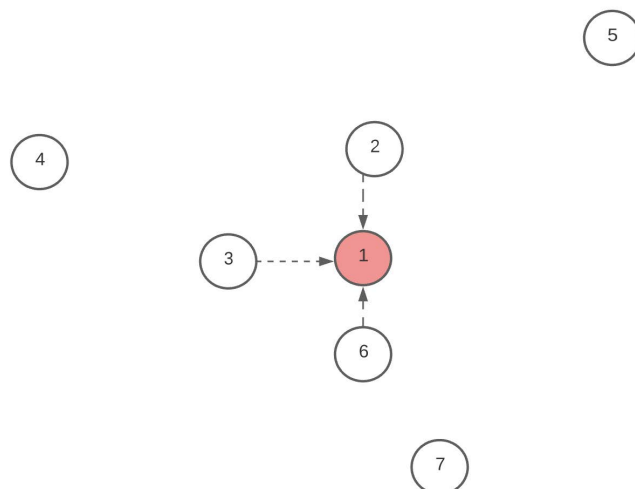


FIGURE 1. General example illustrating our approach of data prediction.

node level will only be transferred if the difference between the predicted value and the observed value of the data is greater than a given threshold state. By doing so, data transmission is limited to a reasonable quantity of information transfer, to the base station. To reduce data traffic flow across wireless network and thereby the energy consumption, the encoder-decoder model proposed is designed to predict two times more output. Compared with Long-Short Term Memory (LSTM) and auto-regressive model, simulation results show that ours model outperforms them in terms of Root Mean Squared Error (RMSE) and network lifetime extension.

A. GENERAL INTUITION

Figure 1 illustrates a simple example to describe the approach we intend to implement. In the figure, we describe the different nodes (from 1 to 7) as the set of nodes deployed in the coverage area. The red node (node 1) represent a default node at given time steps $t, t + 1, t + 2, \dots$, on the other hand the nodes $\{2, 3, 4, 5, 6, 7\}$ are in activity. We then look over the near-by nodes surrounding the default node 1. The intuition is that closer nodes collect almost the same information. Assuming that, each node have similar technical configuration, and using historical data of closer nodes $\{2, 3, 6\}$ at $\dots, t - 2, t - 1, t, t + 1, t + 2, \dots, t + n$ we are able to perform predictions on node 1 in our case at time t and $t + 1$, respectively.

Figure 2 illustrates a simple flow chart of our data prediction model, moving from the simple stages of data pre-processing to the final predictions. Initially the raw data is converted into structured data frame, which is then gotten rid of noise(outliers), using a two stage pre-processing technique. The first stage is a **Z-score** denoiser (Z_t) which remove outliers base on the Z-score of a set of data values. The second stage of pre-processing is an anomaly detection function, using the moving-average principles (MA), which we describe later in this paper. The pre-processed data is converted into a supervised learning problem, using the principle

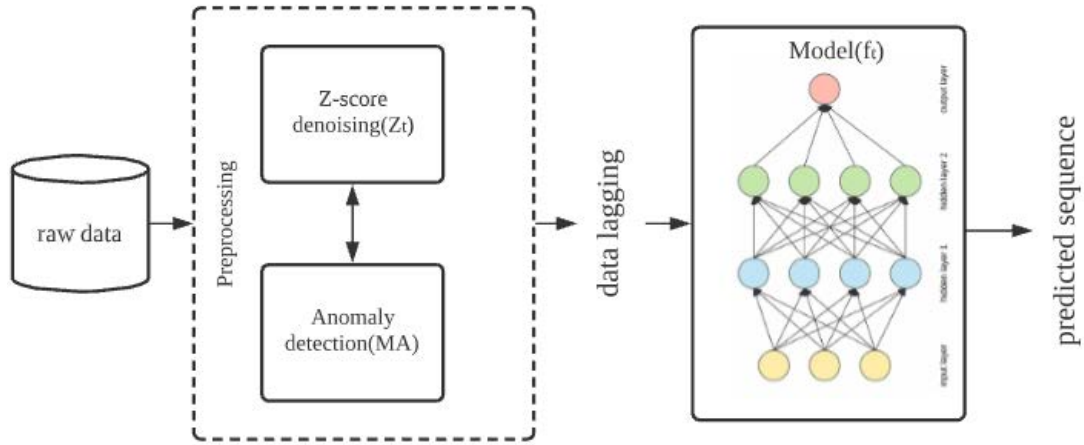


FIGURE 2. Simple flow chart of our proposed model, from left to right, we perform different stages of pre-processing to improve on predictions.

of **lagging**. Once this is done the structured and *clean* data is now fitted into our encoder-decoder model f_t , to perform predictions.

The rest of the paper is organized as follows. Section II is devoted to related work on data prediction in WSN and the identification of the challenges and related solutions. In section III, we introduce a clear mathematical formulation of the data prediction problem and a proposed model associated to the latter. We also introduce in this section the technique used to perform data pre-processing and data lagging in order to convert the data prediction problem into a supervised machine learning task. Next, in section IV, we evaluate the effectiveness of the proposed method through intensive simulations and quantify the model performance in terms of several metric score and energy required for data transmission. Finally, section VI concludes this work and provides futures directions.

II. RELATED WORK

There are numerous studies, proposing approaches to the data prediction problem. As discussed previously, the main constraint is an inefficient data transmission and energy consumption profile of the network. The related work exposed below is divided into two classes based on the mechanism used to remedy this problem.

A. NON-MACHINE LEARNING APPROACHES

Traffic flow analysis that takes into account the nearby environment and the trend between the data has been studied in the field of data prediction. The main property of traffic flow data is that it has a strong spatial correlation with data just like WSNs data. Thus, in [20], the authors put forward a model based on an encoder to capture traffic flow patterns. The experimental setup revealed that, model learning spatio-temporal data distributions perform better than normal models. In the same vein, the effectiveness of implementing the different data prediction models proposed

in real-world data sets is investigated [21]. The result of this review revealed that most of the models are too complex to be applied in real life. In [21], a simpler model known as derivative-based prediction (DBP), which relies on the hypothesis that the data delivered by sensors can be computed using a simple linear equation for both short and long-term predictions is proposed. The aforesaid technique has an inconvenience that it assumes the sensor data to be rid of erroneous values which is not always the case.

A Kalman filter algorithm to forecast data loss due to node disability is proposed in [12]. This algorithm assumes a 1-step Markov model of the data series $\{x_1, x_2, \dots, x_k\}$ as follows: 1

$$x_{k+1} = F_{k+1}x_k + GW_k, \tag{1}$$

where F_{k+1} is the Markov transition matrix, W_k is a white noise process, and G is a noise covariance matrix. Their experimental results confirm the accuracy of the algorithm. However, the filtering algorithm is not robust and thus the predicted values turn out to be very unstable.

In 2017, [16] proposed a solution to solve the problem of energy consumption during electronic transmission of data by sensors. The authors described the energy consumption as a function of the distance to the base station. Sensor nodes far away from the base station will require more energy for data transmission.

To overcome this problem, the solution proposed was based on the Milne-Simpson method equation to predict the value of a sensor given its four previous states as defined in (2) and (3)

$$\hat{y}_{i+1} = y_{i-3} + \frac{4h}{3} (2y'_{i-2} - y'_{i-1} + 2y'_i), \tag{2}$$

$$\hat{y}_{i+1} = y_{i-1} + \frac{h}{3} (y'_{i-1} + 4y'_i + y'_{i+1}). \tag{3}$$

where 3 represents the corrector formula, y'_i, y'_{i-1}, y'_{i-2} represents 3 sensor data at different time steps, \hat{y}_{i+1} the predicted value at time-step $i + 1$ and h the step-size. Basically at time-step i , the cluster heads computes the predicted output \hat{y}_{i+1}

and compare it with the actual sensor value using 4.

$$|y_{i+1} - \hat{y}_{i+1}| < \epsilon \quad (4)$$

with ϵ , defining the threshold value of the comparison process. The authors reveal that varying the step size, considerably improves the prediction accuracy, thus less data transmission is required. The cons of this approach arise when the step size is very big as it leads to abnormal predicted values.

In [15], the authors suggest a method of data prediction using the Adam Bashforth-Moulton method (ABM) as a technique to improve energy consumption in hierarchical WSNs architectures. In the above-mentioned technique, the cluster head predicts the sensor data using the ABM equation defined in (5)

$$\hat{y}_{i+1} = y_i + \frac{h}{24} (55y'_i - 59y'_{i-1} + 37y'_{i-2} - 9y'_{i-3}). \quad (5)$$

where, y and \hat{y} are variables defined exactly as in 2. Once again, data is only transferred if there is a significant difference between the predicted value and the observe data value (defined in 4). The result released by this approach confirmed a better performance than the Milne-Simpson method, but it said to have some drawbacks. This method is inefficient when there isn't sufficient data for prediction. As an attempt to improve the lifetime of WSNs, the hierarchical fractional least mean-square filter is developed in [22] to accurately predict the sensing data, based on weight coefficient matrices of two layers sub-filters. Taking the energy consumption and prediction error as metrics of performance measurement, the experimental results of the work illustrate a decrease in the quantity of energy consumed and an increase in the compression rate of the data. This leads to a decrease in data transmission.

B. MACHINE LEARNING APPROACHES

With the recent advancement in the field of machine learning [23], established a clustering plan to limit the connection between sensor nodes thereby increasing the network lifetime [24], [25]. Data clustering was achieved based on the spatio-temporal correlation of the different sensor nodes. One possible limitation of this approach is that, in the absence of a suitable data denoising method to remove any abnormal values, data correlation could be significantly altered, thereby impacting the clustering outlook. In [26], the LSTM model is presented for traffic flow analysis which they further compared with a statistical model called ARIMA, and finally concluded that LSTM has a good prediction performance than classical statistical models. One year later, [27] established a model based on the concept of deep learning to solve a traffic flow optimization problem. The model was defined with two extraction layers, with the first one extracting features and the second one defining patterns between the raw data to perform predictions. A self organizing map (SOM) model is proposed in [17] to overcome the problem of sensor nodes deterioration

due to large data transmission in smart city systems. The authors designed two SOM models with one learning the time-series patterns between the data at each time-step t , and define it as a regressor. The second SOM model learns the first difference of the time series, defined as

$$y_i = x_{i+1} - x_i. \quad (6)$$

The authors further implemented a transition matrix that links the first and the second SOM-models, and from which we can always obtain the regressor x_{i+1} at time step $i + 1$ as the sum $x_i + y_i$. The experimental result of this technique was measured using the root square mean error (RSME) and the performance revealed a good prediction accuracy provided the step size doesn't exceed the regressor size. Unfortunately, the proposed method provide prediction results for only one sensor at a time and doesn't take into consideration the correlation between the neighbourhood sensors within a given node.

To improve the results obtained in the previous studies, the authors in [19] initiated an approach that investigates the spatio-temporal property of sensing data to solve the problem of data loss due to node failure during the transmission process. Based on deep learning techniques, the authors designed a stacked bidirectional LSTM model with two layers, capable of predicting in a multi-step fashion, the sensing data. The overall performance of the model was measured using several metrics like RSME and the mean absolute error (MAE), just to name a few. The excellent performance of this model reveals that time distribution patterns need to be taken into consideration during data prediction. This is why similar researchers in [11] proposed an even better approach to solve the problem of node failure. Using almost the same neural network architecture, the authors introduced a one-dimension (1D) convolutional layer which had as role to extract features from the sensing data before feeding in the bidirectional LSTM. A comparative analysis was further done to demonstrate the effectiveness of this model with several models proposed in [19].

LSTM and AdaBoost (Adaptive Boosting) have been combined in [28], for the prediction of temporal and spatial data in order to reduce the energy consumption of the network, and to classify the failed nodes which allows to increase the quality of the data to be transmitted to the base station. To achieve this, three concepts were put forward, node clustering, early redundant data prediction and classification of failed nodes. As far as prediction and classification are concerned, they were carried out in three steps. In the first step, the LSTM is applied and its prediction result is passed to AdaBoost. In the second step, AdaBoost reduces classification errors by detecting weak predictions and assigning weights to them at each iteration. Finally, the third step is dedicated to merging the classification results from AdaBoost and the prediction outputs from LSTM to produce a strong and efficient classification.

C. DEEP LEARNING FOR DATA PREDICTION

In this part of the paper, we present other related work associated to deep learning to solve the data prediction problem. Zhang et al. [29] proposed a sequence-to-sequence imputation model using a Bi-directional LSTM network to memorize past and future prediction at a given time t . The authors also made use of a sliding window in order to generate more observations to improve the training process of the model. The experimental results reveals a good performance of the architecture as compared to other statistical models like ARIMA. Authors [30], performed a comparative analysis of deep learning and machine learning techniques on wireless sensor networks dataset, for intrusion detection and prevention systems. The result of this experimentation reveals that, deep learning classifiers perform better than machine learning classifier as far as intrusion detection results are concerned. The result of this work show the effectiveness of deep learning models to solve data prediction problems. As an attempt to improve energy efficiency in wireless network, Mohanty et al. [31], proposed a model based on RNN and LSTM to reduce data transmission by performing data prediction. The experimental results showed a decrease in the signal overhead average delay with a reduction in the amount of data transmitted as compared to simple deep neural network. In the same vein, Weisfeiler-Lehman kernel technique and Dual Convolutional Neural Network (WL-DCNN) have been combined in [32], for data prediction of failed links in order to increase the lifetime of the network while ensuring its resilience. The studies are conducted in a dense and dynamic IoT network context. To carry out the prediction task, the strategy is to use the Weisfeiler-Lehman kernel to extract and label subgraphs which are then transferred to the WL-DCNN for prediction.

D. LSTM GENERAL OVERVIEW

A LSTM is composed of a series of repeated cells and one of them is presented in Figure 3. Unlike RNN, LSTMs are composed of gates, we have the **input**, **forget** and **output** gates. Each gate has a weight matrix and a bias vector denoted as W and b respectively.

- **Forget gate:** Firstly, the LSTM concludes on which information to be forgotten from the cell at a given time-step t (see (7)).

$$f^{(t)} = \sigma \left(W_f \left[h^{(t-1)}, x^{(t)} \right] + b_f \right). \quad (7)$$

The sigmoid function σ , is responsible for this decision, taking as input the previously hidden layer $h^{(t-1)}$ and the input $x^{(t)}$. The forget gate is defined as $f^{(t)}$.

- **Input gate:** The sigmoid function of the second layer (Figure 3), concludes on which inputs to let in ((8)) and the tanh function weight the input's level of importance in the information ((9)).

$$i^{(t)} = \sigma \left(W_i \left[h^{(t-1)}, x^{(t)} \right] + b_i \right), \quad (8)$$

where $i^{(t)}$ represents the input gate.

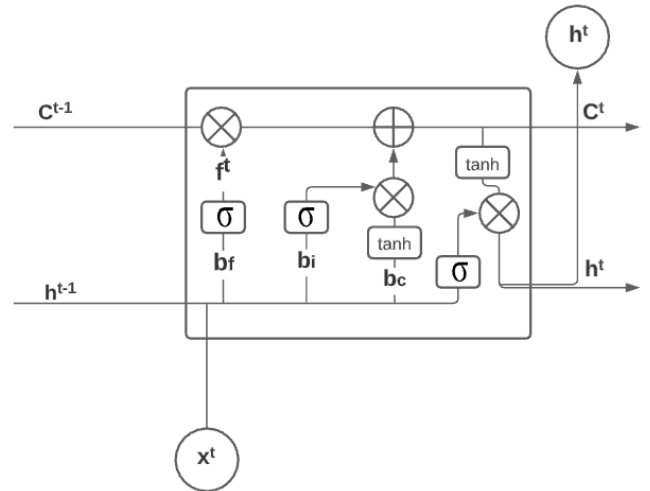


FIGURE 3. To solve the problem of vanishing gradient descend, LSTM quantifies how much previous data it should remember or discard reproduced from [33].

- **Cell state:**

$$C^{(t)} = \tanh \left(W_C \left[h^{(t-1)}, x^{(t)} \right] + b_C \right). \quad (9)$$

- **Output gate:** A sigmoid function is again used to compute what portion of the cell state goes to the output $o^{(t)}$ ((10)) and the \tanh function is used to conclude on the value of the hidden state $h^{(t)}$ ((11)).

$$o^{(t)} = \sigma \left(W_o \left[h^{(t-1)}, x^{(t)} \right] + b_o \right), \quad (10)$$

$$h^{(t)} = o^{(t)} \tanh \left(C^{(t)} \right). \quad (11)$$

III. SYSTEM MODEL

This section is devoted to the mathematical formulation and implementation of the solution to study the data prediction problem in WSNs. We also introduce the Intel indoor data set [34], necessary for the model implementation, after which we figure out the techniques used to perform data denoising and anomaly detection in the data set. Furthermore, a correlation study is also done, in order to check the relationship between the different sensor nodes

A. PROBLEM FORMULATION

In the following, we formulate the data prediction problem based on the knowledge of sequence-to-sequence models. The mathematical proposal is based on the study provided in [11].

We define variables used in our model as follows:

- V is a set of sensor nodes across the WSN;
- $N(v)$ is the set of neighbouring nodes closest to the default sensor node;
- $\{y^{(t)}\}_{n,m}$ is the set of all data value collected by sensor m of node n at a time t ;
- $(\hat{y}^{(t)}, \hat{y}^{(t+1)})$ is the predicted target sequence of the default node;

- $En(h^{(t-1)}, c^{(t-1)})$ is the vector state of the encoder at time step $t - 1$;
- $EnDe$ is the encoder-decoder function.

Consider N -nodes spatially distributed across the coverage area defined in (12).

$$V = \{v_1, v_2, \dots, v_N\} \quad (12)$$

We assume that each nodes have M -sensors and we further define the set of data collected by these sensors, as in (13).

$$\{y^{(t)}\}_{n,m} \quad 0 < n \leq N, \quad 0 < m \leq M, \quad t > 0 \quad (13)$$

Suppose that at time a greater than t a set of nodes doesn't transmit information (due to node failure) and at the same time, we have a set of available nodes. The data prediction problem aims to reconstruct data loss in after time-step t . Assuming that the neighbouring nodes $N(v)$ collect similar information as the unavailable node, the data reconstruction is done based on the historical information of the default node and the neighbouring working nodes.

B. ENCODER-DECODER MODEL

1) PRINCIPLE

The goal of our work is to reconstruct data loss due to node failure, using the data prediction method. At the node level the data collected are now compared to the predicted data values. Any significant difference between the observed value and the predicted value will lead to a transmission of information to the base station, otherwise no data is send to the station, hence limiting data transfer to a certain extend and by so doing, saving energy consumption by nodes. Our hypothesis is that increasing the number of predicted values will also play a role in decreasing the energy consumed by these sensor nodes [35]. To reach our objective, we rely on predicting outputs in the form of sequences as well (much input-many output model). Next, we introduce a notion called **encoder-decoder** models [36]. Primarily, the two building blocks of encoder-decoder models are the **encoder** and **decoder**.

- **Encoder:** It takes in the input sequence and processes it in the form of an information vector (or state), enclosing all the knowledge of the entire data set.
- **Decoder:** The decoder takes as input the context vector and makes predictions (output).

Each block is composed of a series of LSTMs, interconnected together. The fed inputs from these series of LSTM cells helps the encoder to encapsulate all the information which is stored in the hidden state $h^{(t)}$ and the cell state $c^{(t)}$, constituting the vector state.

The decoder block is also composed of LSTMs, whereupon it is activated by the vector state. The output of the decoder is defined in such a way that, each predicted value at time t represents the t^{th} component of the target sequence. To better demonstrate how encoder-decoder models work, Figure 4, shows a three-input and two-output sequence-to-sequence model. The inputs x_1, x_2 and x_3 are fitted into the encoder

block and a vector state composed of the input's information is obtained.

At time step t_1 , the input fed into the decoder is a special initial state $(c^{(t_0)}, h^{(t_0)})$, meaning the start of the output sequence. The encoder makes use of this input and the internal state $(c^{(t)}, h^{(t)})$ to produce the first output y_1 . Then at time step t_2 , the output y_1 at time step t_1 , is fed as input in the decoder cell, to obtain the output y_2 . The process is performed repeatedly with concern to the number of elements that constitute the output sequence.

Training in sequence-to-sequence models is performed in the same way as normal LSTMs or Recurrent Neural Network (RNN).

a: BASELINE MODELS

In order to showcase the effectiveness of our model, we decide to develop two benchmarks models, one based on auto-correlation and the second one is a simple LSTM.

i) AUTO-REGRESSIVE MODEL

A linear regression, models the output value as a linear combination of the input as defined in equation 14, where \hat{y} presents the predicted output, x is the input and b_0, b_1 are the set of weights.

$$\hat{y} = b_0 + b_1x. \quad (14)$$

Similarly, linear regression models can be applied to time series problems, as they perform very well on predicting next time-steps $t + 1$ using information from previous times steps $\{t, t-1, \dots, t-n\}$ (n is an integer) as defined in equation 15.

$$\hat{y}_{t+1} = b_0 + b_1y_{t-1} + \dots + b_ny_t. \quad (15)$$

Because the regression model uses data from the same input variable at previous time steps, it is referred to as an **auto-regression**. An auto-regression model makes an assumption that the observed values at previous time steps are useful to predict the value at the next time step. If both variables change in the same direction, this is described as a positive correlation. If the variables move in opposite directions as values change then this is called negative correlation.

2) MODEL DESIGN

In previous section we presented the concept of encoder-decoder. Based on this principle, the encoder block f in our model takes as input the passed data for both kinds of nodes (neighbouring nodes, and default nodes) and a vector state is obtained as seen in (16).

$$En(h^{(t-1)}, c^{(t-1)}) = f(y^{(t-1)}, \dots, y^{(t-H)}, N(v)). \quad (16)$$

Here H defines the number of time-steps for data collection.

- **Prediction of $\hat{y}^{(t)}$ and $\hat{y}^{(t+1)}$:** The decoder g , takes as input, the vector state $En(h^{(t-1)}, c^{(t-1)})$ and an initial state $(h^{(0)}, c^{(0)})$ to produce the t^{th} component of the missing data value ((17)). The predicted output $\hat{y}^{(t)}$ is

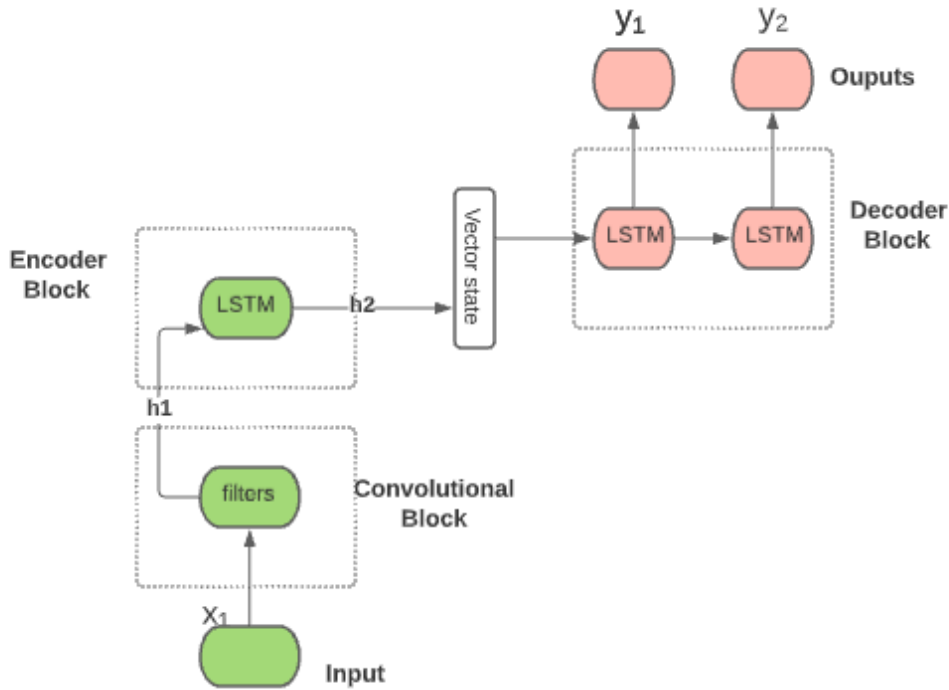


FIGURE 4. Figure represents a simple architecture, of proposed model. In the encoder block, the inputs are discarded and we are left with the hidden h_t and cell c_t states (Encoder vector). In the decoder block, only the outputs are left y_t .

fitted back into the decoder to yield the t^{th+1} component of the target sequence ((18)).

$$\hat{y}^{(t)} = g \left(En \left(h^{(t-1)}, c^{(t-1)} \right), \left(h^{(0)}, c^{(0)} \right) \right), \quad (17)$$

$$\hat{y}^{(t+1)} = g \left(En \left(h^{(t-1)}, c^{(t-1)} \right), \hat{y}^{(t)} \right). \quad (18)$$

(19) generalizes the formulation and with this approach, we can predict a sequence of H values per sequence. For this work, we limit ourselves to predicting two values per sequence.

$$\left(\hat{y}^{(t)}, \hat{y}^{(t+1)} \right) = EnDe \left(y^{(t-1)}, \dots, y^{(t-H)}, N(v) \right). \quad (19)$$

C. TEST DATA SET

The data set used in this study [34] was assembled by Intel Berkeley Research Laboratory with the use of sensors called Mica2Dot. The raw data is made up of 2.3 million queries of sensory data accessed from 54 nodes, with each node collecting information on temperature, humidity ranging from 0 – 100%, light with values between the range 0 – 2000 and voltage varying from 2 to 3 volt. A record of node Id (a number that identifies the nodes), the date time and the timestamp were also associated with the data set. The spatial distribution of the sensor nodes can be found in [19].

From the first observation, some nodes seem to be very close to one another in locations. For instance nodes (4, 3, 2) or nodes (4, 7, 10), with the hypothesis being that closer nodes collect similar information (data redundancy). The WSNs architecture following this node distribution is the

hierarchical clustering architecture with a central node which can be considered as *node4* where all the information are forwarded before transferring to the sink. Sensor nodes default or data transmission abnormalities usually arise with data outliers. Also, the location of a node can lead to erroneous data collection and transmission. In order to get rid of abnormal values, a good data pre-processing needs to be done.

1) DATA DENOISING

Due to limited resources and the large data set in our possession, the proposed method for outlier detection of our work is by the means of Z-score as described in [37].

Z-score quantifies the abnormality of observation when the data follows a normal distribution. As defined in (20), the Z-scores are the number of standard deviations above and below the mean values of the data points.

$$Z = \frac{x - \mu}{\sigma}. \quad (20)$$

where x represents, a data point, from a given sensor, μ the mean value of the sensor's data observation and σ its corresponding standard deviation. For example, a Z-score of +1 signifies that the data value is one standard deviation above the mean and a score of -1, which means the value is one standard deviation, below. A standard benchmark value to detect outliers is a Z-score is ± 3 .

2) ANOMALY DETECTION USING THE MOVING AVERAGE

A moving average (MA) is an indicator that is commonly used in technical analysis. It helps to smoothen the data over a

specific period of time by creating constantly updated average values. Reference [38] has proven that moving average is well known to detect anomalies. In this paper, we use MA to detect and remove unnecessary data values so as to improve the accuracy of our model. For a given matrix of window size W , we perform a convolution on the original data to obtain the MA. we proceed by computing the residual res as the difference between the actual value y and its MA, y_{avg} . Next we calculate the standard deviation σ of the residual. If the a data point lies between $y_{avg} \pm \sigma \times k$, it is considered normal else it is considered an anomaly. k here represents a constant value used to vary the length of interval for the anomaly detection process. Figure 5 illustrates how anomaly detection, is performed on temperature values of node10 with a given window size of $W = 50$ and $k = 3$. The red dotted points, representing the set of outlier values. Also notice how the graph moving average of the temperature values (in green), follows almost the same patterns of the actual observe values (in black).

3) DATA PRE-PROCESSING

As the sensory data has a different range of values to have an accurate and smooth model training, the data values are normalized and re-scaled to the range (0, 1) using the min-max scaler function defined in (21).

$$\min - \max(x) = \frac{x - x_{min}}{x_{max} - x_{min}}. \quad (21)$$

where x , represents the data points, x_{min} and x_{max} denote the minimum and maximum values of the raw data for a given sensor respectively. The advantage of data normalization is that it eases feature extraction to perform better data correlation.

To quantify the correlation between data we use the Spearman correlation coefficient defined in (22), which was proposed in [39].

$$\rho = 1 - \frac{\sum 6 d_i^2}{n(n^2 - 1)}. \quad (22)$$

With d_i^2 being the squared difference between two variables of each observation, and n represents the number of data points. For instance, computing the Pearson coefficient of temperature and humidity using (22), we have $\rho = -0.405$.

Tables 1 and 2 presents examples of the correlation matrix of the temperature values of nodes (4, 3, 2) and (4, 7, 10) respectively.

We notice an overall strong positive correlation between the temperature of the different nodes. This is probably because of the closeness of the nodes in terms of spatial distribution.

4) LAGGING TIME SERIES DATA

To convert a time series problem into a supervised machine learning problem, we use a lag operator. A lag operator is a function which shifts data values of time series such that it

TABLE 1. Correlation matrix of temperature for nodes (4, 3, 2).

Correlation	Temp 4	Temp 3	Temp 2
Temp 4	+1.000	+0.859	+0.964
Temp 3	+0.859	+1.000	+0.794
Temp 2	+0.964	+0.794	+1.000

TABLE 2. Correlation matrix of temperature for nodes (4, 7, 10).

Correlation	Temp 4	Temp 7	Temp 10
Temp 4	+1.000	+0.968	+0.929
Temp 7	+0.968	+1.000	+0.875
Temp 10	+0.929	+0.875	+1.000

produces a new set of inputs, matches with the original time series data.

Lags are very convenient for time series analysis due to a phenomenon called auto-correlation, defined as a high tendency that the data value at time step t is strongly correlated with the data value at time step $t - 1$. Generally, auto-correlation is used to identify trends within time series data. The lagging operation is described in (23) with the generalized form defined by (25)

$$Ly_t = y_{t-1}, \quad (23)$$

$$L(Ly_t) = Ly_{t-1} = y_{t-2}, \quad (24)$$

$$\text{generally, } L^s y_t = y_{t-s}. \quad (25)$$

where L is the lagging operator, y_t is a given observation at time t and s is the number of shift of an observation.

D. MODEL ARCHITECTURE

As stated in the problem formulation (section III-A), we use the neighbouring sensor nodes to reconstruct the data values loss by faulty nodes. The study of the inter-correlation (correlation between data points of two or more different sensor nodes) as presented in Table 2, is an argument for the choice of nodes 4, 7 and 10. To perform our data prediction study, Figure 6 shows the proposed network architecture. We choose node 7 and node 10 to aid node 4, during the prediction process. The architecture therefore takes as inputs data values from nodes 4, 7, 10 on different models denoted by the time steps T_1 , T_2 and T_3 respectively. The motivation behind this architecture is to ensure that the information from each node can be treated independently.

Inspired by [11], it has been proven that taking data inputs at different time windows, yields a better performance. The model architecture is sub-divided into 4 blocks: the inputs block, convolutional block, the encoder-decoder block and the output-block.

- The input block is composed of temperature values (temp) of node A , B and C , at different time windows (T_1 , T_2 , T_3) respectively, which are inserted into the convolutional block.
- There after we have our pre-processing block comprising of the **Z-score** denoiser and the anomaly detection, to effectively remove outlier values from the data points.

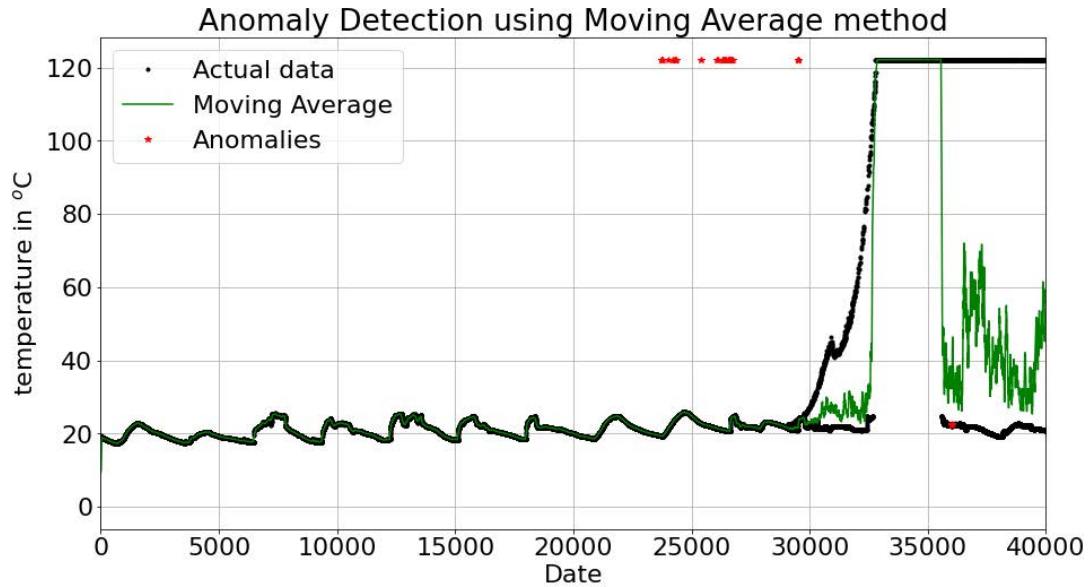


FIGURE 5. Detecting anomalies on the temperature values of node 10.

- The convolutional block (denoted as **conv** block) is subdivided into a 1-dimension (1D) convolutional layer and a 1D max pooling layer (1D max pool). The choice of 1D is a result of the dimension of the input signal. Indeed, 1D signals with temporal ordering are better treated with 1D convolutional filters. By spatial we mean input signals with an order positioning, which is the case of time series data. The max pooling filter is used as a down-sampler to reduce abstract features that could result in over-fitting.
- The encoder-decoder block as described in section III-B, is divided into the encoder and decoder. The encoder memorizes all the information contained in the input signal and produces a vector state ($h(t)$, $c(t)$) for each of the models (represented by the cell named **vector state**). Each vector state is fitted into the respective decoders, and a sequence of temperature output is produced at time step t ($temp(t)$) and time step $t + 1$ ($temp(t + 1)$).
- To unify the predicted output, a merging layer is introduced which is fitted by the two inputs of each model, which in turn is **flattened** and becomes the input, of a fully connected layer (FC) to produce two outputs. Takes 6 inputs and produces 2 outputs

1) SUMMARY OF SIMULATION SET-UP

We made use of the intel indoor dataset. Initially we started, by performing the data pre-processing, data denoising, data lagging. Next we did a rigorous anomaly detection checking to remove outlier values from the data set. We finally transformed our cleaned data into a supervised machine learning problem using data lagging, where the objective was to predict the sensing data at two time step interval (t and $(t+1)$).

IV. EXPERIMENTAL RESULTS

In the final section, we present and analyze the result simulation of our experimentation. That being the case, we start by describing in Section IV-A, the hyper-parameters necessary for training the encoder-decoder model. In addition, we describe in Section IV-B, the variable selection process which account for the best fit of the model. We proceed by representing graphically the predicted output and compare it with the actual observations, for a set of selected nodes, described in the Intel data set. We conclude, by assessing our proposed model in terms of error metric performance.

A. MODEL TRAINING

Having built our model, we describe in this section the fixed parameters necessary for the training process. The data set is divided such that **20,000** data points are used for training and prediction is performed in **3000** observations. The established parameters are:

- **Epoch**: The epoch defines how many times the overall data set passes inside the model during the training phase. The number of epochs used to train the model is the *epoch number* = 10.
- **Batch size**: It measures the total number of observations that are processed after one iteration of training. The model utilises a batch size of 200 during the learning phase.
- **Optimizer**: We used the *Adam* optimizer due to its smooth and fast convergence rate.

We also define the different error metrics necessary to evaluate the model performance.

- **Mean Absolute Error (MAE)**, defined as the error quantifying the average magnitude of errors in a set of

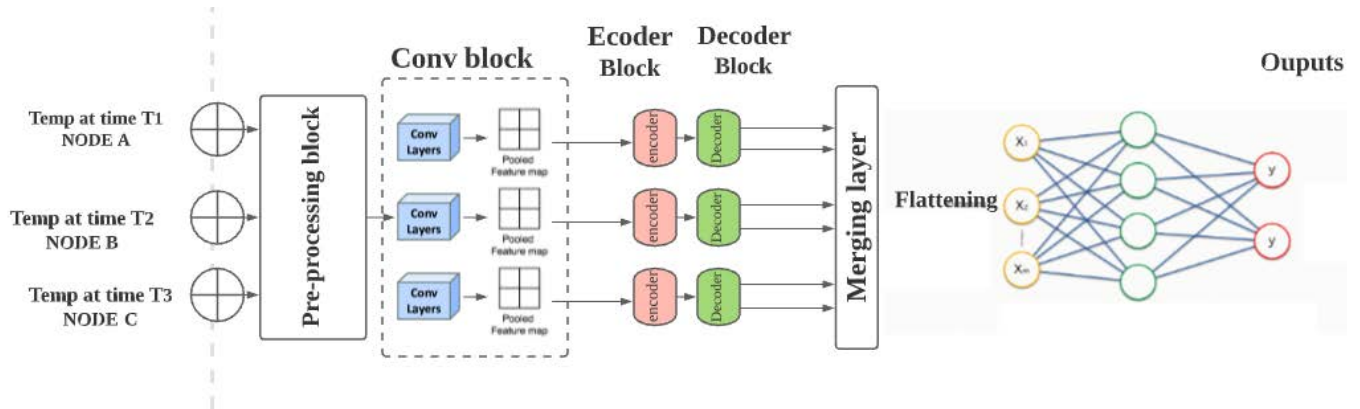


FIGURE 6. Model architecture.

forecast values.

$$MAE(Y, Y_{pred}) = \frac{1}{n} \sum_{i=1}^n |Y_i - Y_{pred,i}|. \quad (26)$$

- The **Mean Absolute Percentage Error (MAPE)** is a percentage measure of the difference between the actual output and the predicted output divided by the predicted output. Both MAE and MAPE are used in regression problems.

$$MAPE(Y, Y_{pred}) = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{Y_i - Y_{pred,i}}{Y_{pred,i}} \right|. \quad (27)$$

- The **RMSE** is a measure of the prediction error which defines how far the predicted values are from the observed values ((28)).

$$RMSE(Y, Y_{pred}) = \sqrt{\sum_{i=1}^n \frac{(Y_i - Y_{pred,i})^2}{n}}. \quad (28)$$

B. PARAMETERS SELECTION

The parameters required to be altered are the time step T , the dimension of the convolutional and pooling filters f , and finally, the number of neurons in the encoder-decoder layer. As already described, we take historical data for node 4, 7 and node 10 to predict the unavailable temperature value of node 4. As such, a variety of parameters can be obtained. Table 3 presents the set of fine-tuned parameters used during the training process.

The column named **input** is subdivided into 3 time steps denoting node 4, 7 and 10, respectively. The $RMSE$ defines the average error of the overall 3000 predictions. For example, when the time step equal (30, 10, 10) with kernel size 5, number of filter 1, pooling size 2 and number of neurons 32, the $RMSE = 0.5741$. We observe also that, the $RMSE$, is highly affected by the size and the number of filters. The highlighted time step (20, 10, 10) yields the smallest $RMSE$. This is because sensing data with moderate time steps turn to have better auto-correlation (correlation between the previous

and the next sensing data value) than bigger or smaller time steps. Smaller time steps sensing data, are not able to capture patterns between values in a very short interval, and result to low auto-correlation. On the other hand larger time steps sensing data have a bigger time interval to study the trend in the data and result to low auto-correlation as well, which causes a higher $RMSE$.

C. NODE SELECTION

We select the nodes via the linear relationship that exist between the default node and its corresponding neighbourhood nodes. The Spearman correlation defined in (22), revealed a good coefficient score for node-4, 7, 10, as presented in Table 1. The reason been that, closer nodes turn to collect almost similar information (provided they have the same sensor types). Figure 7 reports the data prediction result of node 4, for the temperature values. The green line represents the predicted data points while the red-crossed line represents the actual observed values. Notice a deviation of the predictions (peaks) around the 2300th and 2750th observations. Despite the closeness of the three sensor nodes, some abnormality in the prediction could arise due to the presence of other noise in the historical data. Also, the distribution of the nodes around near by obstacles could account for erroneous data points.

Various node combinations are performed, to evaluate our model and the prediction error is computed based upon the MAE the $MAPE$ and the $RMSE$ defined in (26), (27) and (28) respectively. Table 4 and 5 describe the error for predicting humidity and temperature values respectively. The notation **Nodes(A.B.C)** simply gives the ordering of the sensor nodes to perform prediction. For example Nodes(4.3.2) signify that the prediction of node 4 is guided upon by nodes 3 and 2.

Based on the results obtained in, tables 1 and 2, we can conclude that, sensor nodes with high correlation coefficient turn to have a low prediction error. Indeed, Table 2 showcases the correlation matrix of the set of nodes-4, 7 and 10. To further demonstrate the effectiveness of the proposed model, Figure 8 illustrates a comparison with our first baseline model built

TABLE 3. Output dimension of layers in encoder-decoder model.

Input Time steps	Conv		Pool		Encoder-Decoder	Error
	Kernel	Filter	Pool Size	Stride	Neurons	RMSE
10,10,10	5	1	2	1	32	0.5913
40,10,10	5	1	2	1	32	0.6199
30,10,10	5	1	2	1	32	0.5741
20,10,10	5	1	2	1	32	0.4079
20,10,10	5	1	3	1	32	0.4258
20,10,10	6	1	5	1	32	0.4788
20,10,10	4	1	5	1	32	0.4571
20,10,10	5	3	5	1	32	0.5410

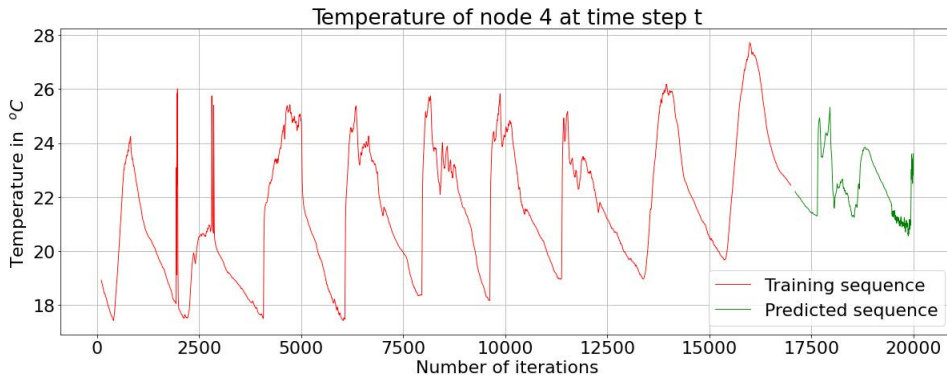


FIGURE 7. Predicted temperature value for node 4(in green) aided by node 3 and node 2.

TABLE 4. Error with humidity.

Nodes(A.B.C)	MAE	MAPE	RMSE
4.3.2	0.1975	0.6540	0.4043
4.7.10	0.0822	0.2137	0.1279

TABLE 5. Error with temperature.

Nodes(A.B.C)	MAE	MAPE	RMSE
4.3.2	0.1975	0.6540	0.4043
4.7.10	0.1731	0.5214	0.3452

under the concept of a simple LSTM. The blue line represents the temperature predicted values using our proposed model, the green line stands for the simple LSTM model and the red lines represents the actual temperature observations. The dark curve on the other hand represent the predicted output of our encoder decoder model using only the default temperature value of node 4. Notice how closed the predictions are to the predictions obtained using neighbouring node data values (in blue).The last 500 values of the data point are so closed together that, it becomes quite difficult for both models to capture this trends unless we decrease the lags considerably(so as to get more auto-correlated values). To reinforce our argument on model performance, Table 6 presents the prediction error of temperature, associated to the set of nodes 4, 3 and 2, for different models. For more evidence, we add our second benchmark model on auto-regression (*AutoReg*), represented graphically in figure 9. Note that, each model was trained under the same conditions as the

proposed model. We remark that, for the last 500 values, the auto-regressive model, turn to perform very well. We conclude that despite a higher *RMSE*, auto-regressive models, capture very well the auto-correlation at very small time interval than deep-learning models. In conclusion, we can say that our proposed model provides better results with decent error values within medium time steps interval and auto-regressive models turn to generalize well on smaller time lags.

In the same vein, Figure 10 and Figure 11 presents the prediction values of humidity for node 4, aided upon by nodes 7 and 10. The latter figure emphasizes more on the model comparison aspect of the results. Table 7 shows the humidity error measure of the three models as well and again, the performance of our proposed model, yield smaller error values.

The particularity of our proposed architecture is that the model can predict two-time steps simultaneously. The concept of encoder decoder model lies on the fact that, it,maps sequences of different length together. In our proposed model, we map three set of inputs and produce two output. This type of architecture is therefore able to predict outputs of different length as compared to fixed length producing outputs models.

V. STEP TOWARD IMPROVING ENERGY EFFICIENCY

One objective of our work is to ensure an efficient data transmission process, capable of saving an important amount of energy in order to sustain the life time of the WSN. As such,

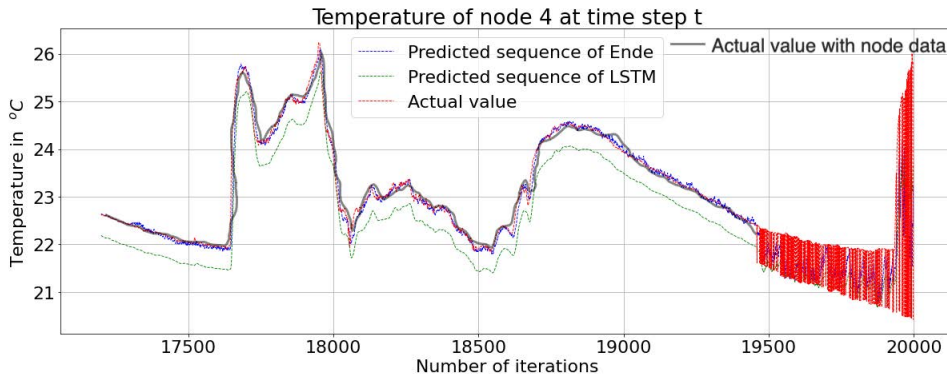


FIGURE 8. Temperature prediction results of node 4, for with the actual observed temperature. The dark curve showcase the predicted temperature value using only default data of node 4 .

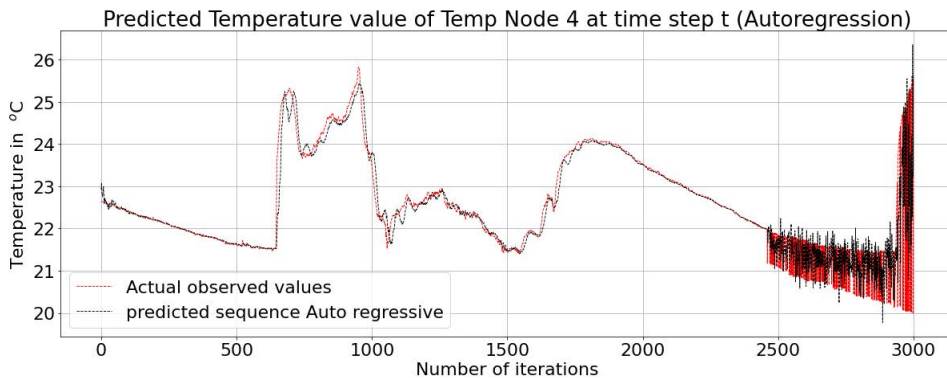


FIGURE 9. Predicted output of node 4 for the auto-regression model,for temperature.

TABLE 6. Model comparison of prediction error for temperature.

Models	MAE	MAPE	RMSE
<i>EnDe</i>	0.1975	0.6040	0.4079
<i>AutoReg</i>	0.2275	0.7079	0.4431
<i>LSTM</i>	0.2188	0.6342	0.4284
<i>BiLSTM</i>	0.2026	0.6203	0.4321

TABLE 7. Model comparison of prediction error for humidity.

Models	MAE	MAPE	RMSE
<i>EnDe</i>	0.615	0.7540	1.3241
<i>AutoReg</i>	1.4741	0.9517	2.3631
<i>LSTM</i>	0.9314	0.8336	2.1047

we quantify the effectiveness of our proposed model in terms of energy. In [35] the author proposed a radio model which defines the energy required for data transmission. Assuming the sensor nodes have the same communication radius R_C and an initial energy ϵ_0 , to transfer l -bit of information over a given distance d , the radio-energy transmission model is defined in (29) like adopted in [14], [40], and [41].

$$E_{TX}(l, d) = \begin{cases} l\epsilon_{elec} + l\epsilon_{fs}d^2, & \text{if } d < d_0 \\ l\epsilon_{elec} + l\epsilon_{amp}d^4. & \text{if } d \geq d_0 \end{cases} \quad (29)$$

where ϵ_{elec} defines the energy dispersed per bit of the transmitter, ϵ_{fs} and ϵ_{amp} is based on the transmitter amplifier model used. Keeping ϵ_{fs} , ϵ_{amp} and d constant, we notice that an increase in the number of bits l , transmitted, requires a high amount of energy for the process. In figure 14 we consider a set of nodes(in red). Using the data prediction approach described in section III, recovering the data from the nodes becomes easy. Also from the correlation tables (Tables 1 and 2), the set of studied node almost have the same information. If we consider clustering these sets of nodes into small clusters as shown in Figure 14, Instead of sending the information of all the nodes inside each clusters, we may consider only sending the data for a node with the smallest amount of energy possible such that the life-time of the network is maintained. Let’s formulate this idea mathematically:

From the subsection III-A, we have V nodes across the WSN and $N(v)$ the set of neighbouring nodes closest to the default node. Lets consider $c(v)$ the cluster associated to this neighbouring nodes, that is :

$$c(v) = N(v) \cup \{v_0\} \quad (30)$$

where v_0 represents our default node (basically we create a cluster composed of both the default node and the set of neighbouring nodes). Since in such cluster the data collected

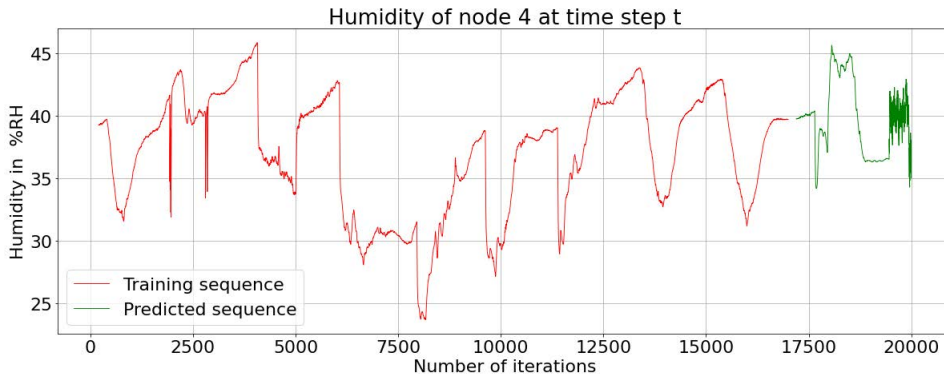


FIGURE 10. Graph presents the distribution of the first 3000 predicted values of humidity (in %RH) on the y-axis against the number of data points on the x-axis. Notice a regular pattern of the predicted values in green which follow almost the same pattern as the observed values (in red).

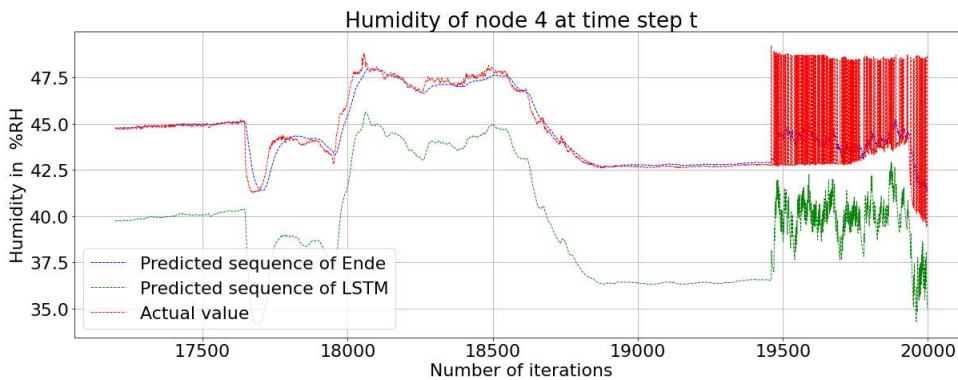


FIGURE 11. Predicted output of the Encoder decoder model is compared with the distribution of the actual humidity values of node 4.

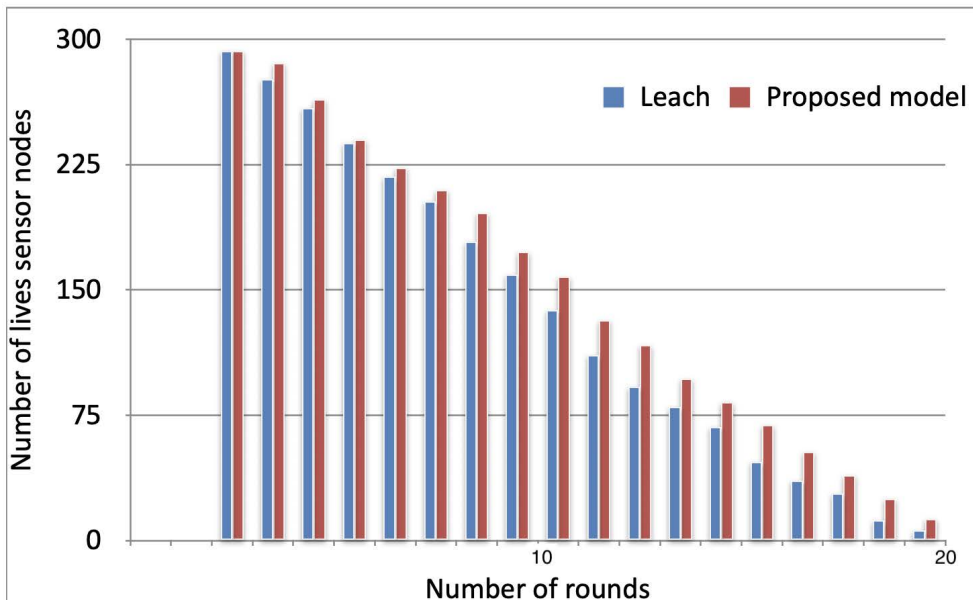


FIGURE 12. Number of living sensor nodes decreases over time (number of rounds). With an increase in number of rounds the our proposed model (in red), always have a higher number of living nodes than the LEACH model.

of each node is almost the same, then the energy required for transmission (using the radio model) will be slightly the same.

One can therefore consider the energy of a cluster as been the minimum energy of the set of sensor nodes in the cluster.

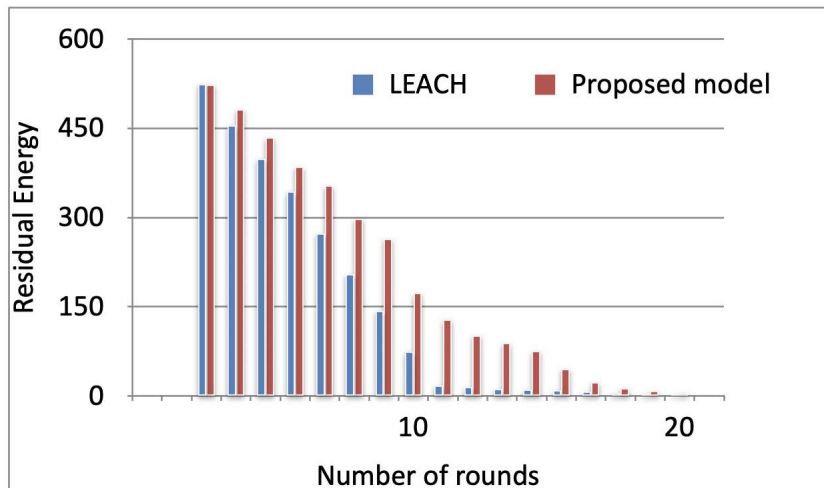


FIGURE 13. Residual energy of the WSN decreases over time (number of rounds). With an increase in number of rounds our proposed model (in red), always have a higher residual energy than the LEACH model.

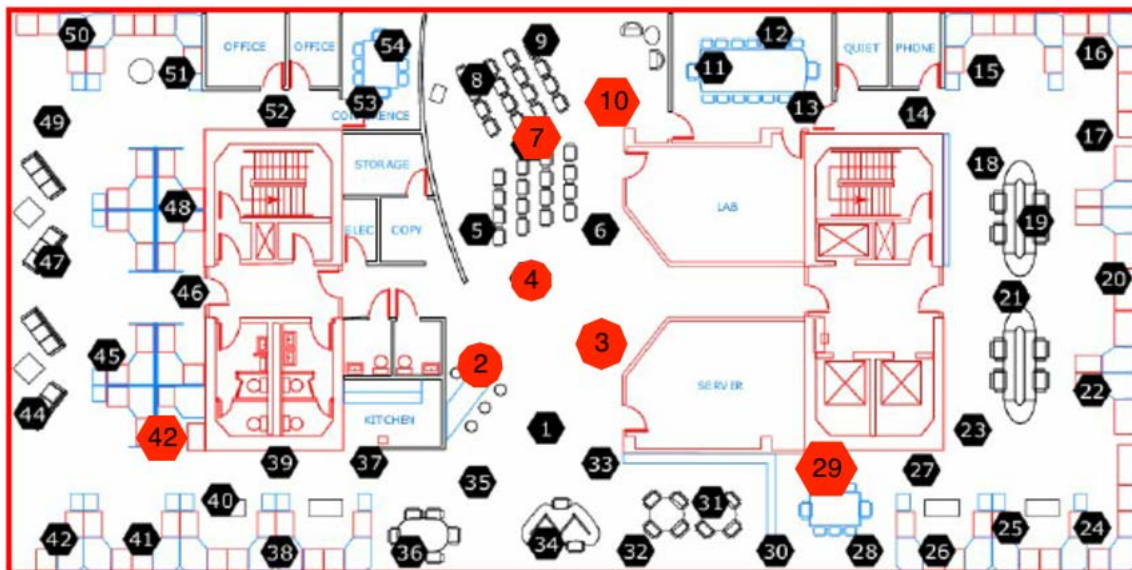


FIGURE 14. Red nodes represents the sets of nodes in which we want to perform data prediction [34].

Let $e_{c(v)}$ be the energy of cluster $c(v)$ and $\{e(v_i)\}_{0 < i < N}$ the energies of each node in the cluster $c(v)$, then

$$e_{c(v)} = \min (\{e(v_i)\}_{0 < i < N}) \tag{31}$$

Considering that, we have k -clusters in the WSN, the total energy required for transmission will be $E(c(v))$:

$$E(c(v)) = \sum_{i=0}^k e_{c(v)_i} \tag{32}$$

where $e_{c(v)_i}$ represents the energy of cluster i . Moreover, the life-time of a WSN decreases as the energy required for transmission increases.

We designed an experimental setup to show the effectiveness of our proposed model. In the experiment we consider

a approximately 300 nodes that we decide to spread across a 1000×1000 grid. Using our proposed algorithm, we study the evolution of the life-time of the sensor nodes over time as well as the residual energy of the WSN. Studying the life-time simply means counting the number of active nodes in the network over time. We also compare our algorithm with a well known algorithm in the field called LEACH [42]. Figure 12 shows how the number number of lives sensor nodes decreases with an increase in the number of rounds (for a total of 18 rounds). Initially both models have the same number of active sensor nodes, but our proposed model turns to have more living nodes than the LEACH algorithm. For example at round number 2, the number of living nodes for our proposed model is 286 nodes and that of the LEACH

algorithm is around 267. We further quantify our approach in terms of energy saved by the WSN. We define the residual energy as the energy left over in the WSN when an amount of node die (or get faulty). A high residual energy simply means that a small number of nodes got destroyed and as a result, the life-time of the network is maintained to a high level. As time goes on, many nodes die due to over-activity, hence the residual energy decreases. Figure 13 presents how our proposed model (in red) conserve energy better than the LEACH algorithm at each round.

VI. CONCLUSION

Different type of sensing data are collected by sensor nodes in wireless network. In our work, we attempted to study the variability and correlation between the different sensing data to reconstruct data loss due to node failure. We achieved the aforesaid task by building a one dimension convolutional neural network and an encoder-decoder model to effectively predict two sensing data output, per sequence of information. The idea behind this approach is that, predicting two times more output, considerably reduce data traffic flow across wireless network, which will directly account to a decrease in the energy required for data transmission.

The use case Intel indoor data set, was initially processed and re-scaled using a Z-score denoising and a min-max scaling function to prevent over-fitting during the learning phase of our model. The next step was to perform an operation called lagging to convert the data prediction problem into a supervised machine learning task. The inter-correlation study between the sensor node reveal a strong positive correlation between the set of nodes 4, 3, 2, 7, 10, which we used as a benchmark to implement the data prediction problem. After a rigorous set of parameter tuning, the experimental result showed that, our proposed model perform very well during the prediction phase with a considerably low root mean square error as compared to the previous related work. The encoder-decoder model is therefore capable of capturing both the spatial and the temporal feature of the sensing data.

Furthering the study of data prediction can rely on two field

- **Data Compression:** Based on the concept of principal component analysis, we could eventually reduce the dimension of the sensing data while keeping the most informative principal component to perform more accurate predictions.
- **Predicting N-outputs:** In this work we predicts two output per sequence. This can therefore be extend to an N-output data prediction problem, which will again result to an abrupt decrease in the data traffic flow across the network and make the WSN to be energy efficient.

REFERENCES

- [1] A. Mahmood, K. Shi, S. Khatoun, and M. Xiao, "Data mining techniques for wireless sensor networks: A survey," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 7, Jul. 2013, Art. no. 406316.
- [2] S. J. Ramson and D. J. Moni, "Applications of wireless sensor networks—A survey," in *Proc. Int. Conf. Innov. Elect., Electron., Instrum. Media Technol. (ICEEIMT)*, 2017, pp. 325–329.
- [3] A. N. Njoya, C. Thron, J. Barry, W. Abdou, E. Tonye, N. S. L. Konje, and A. Dipanda, "Efficient scalable sensor node placement algorithm for fixed target coverage applications of wireless sensor networks," *IET Wireless Sensor Systems*, vol. 7, no. 2, pp. 44–54, 2017.
- [4] A. A. Ari, A. C. Djedouboum, A. M. Gueroui, O. Thiare, A. Mohamadou, and Z. Aliouat, "A three-tier architecture of large-scale wireless sensor networks for big data collection," *Appl. Sci.*, vol. 10, no. 15, p. 5382, Aug. 2020.
- [5] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537–568, May 2009.
- [6] G. M. Dias, B. Bellalta, and S. Oechsner, "A survey about prediction-based data reduction in wireless sensor networks," *ACM Comput. Surveys*, vol. 49, no. 3, pp. 1–35, 2016, doi: 10.1145/2996356.
- [7] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical data compression in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 37–59, Jan. 2012.
- [8] A. N. Njoya, C. Thron, M. N. Awa, A. A. Ari, and A. M. Gueroui, "Lifetime optimization of dense wireless sensor networks using continuous ring-sector model," 2021, *arXiv:2104.06304*.
- [9] R. Hamidouche, Z. Aliouat, A. A. Ari, and A. Gueroui, "Mobile sink path planning in heterogeneous IoT sensors: A salp swarm algorithm scheme," *KSII Trans. Internet Inf. Syst.*, vol. 15, no. 6, pp. 2225–2239, 2021.
- [10] J. E. Z. Gbadouissa, A. A. Ari, C. Titouna, A. M. Gueroui, and O. Thiare, "HGC: HyperGraph based clustering scheme for power aware wireless sensor networks," *Future Gener. Comput. Syst.*, vol. 105, pp. 175–183, Apr. 2020.
- [11] H. Cheng, Z. Xie, Y. Shi, and N. Xiong, "Multi-step data prediction in wireless sensor networks based on one-dimensional CNN and bidirectional LSTM," *IEEE Access*, vol. 7, pp. 117883–117896, 2019.
- [12] R. Avinash, H. Janardhan, S. Adiga, B. Vijeth, S. Manjunath, S. Jayashree, and N. Shivashankarappa, "Data prediction in wireless sensor networks using Kalman filter," in *Proc. Int. Conf. Smart Sensors Syst. (IC-SSS)*, Dec. 2015, pp. 1–4.
- [13] M. Aboubakar, M. Kellil, A. Bouabdallah, and P. Roux, "Using machine learning to estimate the optimal transmission range for RPL networks," in *Proc. IEEE IFIP Netw. Oper. Manage. Symp.*, Apr. 2020, pp. 1–5.
- [14] C. Titouna, A. A. Ari, and H. Moumen, "FDRA: Fault detection and recovery algorithm for wireless sensor networks," in *Proc. Int. Conf. mobile web Intell. Inf. Syst.* Cham, Switzerland: Springer, 2018, pp. 72–85.
- [15] M. M. Islam, Z. Al Nazi, A. B. M. A. Hossain, and M. M. Rana, "Data prediction in distributed sensor networks using Adam Bashforth Moulton method," *J. Sensor Technol.*, vol. 8, no. 2, p. 48, 2018.
- [16] M. M. Islam, Z. A. Nazi, M. M. Rana, and A. B. M. A. Hossain, "Information prediction in sensor networks using Milne-Simpson's scheme," in *Proc. 4th Int. Conf. Adv. Electr. Eng. (ICAEE)*, Sep. 2017, pp. 494–498.
- [17] A. Russo, F. Verdier, and B. Miramond, "Energy saving in a wireless sensor network by data prediction by using self-organized maps," in *Proc. Comput. Sci.*, vol. 130, pp. 1090–1095, Jan. 2018.
- [18] Q. Liu, B. Wang, and Y. Zhu, "Short-term traffic speed forecasting based on attention convolutional neural network for arterials," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 11, pp. 999–1016, 2018.
- [19] H. Cheng, Z. Xie, L. Wu, Z. Yu, and R. Li, "Data prediction model in wireless sensor networks based on bidirectional LSTM," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–12, Dec. 2019.
- [20] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2014.
- [21] U. Raza, A. Camera, A. L. Murphy, T. Palpanas, and G. P. Picco, "Practical data prediction for real-world wireless sensor networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 8, pp. 2231–2244, Aug. 2015.
- [22] P. Ganjewar, S. Barani, and S. J. Wagh, "A hierarchical fractional LMS prediction method for data reduction in a wireless sensor network," *Ad Hoc Netw.*, vol. 87, no. 1, pp. 113–127, 2019.
- [23] H. Jiang, S. Jin, and C. Wang, "Prediction or not? An energy-efficient framework for clustering-based data collection in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 6, pp. 1064–1071, Jun. 2011.
- [24] D. P. Kumar, A. Tarachand, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Inf. Fusion*, vol. 49, pp. 1–25, Sep. 2019.

- [25] M. Aboubakar, I. Quenel, and A. A. A. Ari, "A predictive model for power consumption estimation using machine learning," in *Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET)*, Cape Town, South Africa, Dec. 2021, pp. 1–5.
- [26] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.
- [27] X. Dai, R. Fu, Y. Lin, L. Li, and F.-Y. Wang, "DeepTrend: A deep hierarchical neural network for traffic flow prediction," 2017, *arXiv:1707.03213*.
- [28] R. Amutha, G. G. Sivasankari, and K. R. Venugopal, "A prediction model for effective data aggregation materials and fault node classification in WSN," *Mater. Today, Proc.*, vol. 49, pp. 2962–2967, Jan. 2022.
- [29] Y.-F. Zhang, P. J. Thorburn, W. Xiang, and P. Fitch, "SSIM—A deep learning approach for recovering missing time series sensor data," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6618–6628, Aug. 2019.
- [30] P. R. Chandre, P. N. Mahalle, and G. R. Shinde, "Deep learning and machine learning techniques for intrusion detection and prevention in wireless sensor networks: Comparative study and performance analysis," in *Design Frameworks for Wireless Networks*. Cham, Switzerland: Springer, 2020, pp. 95–120.
- [31] S. N. Mohanty, E. L. Lydia, M. Elhoseny, M. M. G. Al Otaibi, and K. Shankar, "Deep learning with LSTM based distributed data mining model for energy efficient wireless sensor networks," *Phys. Commun.*, vol. 40, Jun. 2020, Art. no. 101097.
- [32] R. Huang, L. Ma, G. Zhai, J. He, X. Chu, and H. Yan, "Resilient routing mechanism for wireless sensor networks with deep learning link reliability prediction," *IEEE Access*, vol. 8, pp. 64857–64872, 2020.
- [33] W. Feng, N. Guan, Y. Li, X. Zhang, and Z. Luo, "Audio visual speech recognition with multimodal recurrent neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 681–688.
- [34] Intel. (2004). *Intel Berkeley lab Sensor Data*. Accessed: May 15, 2021. [Online]. Available: <http://db.csail.mit.edu/labdata/labdata.html>
- [35] X. Liu, "A deployment strategy for multiple types of requirements in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2364–2376, Oct. 2015.
- [36] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.
- [37] E. M. Ferragut, J. Laska, and R. A. Bridges, "A new, principled approach to anomaly detection," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, vol. 2, Dec. 2012, pp. 210–215.
- [38] M. Tsikerdekis, S. Waldron, and A. Emanuelson, "Network anomaly detection using exponential random graph models and autoregressive moving average," *IEEE Access*, vol. 9, pp. 134530–134542, 2021.
- [39] *Spearman Rank Correlation Coefficient*. New York, NY, USA: Springer, 2008, pp. 502–505.
- [40] A. A. A. Ari, N. Labraoui, B. O. Yenke, and A. Gueroui, "Clustering algorithm for wireless sensor networks: The honeybee swarms nest-sites selection process based approach," *Int. J. Sensor Netw.*, vol. 27, no. 1, pp. 1–13, 2018.
- [41] M. Srinivas and T. Amgoth, "EE-hHHSS: Energy-efficient wireless sensor network with mobile sink strategy using hybrid Harris hawk-salp swarm optimization algorithm," *Int. J. Commun. Syst.*, vol. 33, p. e4569, Aug. 2020.
- [42] S. R. Ahmed, M. A. Kadhim, and T. Abdulkarim, "Wireless sensor networks improvement using leach algorithm," *IOP Conf. Mater. Sci. Eng.*, vol. 518, no. 5, 2019, Art. no. 052023.



AROUNA NDAM NJOYA received the Master of Engineering (M.Eng.) degree in telecommunications from the National Advanced School of Engineering, University of Yaoundé I, Cameroon, in 2009, and the Master of Science (M.Sc.) degree in computer engineering and the Ph.D. degree in computer science both from the Faculty of Sciences, University of Ngaoundéré, Cameroon, in 2012 and 2018, respectively. He is currently a Lecturer at the Department of Computer Engineering, University Institute of Technology, University of Ngaoundéré. He has served as a reviewer for various journals and international conferences. His research interests include combinatorial optimization, evolutionary computation, big data, the IoT, and coverage maintaining in WSNs.



ALLASSAN A NKEN TCHANGMENA received the master's degree in mathematical science option data science from the African Institute for Mathematical Sciences, in 2021, and the Diploma degree in robotic and control system engineering from the Polytechnic Higher School of Douala. He is currently enrolled in a master's program with the African Master's in Machine Intelligence (AMMI), Senegal, in 2022. In addition, he is also the CEO and a Co-Founder of Wedoo.ai, a research laboratory in artificial intelligence and its application.



ADO ADAMOU ABBA ARI received the Bachelor of Science (B.Sc.) degree in mathematics and computer science, the Master of Science (M.Sc.) degree in computer engineering, and the master's degree in business administration from the University of Ngaoundéré, Cameroon, in 2010, 2012, and 2013, respectively, and the Ph.D. degree (Hons.) in computer science from Université Paris-Saclay, France, in 2016. He is an Associate Researcher at the DAVID Laboratory, Université Paris-Saclay, University of Versailles Saint-Quentin-en-Yvelines, France, and an Associate Professor at the LaRI Laboratory, University of Maroua, Cameroon. His current research interests include wireless networks, security/privacy in the IoT, and artificial intelligence applications in communication networks. He served/serving on several journals and conferences program and reviewing committees. Thus, he achieved the outstanding reviewer status with the Elsevier *Computer Networks*. Moreover, he has Recognized Reviewer of a number of journals including *Journal of Network and Computer Applications*, *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, *IEEE ACCESS*, *Information Sciences*, *Computer Communications*, *Remote Sensing*, *Sensors*, *Electronics*, *Telecommunication Systems*, *Wireless Personal Communications*, *Sustainable Computing: Informatics and Systems*, and *Journal of Ambient Intelligence and Humanized Computing*.



ABDELHAK (MOURAD) GUEROUI received the M.Sc. degree from Université Paris 6 and the Ph.D. degree in networking and computer engineering from the University of Versailles Saint-Quentin-en-Yvelines, France, in 2000. He is currently an Associate Professor of computer engineering at the University of Versailles Saint-Quentin-en-Yvelines. He has authored more than 100 publications in international conferences and journals, as well as book chapters, including ACM, IEEE, and Elsevier, and has both chaired and served in numerous program committees in prestigious international conferences. His research interests include wireless networks (WATM, WIMAX, LTE, 5G, cloud computing, WLAN, MESH, VANET, and WSNs), particularly performance evaluation and QoS provisioning. He received several awards, including best papers and serves/served on several journals and conferences executive committees.



CHRISTOPHER THRON received the dual Ph.D. degrees in mathematics and physics from the University of Wisconsin, and the University of Kentucky, respectively. He is currently an Associate Professor at the Department of Science and Mathematics, Texas A&M University–Central Texas. Formerly, he was a Systems Engineer with NEC America, Motorola, and Freescale. He has more than 40 journal publications, three books, and nine patents. His research interests include machine learning, operations research, stochastic optimization, agent-based modeling, and algorithm design applications, sensor networks, signal processing, target tracking, scheduling, epidemiological and social modeling, public health statistics, and foundations of quantum mechanics. He has participated extensively in collaborative research in Africa, and has been supported by the U.S. Fulbright program, International Mathematician's Union, and Air Force Research Laboratory.



BERTHINE NYUNGA MPINDA the bachelor's degree in applied mathematics from the University of Kinshasa, DRC, the joint master's program in mathematical science from Stellenbosch University and the African Institute for Mathematical Sciences (AIMS–South Africa), and the master's degree in machine learning from the African Masters of Machine Intelligence (AMMI), Ghana. She is currently a Teaching Assistant at the African Institute for Mathematical Sciences,

AIMS, Cameroon, and an Organizer of the Women in Machine Learning and Data Science Kinshasa Chapter. She is a recipient of several scholarships and grants including the Mastercard Foundation, Google and Facebook, ESMT Berlin, Black in AI, and Women in Machine Learning.



EMMANUEL TONYE was born in Etouha, Cameroon, in 1952. He graduated in micro-electronics. He received the Doctorat d'Etat of science from the National Polytechnic Institute of Toulouse. He is a professor. He teaches at the Department of Electrical and Telecommunication Engineering, National Advanced School of Engineering, University of Yaoundé I. He is the author of several scientific papers and has already supervised several master's and Ph.D. researches.

His current research interests include communication system, the IoT, signal and image processing, and e-learning.

...



OUSMANE THIARE received the Diploma of Advanced Studies (D.E.A.) degree in computer science from the Polytechnic School of the François Rabelais University of Tours (Polytech'Tours), the D.E.A. degree in applied mathematics from the Gaston Berger University of Saint-Louis, Senegal, in 2000, and the Ph.D. degree in computer science from the University of Cergy-Pontoise, France, in 2007. He has been a Full Professor at the Department of Computer Science, Gaston Berger University of Saint-Louis, since 2015.

From 2013 to 2018, he served as the Director at the Ousmane Seck Computer Center, Gaston Berger University of Saint-Louis. He has also been the Director at the African Centre of Excellence for Mathematics, Informatics and Information and Communication Technologies (ACE-MITIC), since 2017. ACE-MITIC is an ambitious World Bank project with a budget of eight million U.S. dollars. He was appointed, in May 2018, a Rector at the Gaston Berger University of Saint Louis, the President of the University's Assembly. His research interests include distributed systems, distributed resource management, computing grids, cloud computing, mobile ad hoc networks, wireless sensor networks, and the Internet of Things (IoT).