Conserving energy through neural prediction of sensed data

Conserving Energy Through Neural Prediction of Sensed Data

Siamak Aram*, Ikramullah Khosa, and Eros Pasero
Politecnico di Torino, Turin, Italy
{siamak.aram, ikramullah.khosa, eros.pasero}@polito.it

Abstract

The constraint of energy consumption is a serious problem in wireless sensor networks (WSNs). In this regard, many solutions for this problem have been proposed in recent years. In one line of research, scholars suggest data driven approaches to help conserve energy by reducing the amount of required communication in the network. This paper is an attempt in this area and proposes that sensors be powered on intermittently. A neural network will then simulate sensors' data during their idle periods. The success of this method relies heavily on a high correlation between the points making a time series of sensed data. To demonstrate the effectiveness of the idea, we conduct a number of experiments. In doing so, we train a NAR network against various datasets of sensed humidity and temperature in different environments. By testing on actual data, it is shown that the predictions by the device greatly obviate the need for sensed data during sensors' idle periods and save over 65 percent of energy.

Keywords: Wireless sensor networks, Neural networks, Data prediction, Power Consumption

1 Introduction

Wireless sensor networks are based on multilayered structure of interactive sensor nodes. Cooperation of the nodes enables performing numerous operations such as event detection, target tracking, environment sensing, security and elder people monitoring [1–8]. Scholars are currently struggling to increase short lifetime of wireless sensor networks [4, 5, 9–14].

The problem, in fact, is rooted in limited energy resources available to sensors. Thus, energy consumption should be efficient at all layers of sensors' operation. To this end, it is advisable to reduce power consumption at each and every level of system routines, network protocols, data processing, and even hardware modules. In particular, there exist data driven approaches that propose data compression, data prediction, and in-network processing [15–17].

Today, time series analysis is applied to a wide range of sciences including biology, physics, economy and technology. Technically, time series is represented as an ordered set of vectors which are determined as per the formula given below [18]:

$$y(t), t = 0, 1, 2, ... \tag{1}$$

Practical aspects of time series forecasting can be found in a variety of related scientific articles and publications. Any time Series forecast requires preliminary choice of the prediction strategy. This choice should be made with due regard to the end objectives of the time series prediction, i.e. facilitating production and activities. So far, time series method has proven useful for many areas. Specifically, time series

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 6, number: 1, pp. 74-97

*Corresponding author: Corso Duca degli Abruzzi, 24, 10129 Torino, Politecnico di Torino – DET Department – Third Floor – Neuronica Lab. (NEURONICS AND POWER ELECTRONICS LABORATORY), Tel: +39 011 090 4065, Web: http://www.neuronica.polito.it/

forecasts are often required to solve problems in medical, econometric and engineering field. Accuracy of predictions are often impeded by the chaotic behavior of time series. Hence the need to determine the exact state of the analyzed domain in the beginning of experiment. Time series forecasts are mainly conducted relying on Autoregressive (AR), Autoregressive Moving Average (ARMA) and Moving Average (MA) models. Nonetheless, neither of these linear models are suitable for application to non-linear signals. Assuming that, the scientists found an alternative to linear module predictions. It was established that Artificial Neural Networks (ANN) could assist predicting time series with better accuracy. As a computational structure, ANN incorporates models which are based on biological patterns. Another option is NN which exploits its non-linear constituent elements in order to select the most accurate data hypothesis. These integral elements are joint by links with weights containing all data formulated in the course of forecasting. NN's exceptional capacity for aggravation and approximation operations gives it an advantage over other networks in terms of predictions' accuracy. Consequently, manufacturers have already devised a number of neural network-based tools for generating statistics and modeling. Owing to the properties of NN referred to above, neural network approaches have recently become highly applicable for time series forecasting. Accordingly, there are two major points which should be taken into consideration while performing neural network based data readings [18]: a) intervals of data sampling and b) sequence of points where sample data will be collected.

It is supposed these matters demand empirical solution. For the aims of this paper, we facilitated selecting appropriate intervals and data sampling points by creating a new algorithm. The aforementioned algorithm was developed upon comparison and analysis of data collected from a number of sources and in variable circumstances. The optimized algorithm proposed by us was devised with regard to the principle of lowering power consumption. This goal was achieved on account of reducing the volumes of data involved in the sampling process.

In this paper, we concentrate on data prediction to conserve energy in wireless sensor networks. To do so, sensed data is thought of as making a time series [19] where there may be a correlation between the points. This fact serves as the rationale behind our method and implies that sensors could be powered down in judiciously chosen time intervals. The correlation among the points would then allow predicting of sensed data during sensors' idle periods. To bring functionality to this idea, we make use of neural networks from NAR model [20].

In accordance with the proposed method, prediction of data was performed via non-linear autoregressive network. Further on, this method was affirmed by conducting a series of empirical experiments conducted within the frame of the research at hand. Sets of data collected specially for the experimental part of the research included humidity and temperature parameters from real sources. Performance of neural networks in differentiated circumstances was observed and assessed by us. In order to meet the objectives of the study the neural network was fed by two delayed targets. As a result, the size of the network's hidden layer was gradually altered. The best prediction outcome was achieved upon feeding the network 20 hidden layers' neurons. In the parallel series of experiments the quantity of data inputs was varied. Notwithstanding the increase in the input, the error rate had no sufficient decrease. Therefore, it has been proven experimentally that the method tested during this study allows decreasing energy costs in WSNs. The results show that the proposed method is very effective and saves over 65 percent of the energy while preserving the qualitative characteristics of the test data. Further on, our method was empirically through the experiments conducted within the frame of this research. The datasets employed contain actual sensed humidity and temperature in different environments.

In our experiments, we do not employ a round robin scheduler to schedule sample acquisition in sensors. Instead, a central control unit schedules sample acquisition in a nondeterministic manner. Thus, it is not required to perform any supportive transmissions within the interval between any two sample acquisitions in a sensor. It is also shown that there is no need for extensive knowledge of the deployment domain so that optimizing the neural network's parameters can merely be accomplished on the basis of residual power volumes and sleep state periods. This approach is elaborated on in Section 3. Experimental results are explained in Section 4. Section 5 discusses the method and the results obtained from experiments. Section 6 concludes the paper.

2 Data Prediction approaches

Data readings in WSNs are often impeded by insufficient energy supply of the network. However, the actual data which could be acquired from the nodes may be substituted by the predicted data, which is the aggregate of the readings from one or several sensors. In [15] and [21] the authors analyzed three basic approaches to data prediction, namely algorithmic, stochastic and time series forecasting methods. For the purpose of our study, these three methods have been summarized in Figure 1 and discussion pertinent to it.

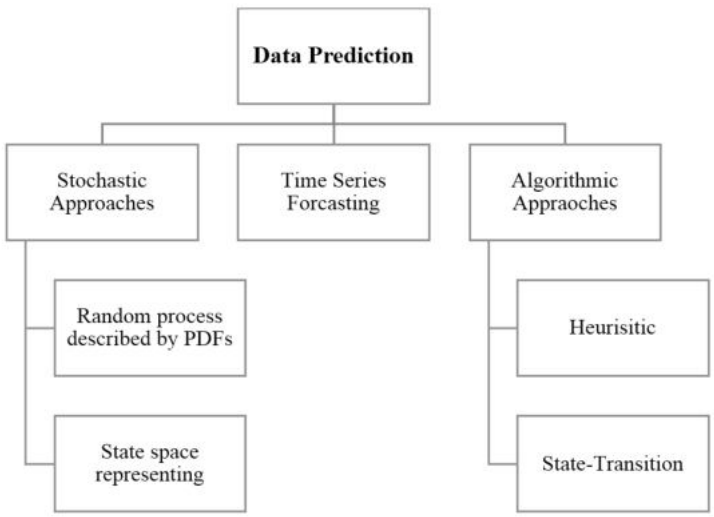

Figure 1: A detailed taxonomy of Data Prediction schemes

The goal of data forecasting method is to substitute any real data with a model which applies predictions to respond to a set of relevant queries. For the purposes of testing the validity of a model, the data is first subjected to regular sampling by sensor nodes and then compared against the forecast. Consequently, validity is considered to be affirmed when the data prediction falls within the extent of set thresholds and/or tolerances. In the contrary case, the sensor node will update the model in question or use actual data sampled before. The structure of the model and its working principles usually define which forecasting method would be compatible with it.

2.1 Stochastic Approaches

For the first time, the description of method for data forecasting applicable for WSNs was described in [22] . The aforementioned technique involved a so-called ”probabilistic model”. In greater details, the framework developed by the authors allowed wireless systems to exploit correlation-aware probabilistic models while processing the queries. Upon incorporation of the model, it was no longer needed for the system to make any direct connection to the network itself. Accordingly, the quantity of data transmissions was sufficiently reduced. Though, the aim of the work referred to above was not confined merely to reducing the data transfers but to find a means to cut down the quantity of required data samplings.

Probabilistic model relies on the forms which apply stochastic characterization methods to statistical properties and probabilities of the system. Basic techniques applied in this respect are as follows:

1) State space representation 2) Random processing.

Firstly, in the course of performing state space representation non-predictable components (noises) are eliminated, which, in turn, enables predicting further coming samples. It is possible to randomize the data, thus fitting it into the probability density function (herein - "pdf"). PDFs generated this way may serve as base for data prediction [23]. However, preliminary combining of PDFs with the samples acquired earlier is obligatory in such case. Secondly, labeling noises as unpredictable components and excluding them from the transmission helps obtaining state space representation for the chosen phenomenon. The approach addressed above may be illustrated by the Ken solution cited in [23]. This solution is targeted at reconsideration of the technique used to process basic tasks on data collection, such as, for example, "SELECT" queries for collecting data and detection of anomalies. Following the contemporary practices (such as BBQ [22]), Ken achieves desirable accuracy by relying on probabilistic models. The key asset of Ken's solution is its capability to ensure compliance of the predicted data with the real value determined by sampling. At the same time, Ken helps shrinking the volumes of data subjected to transferring. During the prediction each model involved in the data transfer is duplicated two times: first time when it leaves the source, second time when it enters the sink. Following this scheme, it becomes possible to acquire PDFs which correlates with specified attributes. Should the probabilistic base model lose its validity, it will be automatically upgraded by the corresponding node. As soon as the upgrade of the model is effected, the sink will receive new samples required for further updates. It should be emphasized that models built with regard to temporal and special correlations may be used during the training of Ken's method with the same result as in case of using the ordinary models. This also refers to the models devised to deal with the peculiarities of particular phenomena.

2.2 Time Series forecasting

Time series prediction is the second data prediction method which will be considered within the frame of the research at hand. During the time series prediction most credible values of future transactions are generated by analyzing values acquired from previous data samplings. In contrast both to probabilistic and statistical methods, within duration of time series prediction only the internal data structuring is being processed. The forecasting involves the following successive steps: 1) An error is randomly selected and compared against the established pattern 2) The pattern is defined as regards its inherent features, i.e. fluctuation, periodicity etc. 3) Generating prediction model is generated based on accomplished characterization of the pattern.

Then it becomes possible to predict future values by analyzing the generated model. So far, time series forecasting proves to be most compatible with non-complex basic models, such as, for example, auto-regressive, moving average or combined techniques. In theory, the aforementioned models may be substituted by more contemporary and more complicated solutions, for example, GARCH and ARIMA [24]. However, in the case with WSNs more lightweight technique is preferred, since high intricacy of the models threatens the stability of entire systems. [25] refers to PAQ which makes predictions of future values dependent on the in-built autoregressive models, attributable to every sensor. During the transmission, any immediate communication of the models with sensors is set aside. Instead, models are processed by a sink node and predicted values are formulated. The sink is regularly updated to keep up with any new developments on the models or acquisition of data on external readings. Upon implementing this method the monitoring of sensors becomes more straightforward on account of dropping mostly unnecessary communications. Also, error-bound rate of forecasted data remains within control of WSN's users. The prediction itself starts at the learning phase when values acquired previously are used for generating appropriate model. Meanwhile, the sampled data is queued with the aid of the corresponding nodes. As soon as the queue is completed the model can be generated and transferred further to the sink. In order for the model to be regarded as feasible the values obtained via it should not exceed the acceptable rate of errors. In the opposite case, the system may follow the scenarios: a) Defining outliers among the data sets and excluding them from the reading (marking the samples), b) Singling out invalid models and forwarding the latter for recalculation (marking model). It should be mentioned that the model is no longer valid when a sufficiently high quantity of readings performed within a series overlaps the error threshold. Therefore, as soon as the update is completed, the model is directed back to the sink.

Further on, in [26] the author overview another type of time series models known as Similarity-based Adaptive Framework (SAF). SAF represents the combination of AR and a time-varying function. SAF encompasses benefits that would be enlisted below: 1) It is efficient in performing value predictions for the sensors which evaluate environmental parameters like humidity, temperature and others, 2) It has low operational cost, 3) It is compatible with contemporary WSNs. Unlike PAQ, SAF is not devised for performing repeated readings for the purposes of increasing precision of the prediction. Instead, SAF seeks to prevent the involvement of highly noisy data and outliers in the readings. Moreover, implementation of SAF allows forecasting the values disregarding abnormalities of their variations. This is accomplished by way of including the trend component into the volume of data under sampling. This feature of SAF contributes in the accuracy of performed prediction and extends the scope of detections to discrepant data. There is a risk for the data to become inconsistent should any complications impede the sensors' calculation of models. Assuming that data degrade actually happens, the node will be commanded to initiate the scenario for restoring the stability of the model.

2.3 Algorithmic Approaches

The following chapter on comparing methods for data forecasting is dedicated to heuristic models. Heuristic models are also referred to as ”state transition models”. The task attributed to heuristic models lies in selecting correct techniques for devising novel models or inputting updates on characterization into currently valid models. There also exist some alternative models which can be tailored to technical requirements of WSNs. [27] addresses one of the alternative solutions mentioned earlier. In particular, the authors refer to Energy Efficient Data Collection (EEDC) mechanisms. It serves as an example of a behavioral model. The role of EEDC may be described as conducting source-initiated updates. In the course of source-initiated update, real value of sensed data is compared against the upper and lower node bounds. The precision of performed readings is confirmed by calculating differences between these two bounds. Later on, the sink distributes upper and lower bounds between the sensors which altogether form the network. Further data acquisition envisages matching bounds to the acquired samples. If the anticipated precision is not met, the sink would be immediately updated.

Besides, reduction of power consumption is also possible to achieve upon compressing the sensed data. In [28] scientists provide overview for PREMON, which aims to observe different kinds of correlations typical for the readings performed by spatially proximate sensors. Respectively, temporal, spatial and spatial-temporal correlations may occur. PREMON method relies on the same principles as adopted in compressing the size of videos. In other words, MPEG technique embraces on wireless sensor Networks' behavior from the moment when sink receives their first readings. Here, the role of the sink is to build a prediction model based on correlative properties. The sensors receive the model as soon as it is devised. Similarly to other techniques discussed in this chapter, MPEG presupposes comparison of the real values and predictions formulated by the model. If the discrepancy of the actual data and predicted values is insignificant, there is no necessity to communicate the real value to the sink. Pursuing the objective to increase the accuracy of predictions, model is occasionally annulled and new models are created based on more recent data samples.

2.4 Comparison

Among the aforementioned options, the stochastic approach is the most holistic and integrated solution. Feasibility of the said method is widely acknowledged. Moreover, such technique offers new opportunities for conducting data aggregation and other related high-level tasks. At the same time, the major disadvantage of stochastic approach and/or similar methods is their excessive consumption of energy. Our observations reveal that the stochastic framework is especially feasible when applied to many sensors. In this respect, stochastic methods do not suit the purposes of the present study, which primarily concentrates on low-power sensor networks. Yet, the computational cost of the stochastic framework can be reduced with the aid of a distributed model which retains robustness of network without extra energy losses. Referring to Algorithmic methods it should be highlighted that such methods cannot be analyzed in the cumulate. There is no general concept of an algorithm. On the contrary, each algorithm serves a certain purpose, the peculiarities of which should be considered in development process. Specific features of these algorithms, as said earlier, are mainly revealed during their practical application. With regard to that, it is advisable to analyze each algorithm separately. Hence, Time series predictions most effectively meet the particular goals of the study in question. Simple time series prediction usually runs at a moderate energy cost. In this regard, it has been proven experimentally that time series forecasting may be performed on low power networks without compromising the accuracy of predicted data. In the same vein, up-to-date technical solutions do not involve the whole amount of data in the sensing processes before enabling a compatible model. As a result, sufficient computational reduction is achieved. It should be also apparent that the extent of possible reduction is directly proportional to the volumes of stationary data subject to sensing. Upon comparison of corresponding approaches, it appears that Time Series predictions are most prevalent in the WSNs realm. During past ten years the scientists who are working in this field have aptly compared the techniques designed for time series predictions demonstrated in Table 1 and discussed in [21].

Table 1: Time series samples of data prediction

Predicting Method	Samples	Description
Dual Prediction Scheme(DPS) or prediction approach based on, Kalman Filter	[29–34]	Agreement between node and sink with threshold
Least Mean Square (LMS)	[30–32]	No Agreement between node and sink – No prior knowledge
Moving Average or Autoregressive based models (AR, ARMA and ARIMA)	[21], [35], [36]	Sink and sensors exchanging data and performing prediction on both sides.
A hybrid model based on Grey-Model-based and Kalman Filter	[37]	-
Proportional Integral Derivative (PID)	[38]	-
“Send on delta”	[39]	Calculates the difference between the current value and the predicted value.
Mean square error (MSE)	[31], [32]	Calculates the difference between the current value and the predicted value.
Root mean square error (RMSE)	[36], [40]	Calculates the difference between the current value and the predicted value. / Ratio reduction/RMSE.

Today's WSNs are capable of processing complex algorithms but the highly variable data should be avoided for the sake of accuracy. In the meantime, WSN data prediction is usually performed via models based on time series forecasting instruments, such as MA, ARMA [35] and GM(1,1) [41]. Additionally, stochastic approaches have proved to be more effective in the cases when the probability density model is tried on the data in laboratory conditions. Meanwhile, it is still impossible to disregard the risk of computational overhead of the applied algorithms approaches. In view of the foregoing, this paper aims to propose the time series based method of data reduction essential for decreasing energy consumption during sensors' communication. This objective is met by putting the network through meticulous examination prior to selecting the appropriate time to commence interrogation of certain sensors.

3 Method

The efficient time series prediction of the sensor's output is needed to achieve the goal of minimization of power consumption by the sensor. The lower the communication, the less power will be consumed. As discussed earlier, there are different kinds of time series prediction methods depending on the applicable parameters and the practical usage. We have implemented a Nonlinear Autoregressive model for prediction. This model is used for prediction of an output at time t by using the subsequent outputs as shown in Figure 2.

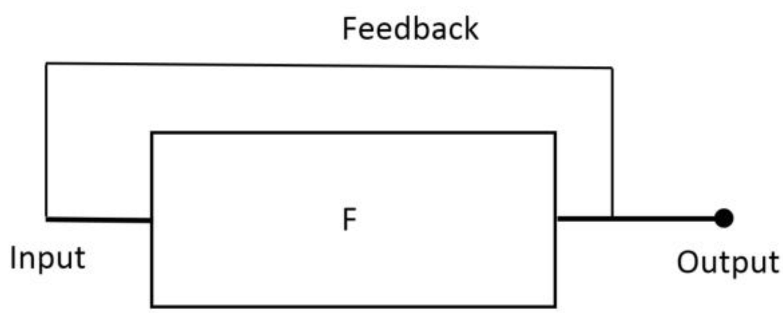

Figure 2: Nonlinear Autoregressive model

In other words, the model provides the current target output by using target values at previous time stamps. Practically, we used an Artificial Neural Network to make it learn from real values and then predict for unknown input. An ANN is a network composed of large number of inter-connected units called neurons. An ANN architecture may have one or more hidden layers, but typically one hidden layer is sufficient to map any kind of linear as well as nonlinear approximation as shown in Figure 3 [42].

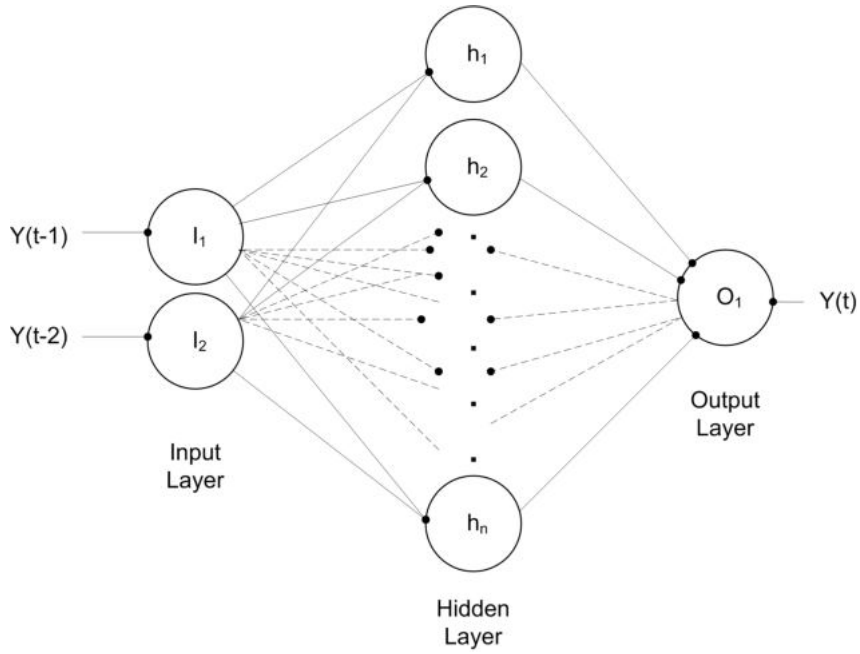

Figure 3: A three layer artificial neural network architecture

Estimation of optimized number of neurons in the hidden layer is a vital task. Higher number of hidden layer neurons may result as overfitting due to over parameterization. On the contrary, a small number of hidden neurons may become insufficient to fit the data. Hence our approach employs an ANN to learn from samples, while it is fed with inputs by following the NAR model where subsequent target values are fed as inputs to predict the current one. The approach is used to prepare an efficient neural network which after training may perform efficient time series prediction. Since the method makes use of previous outputs to predict the current one, it is believed that sampling can be minimized if higher prediction accuracy is achieved. Sampling or communication interval with the sensor can be reduced to save power. Alternatively, the idle time is increased, and meanwhile the sensor's reading is predicted by the trained neural network. This idle time can be regulated according to the prediction error provided by the neural network. More power can be saved by increasing the idle time if prediction error is lower. Alternatively in case to higher error, frequent communication will be required, to maintain the accuracy.

3.1 Algorithm for efficient sampling

Our proposed solution's artificial neural network architecture is shown in Figure 4; The dataset contain sensor's values for temperature as well as humidity. We used these sensor's values as targets and corresponding time stamps as inputs. The temperature values vector $[T_1 T_2, \ldots, T_n]$ against n time-steps are fed as input to the network. Initially, we employed the NAR model with two feedbacks. Hence the network has two input neurons where two delayed feedbacks are provided for each of the target values. Concretely, we introduced two (feedback) delays in the input layer to store the previous two values: T_{j-1} and T_{j-2} for the prediction of target value T_j at the jth time stamp. Therefore the network uses the temperature values at two delayed timestamps to predict the current value (see equation 1) [42]. Learning of the neural network plays an important role in achieving optimum results. Back propagation is a classic and widely used learning algorithm in neural networks. We used the Levenberg-Marquardt (LM) algorithm [43] as the training algorithm of the classifier. This is a sophisticated form of gradient descent back-propagation algorithm which performs nonlinear least square minimization. The mathematical details of the LM algorithm are included in [44]. Parameters for network training are summarized in Table 2.

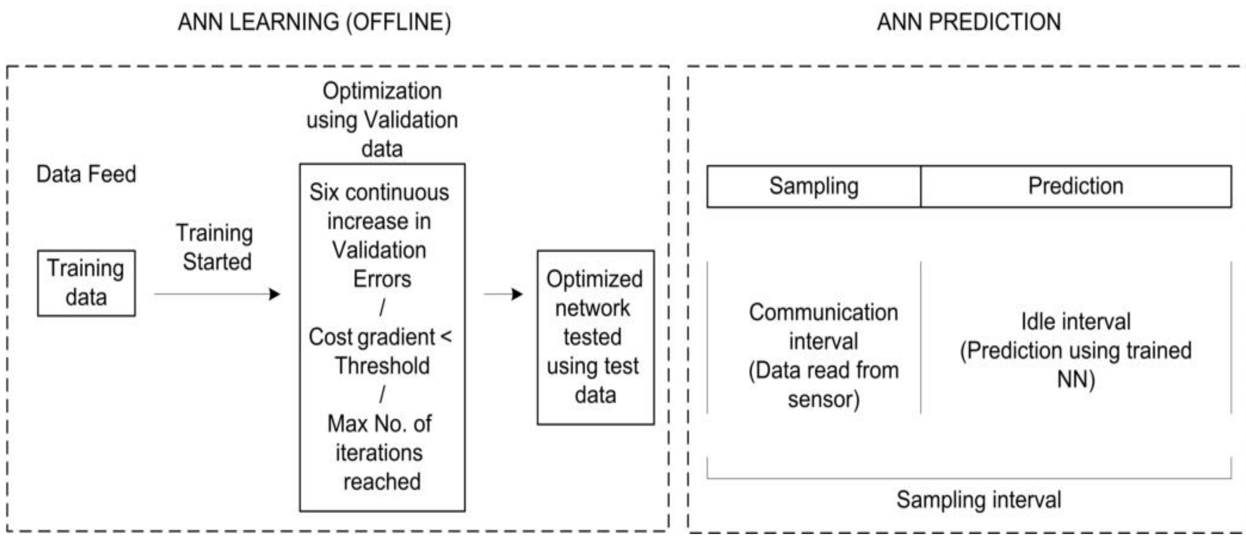

Figure 4: Block Diagram of overall Methodology

$$y(t) = F[y(t-1), y(t-2), ..., y(t-d)] \tag{2}$$

Table 2: Network's training parameters

Parameter	Value
Minimum gradient threshold	$1e^{-10}$
Initial learning rate (λ)	0.01
Increasing ratio of λ	10
Decreasing ratio of λ	0.1
Maximum value for λ	$1e^{8}$

For each set of temperature as well as humidity sensor, the data is divided as follows: 70% for network training, 15% for cross validation, and the rest 15% for test purpose. The network is set to be trained using the training data, and simultaneously to be optimized based on cross validation outcome. In every iteration regularized cost for the training data is calculated as:

$$J(\beta) = [-\frac{1}{m}\sum_{i=1}^{m} y^i log(P(x^i)) + (1-y^i) log(1-(P(x^i))] + \frac{\lambda}{2m}\sum_{j=1}^{n} {\beta_j}^2 \tag{3}$$

Where x^i represents the input feature vector for ith sample, y^i represents target value of ith sample, λ is the regularization parameter, set as 0.01, and β_j represents the weight parameter for jth sample. $P(x^i)$ represents the sigmoidal output for jth sample and is calculated as

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{n} \beta_i x_i)}} \tag{4}$$

During network training, the cost (equation 2) is reduced after every iteration. Weight parameters are optimized after each iteration according to the LM algorithm. For the hidden layer, sigmoid activation function (see Equation 3) is used whereas linear function is applied at output neuron. Training is set to be stopped if either there are six consecutive increases in validation error or the gradient becomes less than the selected threshold (see Table 2). After training, the trained network with optimized weights is used to calculate the test data results.

Algorithms (1 and 2) for network training for efficient sampling are defined as follows:

Algorithm 1 Neural Network Training

```
Input: training data for n time stamps at input layer
Return sum
  while Number of iterations = Max iterations do
     Calculate cost J(β), as in (3)
     Calculate gradient of J(β),
     Compute the validation data errors,
     Compute the no. of iterations c, if validation error continues
     increasing
     if c == 6 then
        Stop training
     else if gradient ¡ Min threshold defined then
        Stop training
     end if
  end while
   Test the network by using test data
```

Algorithm 2 Efficient sampling from sensor

```
Define: Max number of time stamps
  while time stamps = Max time stamps do
     Define no. of time stamps for sampling (opted equal to 2)
     Define sampling interval (8 for Temp sensor)
     if sampling interval ends then
        Read data from sensor
        if sampling sampling time ends then
           Start prediction
        end if
     end if
  end while
```

3.2 Data test bed

The algorithm was verified based on two types of data. Likewise, the proposed solution was analyzed under different conditions of datasets, i.e. in the chamber and in the natural environment of the Neuronica Laboratory of Politecnico di Torino.

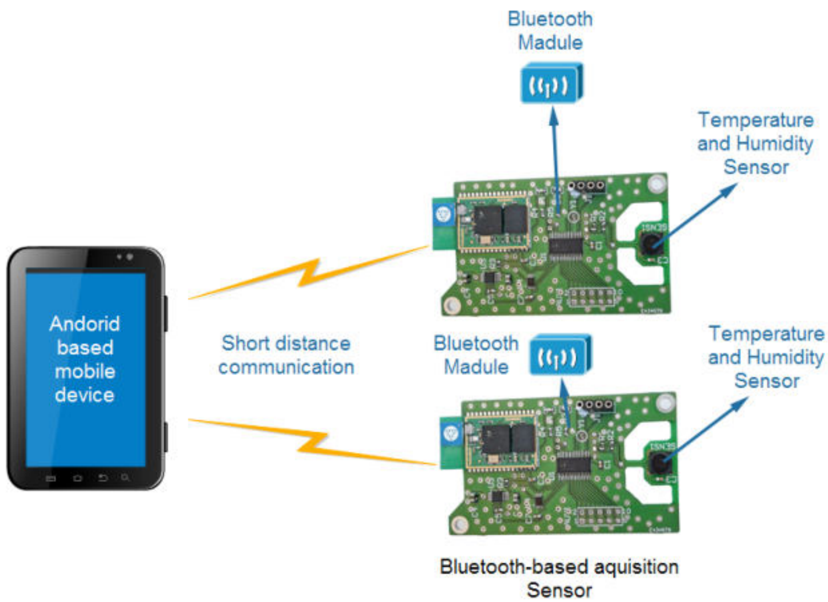

Figure 5: The Bluetooth-based acquisition system

3.2.1 First Dataset

In [5], the choice of Texas Instruments MSP430F2132 is determined by its low power consumption. The sensor Sensirion SHT21 was selected for the same reasons [4]. This Bluetooth-based sensor incorporates 3V lithium battery (CR2247). In the course of the experiment the environmental data was obtained via three Bluetooth-based temperature and humidity acquisition systems and the experiment was conducted in the controlled environment, in particular, in a climatic chamber with temperature range for climatic test from -40°C to +180°C and Angelantoni Challenge 250. Initial environment of the chamber was established as follows: a relative humidity of 50%; temperature of 25°C. These circumstances were maintained for the period of 10 min. Then the temperature was decreased until -20°C. Its gradient was set as -0.5°C per minute. The lowered temperature was preserved within the chamber for about 10 minutes and then brought back to 25°C. Then gradient was estimated as 0.5°C per minute. At the final stage of the experiment the stable temperature of 25°C had been supported within chamber for 10 minutes. The sampling frequency of the wireless sensor network was set to a rate of one sample per minute. The interval between two corresponding measurements was set to 60-second schedule.

3.2.2 Second Dataset

The second consequent sampling was performed to assess the temperature and humidity parameters via Bluetooth-based tool in real conditions [11] (Figure 5). Sampling of the datasets was performed in the laboratory, where carrier was used to move the sensors between the warm and cold sources. The data reading at every source point lasted for about 2-3 minutes. In accordance with the scientific requirements, the laboratory sources used in the experiment could vary their own temperature by 5C and humidity by %10. Data readings were performed every 15s. The length of entire experiment was one-hour. The stationary conditions of the experiment can be potentially reproduced in any ordinary environment.

4 Result

We have 2 data sets obtained from temperature as well as humidity sensors placed in the indoor environment covering a time span more or less of two hours with 96 samples. In addition, we have another dataset composed of temperature values for 260 samples, recorded from the sensor placed in an environmental chamber for 4.5 hours. This data can be assumed as noise free simulated data for the temperature. We applied the sensor's values as targets against time to the network and analyzed the time series prediction response.

As we discussed earlier number of hidden layer neurons plays an important role in achieving the optimized network accuracy. Few neurons in the hidden layer may underfit the data, while large number of hidden neuron may lead to overfitting. Hence to estimate a good choice for hidden layer neurons, we varied this number and analyzed the network performance. For each of the sensor's dataset, we varied the number of hidden neurons as 5, 10, 20 and30, and observed the performance with respect to mean squared error (MSE). We introduced two feedback delays (two delayed values were fed to the network) to predict the current value. The prediction response for the humidity sensor, data set 1, showing target and the predicted outcomes for each timestamp is shown in the Figure 6. The error for each quantized timestamp is also plotted at the bottom of each response (recorded with variable number of hidden layer neurons). Similarly, the prediction response for humidity sensor, data set 2 is also recorded, shown in Figure 7.

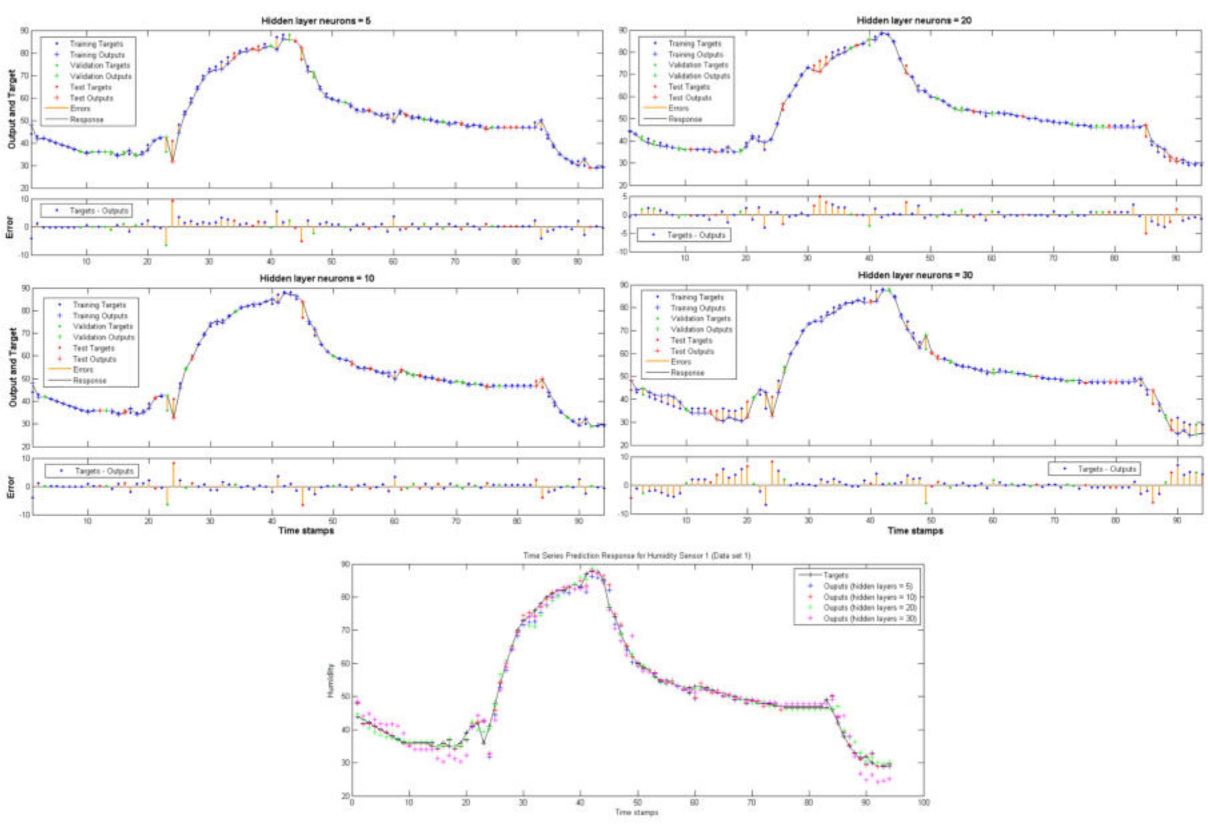

Figure 6: Time series prediction response of the neural network with the error plots for Humidity Sensor 1, Data 1 (Training data = 70%, Validation and Test = 15% each) by varying the size of hidden layer of the network

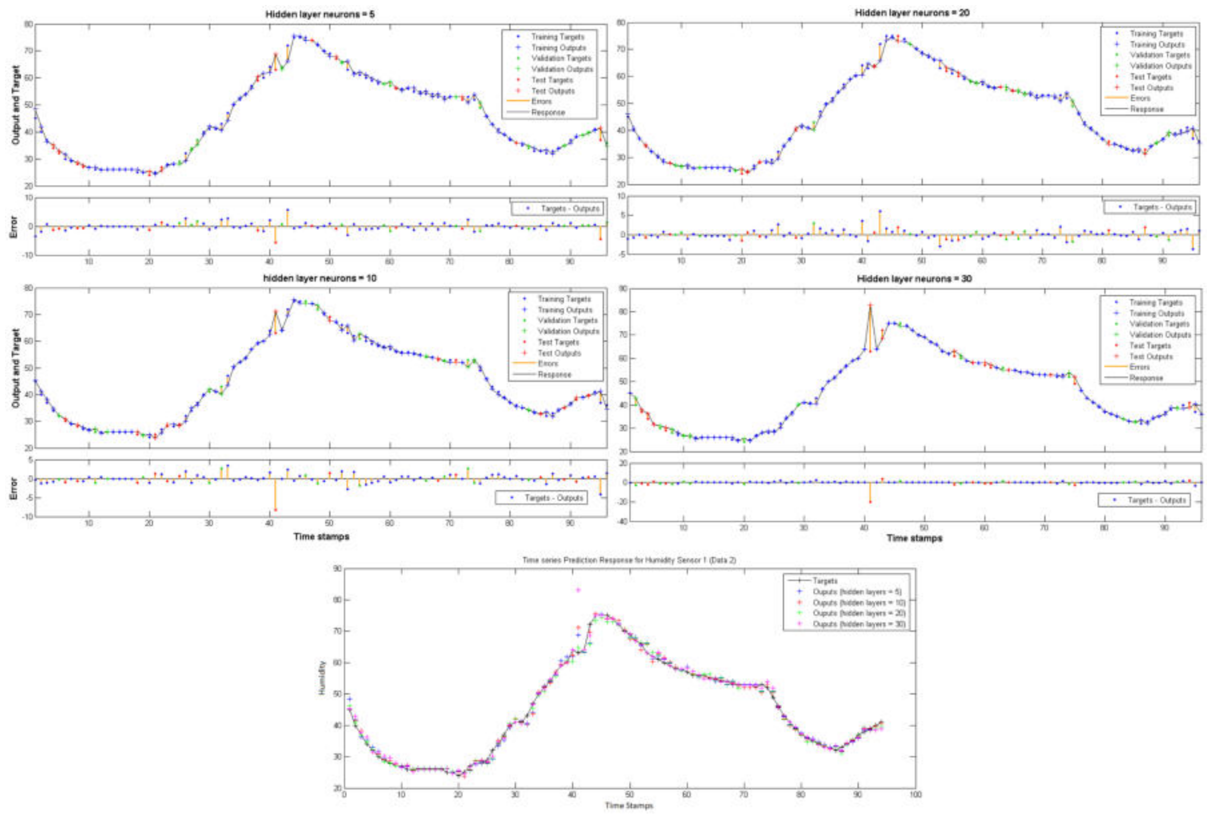

Figure 7: Time series prediction response of the neural network with the error plots for Humidity Sensor 1, Data 2 (Training data = 70%, Validation and Test = 15% each) by varying the size of hidden layer of the network

It can be noticed that the samples belonging to training, validation and test data are inconsistent in different results. This is due to random division of data before calculation of results. We recorded the network prediction response for one sensor's data. Then, we randomized the data and divided it again into: training, validation and test data, and calculated the results again. The process has been repeated five times, and an average outcome was calculated. Hence the presented results reflect the average response of the network. We used the data sets obtained from both the sensors and recorded the network performance. Figure 8 and Figure 9 represent the network prediction response for the temperature sensor data: set 1 and data set 2 respectively. The response for the data acquired from temperature sensor placed inside environment chamber is shown in Figure 10. The mean squared error (MSE) for each of the sensor's data is plotted against variable number of hidden layer neurons, presented in Figure 11. We recorded the network prediction response for each of the sensor's datasets. If we consider the network performance with a particular architecture, it can be seen that the network response is almost similar for each of sensor's datasets. The network shows good prediction even with fewer numbers of hidden neurons (like in case of 5). With the increase in hidden layer size (selecting 10 hidden neurons), the accuracy of the network is further increased. However, the maximum accuracy (corresponding to minimum MSE) is achieved with 20 hidden layer neurons among the selected choices for hidden layer size, for each of the datasets. Figure 11 (MSE plot) demonstrates continuous error reduction to 20 hidden neurons. Later by selecting 30 hidden neurons, the mean squared error rate started to increase.

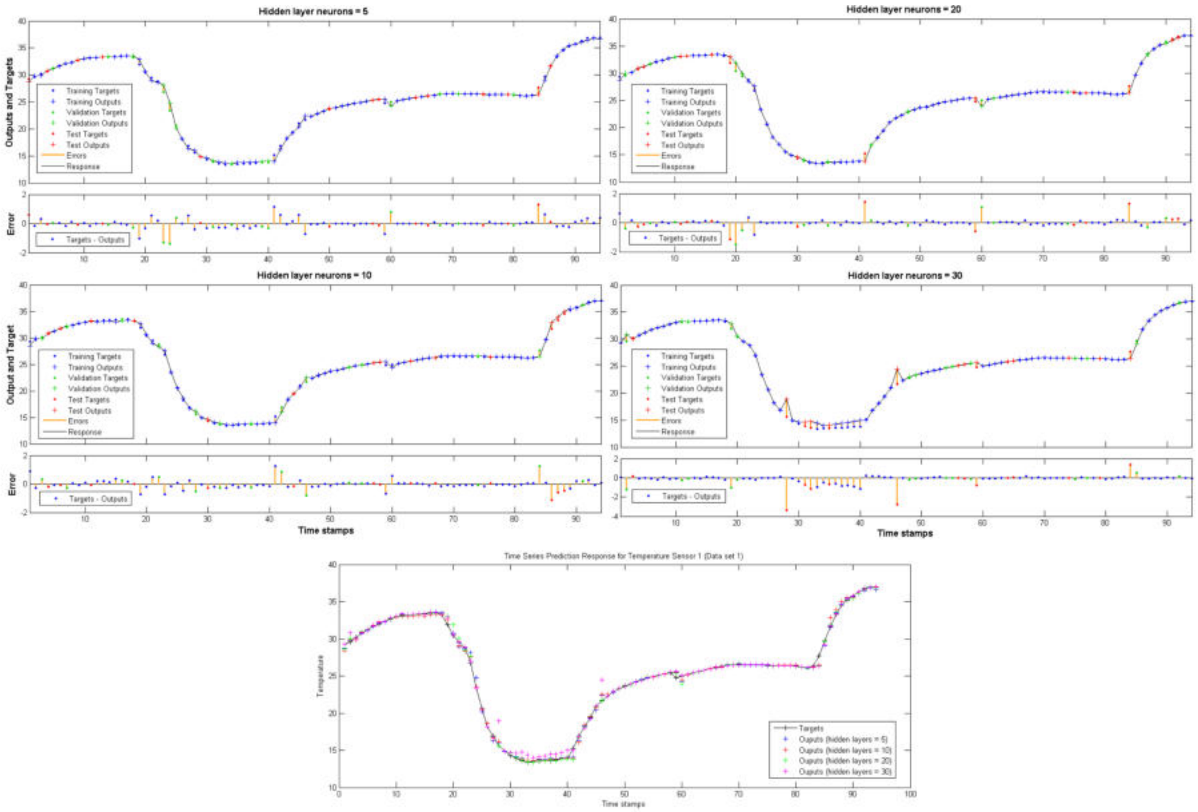

Figure 8: Time series prediction response of the NN with the error plots for Temperature Sensor 1, Data 1 (Training data 70%, Validation and Test 15% each) by varying the size of hidden layer of the network

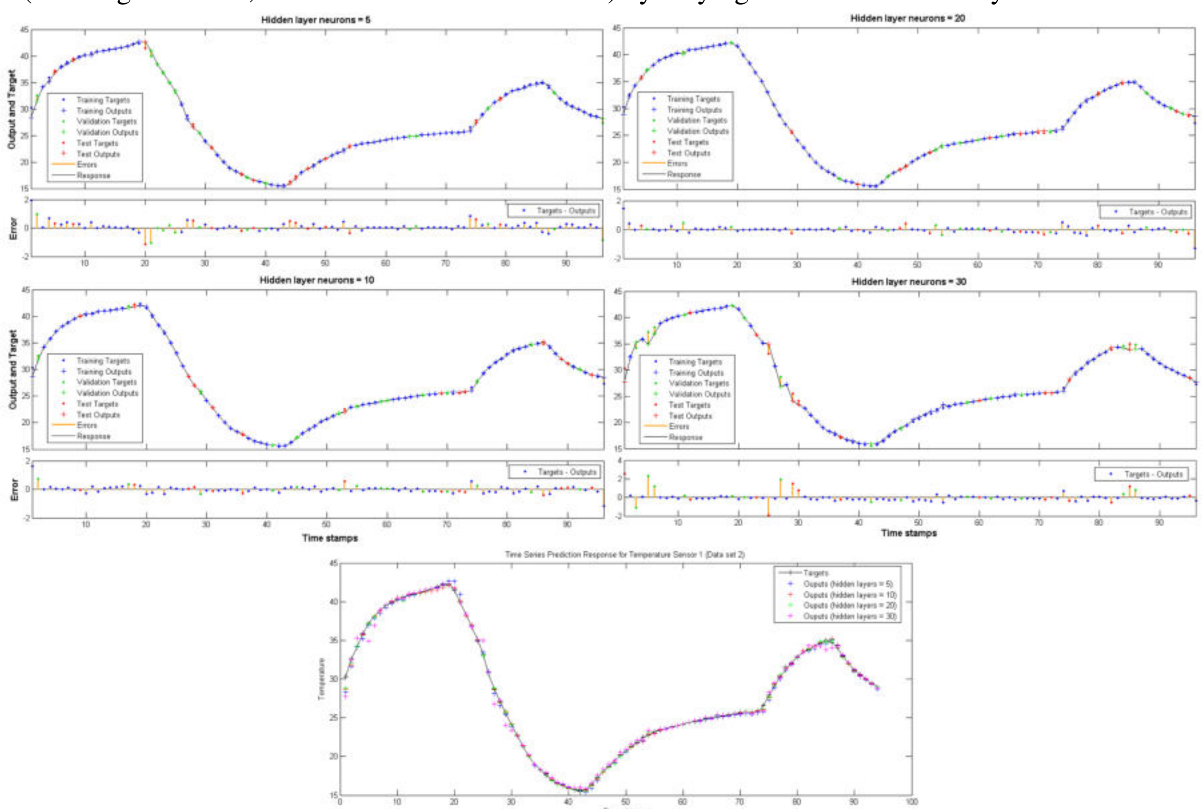

Figure 9: Time series prediction response of the NN with the error plots for the Temperature Sensor 1, Data 2 (Training data 70%, Validation, Test 15% each) by varying the size of hidden layer of the network

The problem of overfitting occurred here due to over-parameterization. This overfitting response can be traced in all the sensor's datasets with the choice of 30 hidden layer neurons.On the basis of aforementioned results it can be concluded among the selected choices, 20 is acceptable quantity for hidden neurons. This network architecture estimation was carried out by using two previous outcomes in a NAR system. We wondered what could happen if we changed the number of inputs (delayed outcomes). Thusly, we kept the hidden layer neurons equal to 20 (as estimated), and varied the number of inputs to the network. We varied the number of inputs (subsequent output delays) as 1, 2, 3 and 4. Further, we analyzed the network performance on the basis of variable number of input features (which are delayed outputs, see Equation 1). We calculated the results in the same manner as in the previous section. We recorded the network prediction responses for variable number of inputs, by keeping the hidden neurons equal to 20. The (average) response of the network is recorded for each of the sensor's data and comparative results are presented by using different number of inputs to the network. Figure 12 and 13 demonstrate the comparative results for humidity sensor for data set 1 and data set 2 respectively. Comparative results for temperature sensor data set 1 and 2 are presented in Figure 14 and 15 respectively. Figure 16 shows the results for temperature sensor of environmental chamber. Evidently, network prediction response has improved for all sensor's datasets by feeding more information to the network. The best response is given by the network when 4 delayed targets are being used as input to the network i.e. more information, better learning and higher accuracy. We calculated the MSE for each of the sensor's data against variable number of network inputs presented in Figure 17.

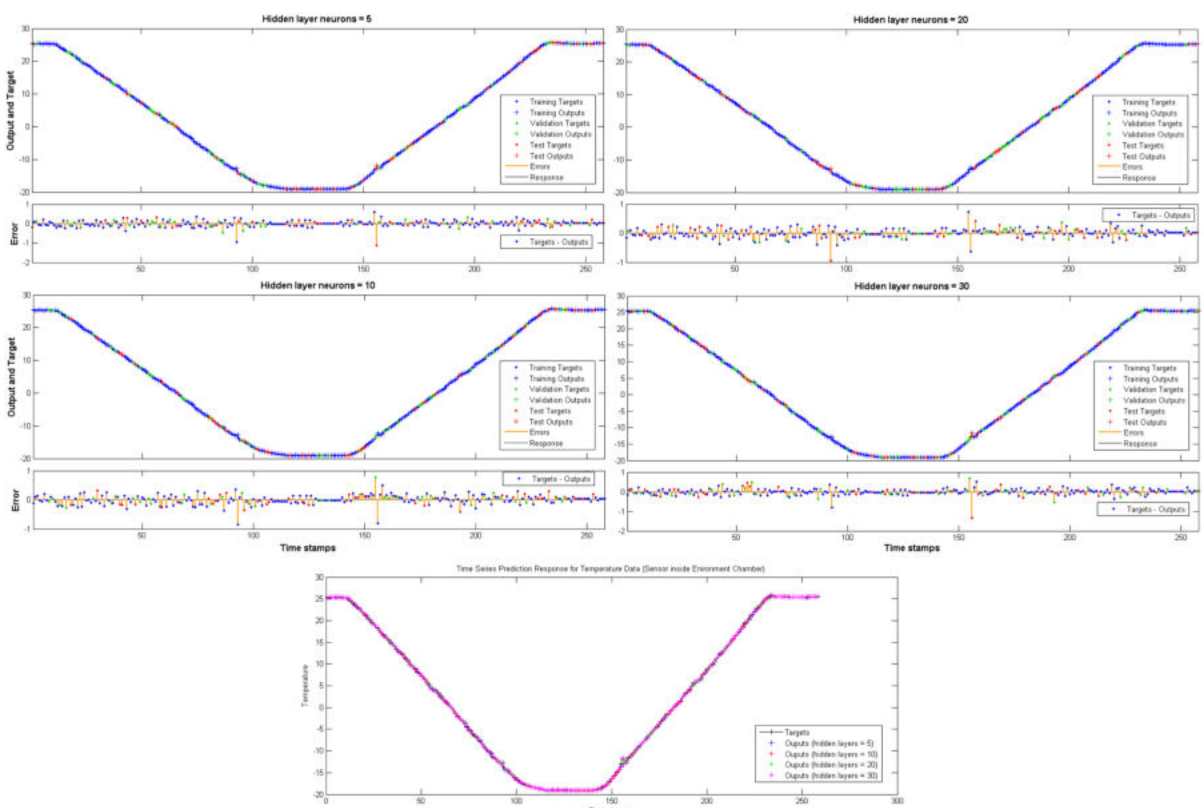

Figure 10: Time series prediction response of the neural network with the error plots for the Temperature Sensor inside Environment Chamber (Training data 70%, Validation and Test 15% each) by varying the size of hidden layer of the network

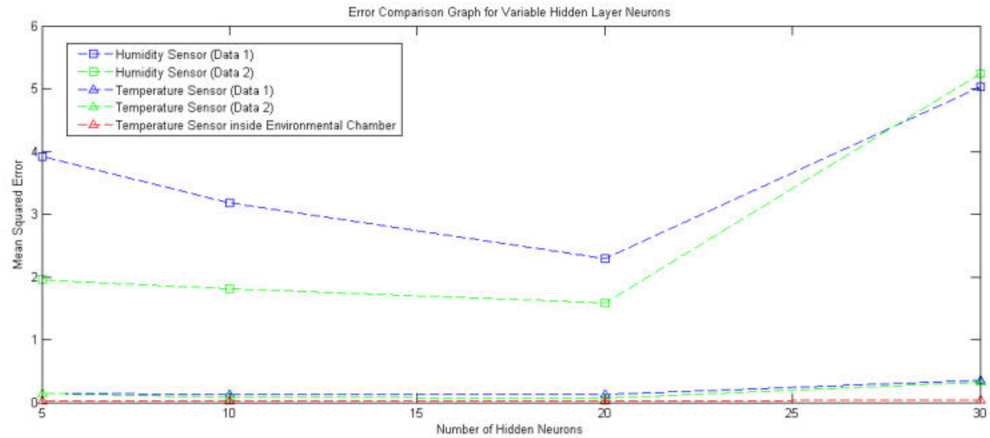

Figure 11: Mean Squared Error plot for each of the sensor's data against different number of hidden layer neurons used in the network

In Figure 17, it can be observed that the error is reducing continuously with the increase in number of inputs (by feeding more information). A large gradient in the error can be observed upon changing the number of inputs from 1 to 2. However, there is a slight reduction in error with further increase in number of inputs. Therefore by keeping the number of inputs more than 2 did not significantly improve the accuracy. This typical behavior can be observed for all the datasets. Hence it can be concluded that choice of number of inputs can be made equal to 2 since the error is not significantly reduced beyond this number (see Figure 17). On the other hand the network showed higher error for humidity sensor's data due to its wider range. The temperature data range is comparatively smaller than that of humidity. Hence for the same number of samples, the network performed better for temperature data. After estimating a choice for number of hidden layer neurons earlier equal to 20, we are now interested in finding the best trade off between the number of hidden layer neurons and the number of inputs to the network which lead to an optimum solution.

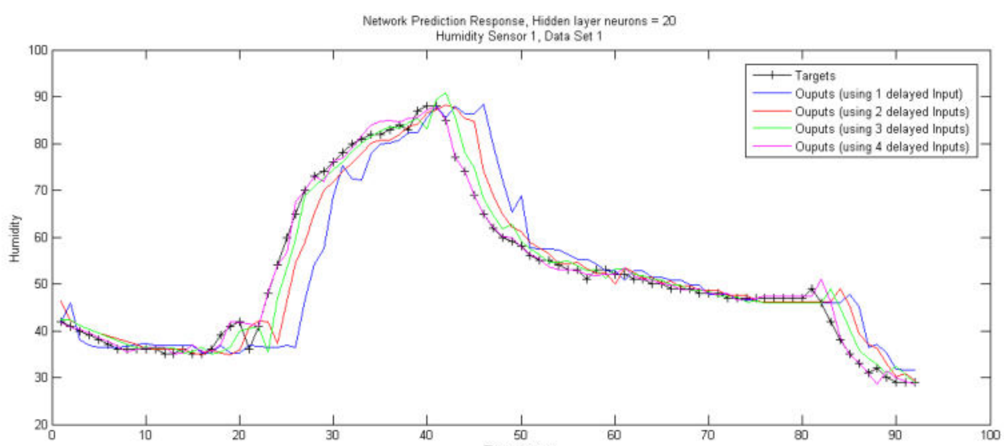

Figure 12: Network prediction response with 20 hidden neurons by varying the number of inputs, Humidity sensor 1, dataset 1

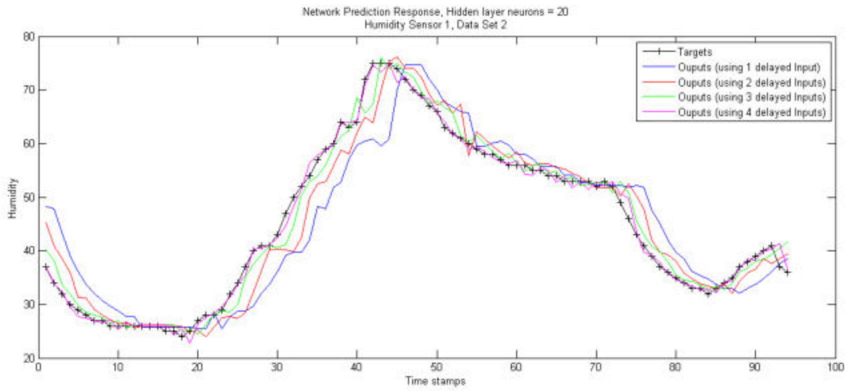

Figure 13: Network prediction response with 20 hidden neurons by varying the number of inputs, Humidity sensor 1, dataset 2

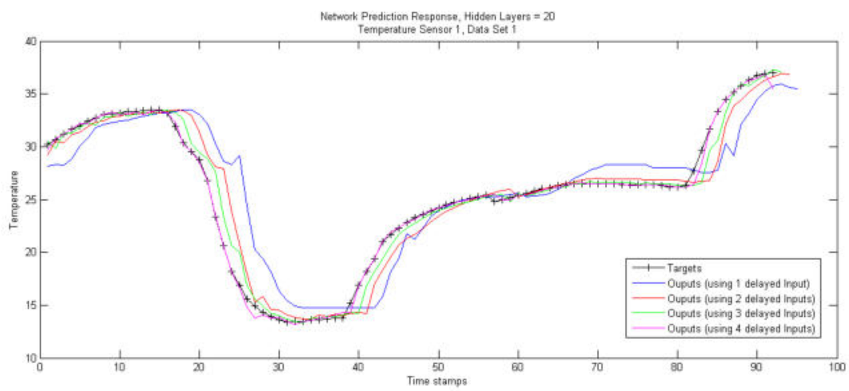

Figure 14: Network prediction response with 20 hidden neurons by varying the number of inputs, Temperature sensor 1, dataset 1

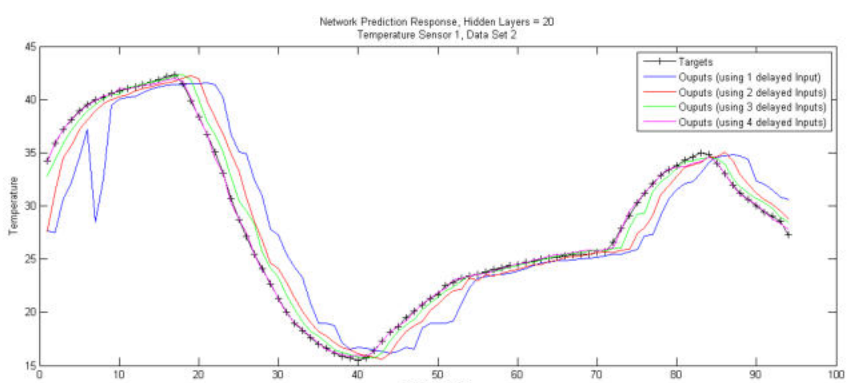

Figure 15: Network prediction response with 20 hidden neurons by varying the number of inputs, Temperature sensor 1, dataset 2

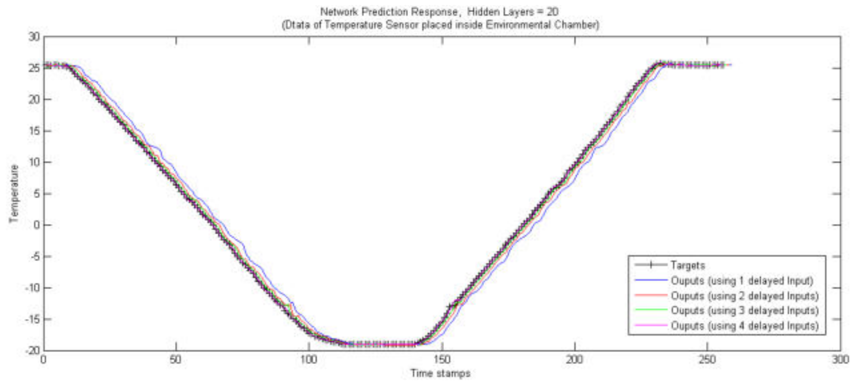

Figure 16: Network prediction response with 20 hidden neurons by varying the number of inputs, Temperature sensor kept inside environmental chamber.

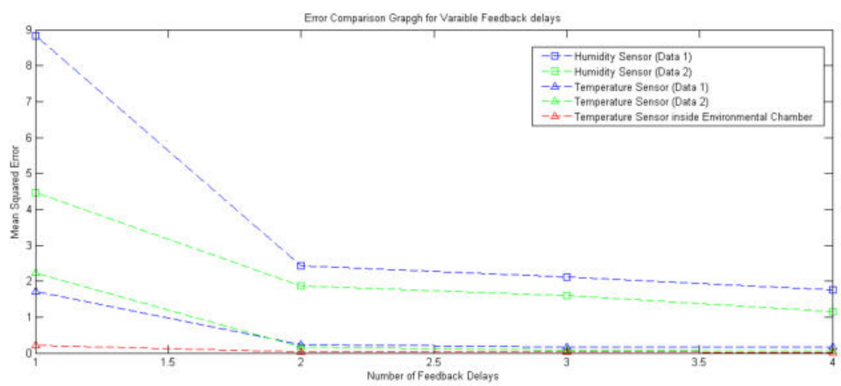

Figure 17: Mean Squared Error plot for each of the sensor's data against different number delayed outcomes used as input to the network

To estimate the existing margin of error in the network setup, we calculated the Mean Absolute Percentage Error (MAPE). It provides an estimate of an average unsigned percentage error exist in the network, and is useful to estimate the error with the tolerable margin. We calculated the MAPE as;

$$MAPE = \frac{1}{n}\sum_{k=1}^{n}\left|\frac{T_k - P_k}{T_k}\right| * 100 \qquad (5)$$

Where T_K represents the target value for timestamp k, and P_k represents the corresponding predicted value, and $k = 1, 2, 3, \ldots, n$ for n number of samples.

The MAPE plot against variable number of network inputs for all data sets is presented in Figure 18. We can see that this percentage error plot response is similar to the error plot in Figure 16. The minimum percentage error was recorded when 4 inputs fed to the network. Once again it can be observed that reduction in MAPE by choosing more than 2 inputs is very small. Thus, restricting the number of inputs to 2 with 20 neurons in the hidden layer seems to be a favorable solution.

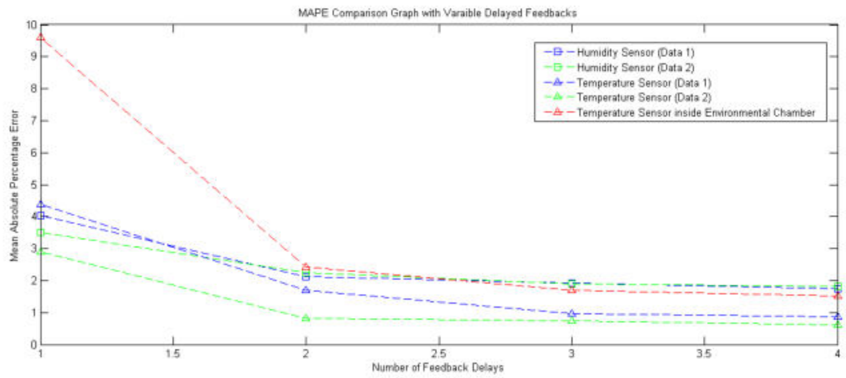

Figure 18: MAPE plot for each of the sensor's data against different number of network inputs

We also calculated the error margin by reducing the size of the network. Figure 19 shows MAPE calculated by choosing different sizes of hidden layer of the network. It is noted that although the error is higher with fewer hidden neuron (5 neurons), however the error gradient by switching from 5 to 20 remains significantly low. With this results we can make a general conclusion that in case of larger tolerable error margin, even a network with few hidden neurons is choosable.

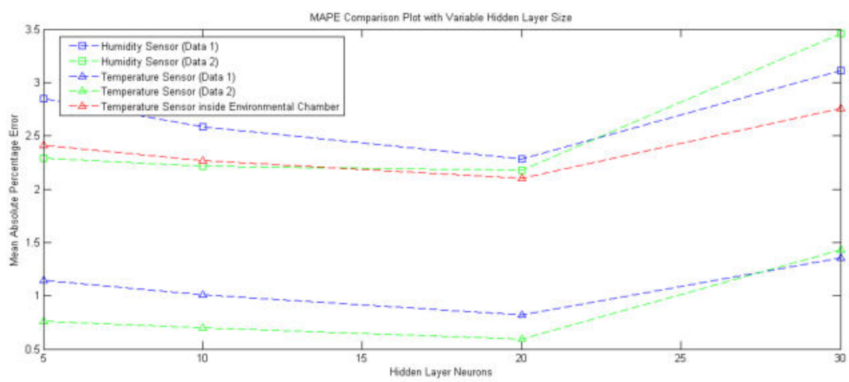

Figure 19: MAPE plot for each of the sensor's data against different number of hidden layer neurons used in the network

5 Discussion

We used the neural network for time series prediction of the sensor's data. For this purpose, Non-linear Auto Aggressive (NAR) network was chosen. For information, NAR network performs time series prediction by using the target values at subsequent delayed time stamps as inputs, and predicts the value at the current time stamp.

We recorded the actual data from the sensors for humidity and temperature and used this data with the network for prediction. A couple of datasets with 96 samples were derived from each of the sensors in a time span of 2 hours. In addition, we obtained 260 samples from another temperature sensor, placed in an environmental chamber. We altered the temperature inside the chamber and recorded the sensor's values by communication with the sensor. Initially, we analyzed the performance (based on mean squared error) with the aforementioned data to estimate a network architecture with optimum choice of hidden layer neurons. The network performance was better for temperature sensor data, set 1 and set 2, which correspond to indoor environment readings and range between 15C and 42C (Δ = 27C). There was no significant improvement recorded in accuracy with the increase in hidden layer size. The network showed good performance with fewer hidden layer neurons. On the contrary, the error recorded for humidity sensor data (ranges 20-90, Δ = 70) was higher. The range of humidity data is quiet larger than that of temperature with the same number of samples. Consequently for humidity sensor, the network performed better with larger hidden layer size. The environmental chamber's sensor data contains large number of samples with linear change in temperature, so the network outperformed for this data even with the smallest network size.

Later, the data was used to analyze the network performance by altering the size of input data fed to the network. To this extent, the NAR network predicts the time series by using the target values at subsequent delayed timestamps. While keeping the hidden layer size equal to 20, the network was fed with 1, 2, 3 and 4 subsequent delayed targets alternatively, and the performance was recorded. Continuous reduction in the error was observed as a result, however it was not significantly reduced by using more than two inputs.

With 20 hidden neurons and feeding 2 delayed target values as input, we calculated the Mean Absolute Percentage Error (MAPE) to estimate the error margin projected by the network. The network showed the MAPE up to 1.6% for the indoor temperature sensor and 2.2% for the indoor humidity sensor overall. At the same time, the average error margin for the sensor in the chamber was recorded as 2.4%. It can be thus concluded that the network provides lower error disregarding its small architecture. The reduction in percentage error by increasing the hidden layer size is less than 1% for humidity sensor, and even lesser for temperature sensor. Hence, keeping the number of inputs equal to 2 with 20 hidden neurons produces the optimized results. However, the network can be arranged with five hidden neuron by compromising 1% of error. This can be adopted to improve computational efficiency in the case where the large error margin is allowed.

Regarding the power saving, it is obvious that less communication with the sensor corresponds to more power saving. By selecting the optimum network architecture with 20 hidden neurons, test set data prediction is carried out for every sample by using two delayed feedbacks as input. Hence to predict a target value (of temperature or humidity) with the aforementioned accuracy, the network requires target values at previous 2 timestamps. In this way 66.6% is the communication time to get 2 samples (in which sensor is communicating), while 33.3% is the prediction time (for which target is predicted). Then for each next sample, target can be predicted by using last 2 values (by feedback). This time series prediction can be continued as far as error remains within threshold. Since the gradient of temperature in an indoor environment is small, it is possible to minimize communication time with the sensor and maximize prediction time (by the network) to save more power. By fixing the communication period equal to 8 time stamps (supposing tn as nth time stamp), the communication time is 25% (2/8*100) at t1 and t2. The idle time is 75% (6/8*100) where prediction is carried out at t3-t8 by feeding back subsequent outputs. For t3, the prediction error is the lowest (as mentioned) since sensor's readings are used for prediction, whereas for t8, the error will be the highest. This is due to predicted outcomes are being used as inputs to predict. For the humidity sensor where there is a large range of observations, the idle time can be reduced to maintain the accuracy. Therefore, we can conclude that for the temperature sensors, up to 75% of power can be saved with an error margin of 2.6% (1.6% by network + 1% of the sensor) for single prediction. Similarly for the humidity sensor, 66% of power can be saved for predicting once in 6 timestamps, with an error margin of 3.2% (2.2% by network + 1% of the sensor). For the other temperature sensor inside the environmental chamber, 75% of power can be saved within an error margin of 3.6% (2.6% by network + 1% of the sensor).

6 Conclusion

We devise a data driven approach to reduce power consumption in wireless sensor networks. The method is based on the prediction of sensed data using non-linear autoregressive neural networks. Evaluation is also performed using the actual data obtained from temperature and humidity sensors. The performance of the network is assessed under different conditions. In fact, we feed two delayed targets to the network and change the size of the hidden layer. The results imply that the most accurate forecast is obtained with 20 hidden layers. Another observation pertains to the number of inputs. The results show that increasing the number of inputs does not lead to a significant decline in the error rate.

The experiments conducted in this research indicate that our method substantially reduces power consumption in wireless sensor networks. Implementing the proposed method in real-life sensor networks will help prevent unnecessary sampling and, in turn, will reduce energy and costs. There is still much to be done. More theoretical work on the proposed method is required. Likewise, characterizing the environments for which the method leads to satisfactory results deserve future research.

References

[1] R. I. da Silva, V. Del Duca Almeida, A. M. Poersch, and J. M. S. Nogueira, "Wireless sensor network for disaster management," in *Proc. of the 2010 IEEE/IFIP Network Operations and Management Symposium (NOMS'10), Osaka, Japan*. IEEE, April 2010, pp. 870–873.

[2] D. Wu, S. Ci, H. Luo, Y. Ye, and H. Wang, "Video surveillance over wireless sensor and actuator networks using active cameras," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2467–2472, 2011.

[3] S. Kim and E. Paulos, "Inair: sharing indoor air quality measurements and visualizations," in *Proc. of the 2010 SIGCHI Conference on Human Factors in Computing Systems (CHI'10), Atlanta, Georgia, USA*. ACM, April 2010, pp. 1861–1870.

[4] S. Aram, A. Troiano, and E. Pasero, "Environment sensing using smartphone," in *Proc. of the 2012 IEEE Sensors Applications Symposium (SAS'12), Brescia, Italy*. IEEE, February 2012, pp. 1–4.

[5] S. Aram, A. Troiano, F. Rugiano, and E. Pasero, "Low power and bluetooth-based wireless sensor network for environmental sensing using smartphones," in *Proc. of the AIAI 2012 International Workshops: AIAB, AIeIA, CISE, COPA, IIVC, ISQL, MHDW, and WADTMB, Part II, Halkidiki, Greece, IFIP Artificial Intelligence Applications and Innovations*, vol. 382. Springer Berlin Heidelberg, September 2012, pp. 332–340.

[6] J. Goldman, K. Shilton, J. Burke, D. Estrin, M. Hansen, N. Ramanathan, S. Reddy, V. Samanta, M. Srivastava, and R. West, "Participatory sensing: A citizen-powered approach to illuminating the patterns that shape our world," Foresight & Governance Project, White Paper, May 2009. [Online]. Available: http://www.mobilizingcs.org/wp-content/uploads/Participatory_Sensing.pdf

[7] A. Chehri, P. Fortier, and P.-M. Tardif, "Security monitoring using wireless sensor networks," in *Proc. of the 5th Annual Conference on Communication Networks and Services Research (CNSR'07), Frederlcton, New Brunswick, Canada*. IEEE, May 2007, pp. 13–17.

[8] R. Bogue, "Environmental sensing: strategies, technologies and applications," *Sensor Review*, vol. 28, no. 4, pp. 275–282, 2008.

[9] L. Mesin, S. Aram, and E. Pasero, "A neural data-driven algorithm for smart sampling in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 23, pp. 1–8, February 2014.

[10] M. Cordina and C. J. Debono, "Maximizing the lifetime of wireless sensor networks through intelligent clustering and data reduction techniques," in *Proc. of the 2009 IEEE conference on Wireless Communications & Networking Conference (WCNC'09), Budapest, Hungary*. IEEE, April 2009, pp. 1–6.

[11] S. Aram, L. Mesin, and E. Pasero, "Improving lifetime in wireless sensor networks using neural data prediction," in *Proc. of the 2014 World Symposium on Computer Applications & Research (WSCAR'14), Sousse, Tunisia*. IEEE, January 2014, pp. 1–3.

[12] L. Mesin, S. Aram, and E. Pasero, "A neural data-driven approach to increase wireless sensor networks' lifetime," in *Proc. of the 2014 World Symposium on Computer Applications & Research (WSCAR'14), Sousse, Tunisia*. IEEE, January 2014, pp. 1–3.

[13] A. Abu-Baker, H. Huang, and S. Misra, "Maximizing lifetime sequences of wireless sensor networks powered by renewable energy," *Sensor Review*, vol. 34, no. 4, pp. 374–381, 2014.

[14] T. Enokido, A. Aikebaier, and M. Takizawa, "Computation and transmission rate based algorithm for reducing the total power consumption," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 2, no. 2, pp. 1–18, 2011.

[15] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, May 2009.

[16] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *Proc. of the 2005 International Conference on the Information Technology: Coding and Computing (ITCC'05), Las Vegas, Nevada, USA*, vol. 2. IEEE, April 2005, pp. 8–13.

[17] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer communications*, vol. 30, no. 14, pp. 2826–2841, October 2007.

[18] E. Diaconescu, "The use of narx neural networks to predict chaotic time series," *WSEAS Transactions on Computer Research*, vol. 3, no. 3, pp. 182–191, March 2008.

[19] H. Kantz and T. Schreiber, *Nonlinear time series analysis*, 2nd ed. Cambridge university press, 2004.

[20] T. Chow and C. Leung, "Nonlinear autoregressive integrated neural network model for short-term load forecasting," *IEE Proceedings - Generation, Transmission and Distribution*, vol. 143, no. 5, pp. 500–506, 1996.

[21] B. Stojkoska and K. Mahoski, "Comparison of different data prediction methods for wireless sensor networks," in *Proc. of the 10th Conference for Informatics and Information Technology (CIIT'13), Bitola, Macedonia*. Faculty of Computer Science and Engineering, Skopje, Macedonia, April 2013, pp. 307–311.

[22] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. of the 30th international conference on Very large data bases (VLDB'04), Toronto, Canada*, vol. 30. VLDB Endowment, September 2004, pp. 588–599.

[23] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *Proc. of the 22nd International Conference on Data Engineering (ICDE'06), Atlanta, Georgia, USA*. IEEE, April 2006.

[24] G. Box and G. Jenkins, *Time series analysis: forecasting and control*, ser. Holden-Day series in time series analysis and digital processing. Holden-Day, 1976. [Online]. Available: http://books.google.com/books?id=1WVHAAAAMAAJ

[25] D. Tulone and S. Madden, "PAQ: Time series forecasting for approximate query answering in sensor networks," in *Proc. of the 3rd European Conference on Wireless Sensor Networks (EWSN'06), Zurich, Switzerland, LNCS*, vol. 3868. Springer Berlin Heidelberg, February 2006, pp. 21–37.

[26] D. Tulone and S. Madden, "An energy-efficient querying framework in sensor networks for detecting node similarities," in *Proc. of the 9th international symposium on Modeling analysis and simulation of wireless and mobile systems (MSWiM'06), Torremolinos, Malaga, Spain*. ACM, October 2006, pp. 191–300.

[27] Q. Han, S. Mehrotra, and N. Venkatasubramanian, "Energy efficient data collection in distributed sensor environments," in *Proc. of 24th International Conference on Distributed Computing Systems (ICDCS'04), Tokyo, Japan*. IEEE, March 2004, pp. 590–597.

[28] S. Goel and T. Imielinski, "Prediction-based monitoring in sensor networks: taking lessons from MPEG," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 5, pp. 82–98, October 2001.

[29] Y.-A. Le Borgne, S. Santini, and G. Bontempi, "Adaptive model selection for time series prediction in wireless sensor networks," *Signal Processing*, vol. 87, no. 12, pp. 3010–3020, December 2007.

[30] S. Santini and K. Romer, "An adaptive strategy for quality-based data reduction in wireless sensor networks," in *Proc. of the 3rd International Conference on Networked Sensing Systems (INSS'06), Chicago, Illinois, USA*. Transducer Research Foundation TRF, May 2006, pp. 29–36.

[31] D. D. Biljana Risteska Stojkoska, Dimitar Solev, "Variable step size lms algorithm for data prediction in wireless sensor networks," vol. 14, no. 2, pp. 111–124, March 2012.

[32] B. Stojkoska, D. Solev, and D. Davcev, "Data prediction in wsn using variable step size lms algorithm," in *Proc. of the 5th International Conference on Sensor Technologies and Applications (SENSORCOMM'11), Nice, Saint Laurent du Var, France*. IARIA, August 2011, pp. 191–196.

[33] O. Maron and A. W. Moore, "The racing algorithm: Model selection for lazy learners," in *Lazy learning*, D. W. Aha, Ed. Springer, 1997, pp. 193–225.

[34] A. Jain, E. Y. Chang, and Y.-F. Wang, "Adaptive stream resource management using kalman filters," in *Proc. of the 2004 ACM SIGMOD international conference on Management of data (SIGMOD/PODS'04), Paris, France*. ACM, June 2004, pp. 11–22.

[35] W. W.-S. Wei, *Time series analysis*. Addison-Wesley publ, 1994.

[36] C. Liu, K. Wu, and M. Tsao, "Energy efficient information collection with the arima model in wireless sensor networks," in *Proc. of the IEEE 2005 Global Telecommunications Conference (GLOBECOM'05), St. Louis, Missouri, USA*, vol. 5. IEEE, November-December 2005, pp. 2470—-2474.

[37] G. Wei, Y. Ling, B. Guo, B. Xiao, and A. V. Vasilakos, "Prediction-based data aggregation in wireless sensor networks: Combining grey model and kalman filter," *Computer Communications*, vol. 34, no. 6, pp. 793–802, May 2011.

[38] F. Xia, Z. Xu, L. Yao, W. Sun, and M. Li, "Prediction-based data transmission for energy conservation in wireless body sensors," in *Proc. of The 2010 International Workshop on Ubiquitous Body Sensor Networks (UBSN'10), in conjunction with the 5th Annual International Wireless Internet Conference (WICON'10), Singapore*. IEEE, March 2010, pp. 1–9.

[39] Y. S. Suh, "Send-on-delta sensor data transmission with a linear predictor," *Sensors*, vol. 7, no. 4, pp. 537–547, April 2007.

[40] J. Lu, F. Valois, M. Dohler, and M.-Y. Wu, "Optimized data aggregation in wsns using adaptive arma," in *Proc. of the 4th International Conference of the Sensor Technologies and Applications (SENSORCOMM'10), Venice, Italy*. IEEE, July 2010, pp. 115–120.

[41] S. K. Soni, N. Chand, and D. Singh, "Reducing the data transmission in wsns using time series prediction model," in *Proc. of the International Conference Signal Processing, Computing and Control (ISPCC'12), Waknaghat Solan, India*. IEEE, March 2012, pp. 1–5.

[42] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[43] A. Ranganathan, "The levenberg-marquardt algorithm," June 2004. [Online]. Available: http://www.ananth.in/docs/lmtut.pdf

[44] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Proc. of the 1978 Biennial Conference Held at Dundee, Scotland, Lecture Notes in Mathematics*, G. A. Watson, Ed. Springer Berlin Heidelberg, June-July 1978, pp. 105–116.

Author Biography

Siamak Aram received his BS degree in Software Engineering. In December 2007 he was awarded the first prize in the student passive robotic contests organized by Sharif University. Further on, he graduated from Sharif University with M.Sc. degree in Information Technology in 2010 (International campus), with the thesis titled as "Automatic Web Services Composition using HSIP Model". As of today (at the time of this paper's submission), Siamak Aram is a final year PhD student in Control and Computer Engineering Department (DAUIN) at Politecnico di Torino under the academic supervision of Prof. Eros Pasero. Apart from his regular study on the Environmental Sensing applications, he has recently initiated another research project in the field of Medical Devices' applications. In 2014 he has commenced scientific work in Howard University in Washington DC as visiting scholar. His research interests include Distributed Systems, Web Services, Wireless Sensor Networks, Data Security and Data mining as well. His main project in Turin is dedicated to Low Power Wireless Sensor Networks. It is carried out in the premises of Neuronica Laboratory. Beside his scientific activities he is also a Politecnico di Torino's Academic Senate member and PhD representative at the Council of the Doctorate School.

Ikramullah Khosa is a PhD student in Department of Electronics and Telecommunication at Politecnico di Torino under the supervision of Prof. Eros Pasero. He graduated from National University of Sciences and Technology (NUST), Pakistan with MS degree in Electrical (Telecommunication) Engineering in 2010. His research interests include Image Analysis, Image Processing and Artificial Neural Networks. His current research is on ITACA Project at Neuronica Lab.

Eros G. Pasero is Professor of Electronics at the Politecnico of Turin since 1991 after a four year appointment as Professor at the University of Roma, Electronics Engineering. He was also Visiting Professor at ICSI, UC Berkeley, CA in 1991, Professor of digital electronics and electronic systems at Tongji University, Shanghai, China in 2011 and Professor of digital electronics and electronic systems at TTPU (Turin Tashkent Politechic University), Tashkent, Uzbekistan since 2012. Prof. Pasero interests lie in Artificial Neural Networks and Electronic Sensors. He created in 1990 the Neuronica Lab where hardware neurons and synapses are studied for neuromorphic approaches; neural software applications are applied to real life proof of concepts. Innovative wired and wireless sensors are also developed for biomedical, environmental, automotive applications. Data coming from sensors are post processed by means of artificial neural networks. Prof. Pasero is now the President of SIREN, the Italian Society for Neural Networks; he was v. General Chairman of IJCNN2000 in Como, General Chairman of SIRWEC2006 in Turin. He holds 5 international patents (two were the first silicon European neurons and synapse together Texas Instruments). He was supervisor of tenths of international Ph.D and hundredths of Master students and he is author of more than 100 international publications. He is now involved in several funded research projects among which: AWIS (Airport Winter Information Systems), ITACA (coffee and hazelnuts quality assessment processes, THOR (electrical intelligent car), OPLON (remote control of at risk persons), MIE (Intelligent Suistanable Mobility), NEC (Neural Engineering and Computation Lab), MAKE LAB.