ANFIS Analysis of Wireless Sensor Data with FPGA

Ahmed Khazal, Tuncay Ercan*

ABSTRACT

Applications related with WSNs may include thousands of separate sensor nodes, production and control data for different industrial sectors. It is important to manage these applications, monitor the network and reprogram the nodes to avoid operational problems. In this study, we propose a smart wireless sensor network using a reconfigurable embedded system of Field-Programmable Gate Arrays (FPGAs) with a soft-core processor. This software-based processor can be programmed dynamically and synthesized to implement the pre-processing of sensed data by ensemble Hybrid Neuro-Fuzzy algorithms such as Adaptive Neuro-Fuzzy Inference System (ANFIS). The first part of the proposed work is based on Matlab software to develop and train the ANFIS algorithm. Two different types of data sets (temperature and humidity) downloaded from Internet have been used in order to make a comparison between the Matlab Toolbox and modified ANFIS algorithm with momentum factor. The results obtained in this study have shown that the modified ANFIS algorithm is the convenient choice in terms of speed and accuracy.

Keywords: ANFIS, Neuro-Fuzzy System, FPGA, Sensor nodes, Smart node.

FPGA ile Kablosuz Sensör Verilerinin ANFIS Analizi

ÖZ

Kablosuz algılayıcı ağlarla ilgili uygulamalar binlerce ayrık algılayıcı cihazı, farklı endüstriyel sektörlerdeki üretim ve kontrol verilerini içerebilir. Bu uygulamaları yönetmek, bilgisayar ağını izlemek ve her bir cihazı tekrar programlayarak işletme problemlerinden kurtulmak çok önemlidir. Bu çalışmada yazılım tabanlı işlemci özelliğiyle yeniden yapılandırılabilir bir FPGA (Alan Programlanabilir Kapı Dizini) cihazı kullanan akıllı bir kablosuz algılayıcı ağı mimarisi önerilmiştir. Bu yazılım tabanlı işlemci, dinamik olarak tekrar programlanabilir ve ANFIS (Uyarlamalı Sinir-Bulanık Çıkarsama Sistemi) gibi bir toplu hibrit Sinir-Bulanık algoritması ile algılanmış verileri önişlemden geçirebilir. Önerilen çalışmanın ilk bölümü ANFIS algoritmasını geliştirmek ve eğitmek için Matlab yazılımı kullanmaya dayanır. Bu çalışmada internet ortamından tedarik edilmiş, sıcaklık ve nem verileri içeren iki veri seti ile Matlab ortamında hazır bulunan ve çalışma kapsamında momentum faktörü ile değiştirilerek geliştirilmiş ANFIS algoritmalarının karşılaştırması yapılmıştır. Bu çalışma değiştirilmiş ANFIS algoritmasının hız ve doğruluk açısından daha uygun olduğunu göstermektedir.

Anahtar Kelimeler: ANFIS, Bulanık Sinir Sistemi, FPGA, Algılayıcı düğüm, Akıllı düğüm.

Information of Author(s):

Ahmed Khazal
ORCID: 0000-0001-8171-5582
ahmedkhazal@gmail.com
Yasar University, Department of Computer Engineering

Tuncay Ercan
ORCID: 0000-0003-0014-5106
tuncay.ercan@yasar.edu.tr
Yasar University, Department of Computer Engineering

İSTANBUL UNIVERSITY — PRESS

(*) Contact Author
Address: Yasar University, Department of Computer Engineering, İzmir, Turkey ● **Telephone Number:** +90 232 570 82 41

1. INTRODUCTION

It is now a well-known fact that the use of Internet with the capabilities like communication, sharing of information and mutual interaction among people has changed our daily life significantly. The Internet of Things (IoT) is a new technological concept that the intelligent devices communicate with each other and form an intelligent and autonomous network in order to access Internet. Real-time continuous stream of data coming from the sensing devices and industrial equipment in the environment are aggregated, modified and transferred into the Information Systems (Storage, database, application services) provided by Cloud Service Providers on the internet. This kind of Industrial IoT (IIoT) information flow will result in changes that can positively contribute in our daily life, business life and industrial production systems.

Traditional internet communication infrastructure connects different communication systems and end users into each other. With the new developments in technology, many sensor devices can be integrated into the internet environment through WSN (Wireless Sensor Networks) systems [2][16] in order to collect the information and monitor the areas like the status of patients and elderly people, transportation, traffic flow and security and then take the appropriate decision for each case.

The requirements for wireless sensor networks like energy consumption, cost and processing capabilities, lead and motivate science and engineering disciplines to think in a different way. Applications related with WSNs may include huge number of nodes. Thus, it is important to be able to manage these applications, monitor the network and reprogram the nodes to avoid existing problems. In general, hardware part of sensor networks include processing elements (Microcontrollers), sensors, battery and wireless communication elements which sends data from sensors to sink node that routes the data to the central management system like a gateway [2][17]. Limited battery power, storage and computation capabilities, and security problems are limitation and disadvantages of traditional sensor nodes.

Due to all of these limitations, smart sensors have been developed to handle these challenges. In most cases, the smart sensor represents a system on chip (SoC) when integrated with FPGA devices. In such cases, the smart sensors can implement many intelligent functions such as pre-processing data, testing and performing complex algorithms, in addition to some communication tasks. FPGAs support reprogrammable technology which can be used for a reconfigurable sensor system [17]. They improved the efficiency of sensor systems by their architectural flexibility, re-configurability, processing capabilities, interfaces. They also perform many simple and complex functions [14][15]. These types of sensors are usually used for high-speed applications that need online measurements.

In many monitoring applications, sensor nodes route the aggregated data to a few number of sink nodes. In this case, transmitting unnecessary redundant data will cause a waste of communication channel bandwidth and energy consumption. The performance of data collection in a wireless sensor network will decrease when every node sends all collected data to the sink node. Moreover, this will increase collisions in the transmission channels. In order to increase the power of computation of sensor nodes, Field-Programmable Gate Arrays (FPGAs) and soft computation strategy such as Neuro-Fuzzy algorithms can be used within node architecture to overcome the limitation of traditional microcontrollers.

In our work, we present a new methodology to build and develop multiple soft-core processors of the FPGA set that perform ensemble soft computing algorithms in an IoT environment based on wireless sensor network. Ensemble technique has been proposed to design and build parallel embedded soft core FPGA microcontrollers within the reconfigurable sensor hardware part that can implement different types of soft computing algorithms. The ensemble of soft computation algorithms such as ANFIS algorithms offer more successful and highly efficient techniques rather than separate algorithms. Our work offers multi-functions and multi-sensing data to measure and deal with more than one physical signal simultaneously like temperature, humidity. Researchers recently use ensemble techniques to perform many tasks such as, control systems, self-testing error correction, weather forecasting [15], data classification, and prediction of energy consumption.

2. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM

In this approach, artificial neural networks are used to train some parameters of the fuzzy control system in an iterative way. This technique refers to a hybrid neuro-fuzzy system. In this technique, artificial neural network and control fuzzy system are performed together by keeping their different functions. This type of merging is suitable for control application, decision making, pattern recognition applications and classification. The idea of such hybrid model is to use a gradient-based learning algorithm in neural networks to determine its parameters through input and output patterns, so this will enable us to have advantages of learning through different patterns [11]. One of the most common examples that can be used in many different applications of Hybrid Neuro-Fuzzy systems is Adaptive-Neuro-Fuzzy Inference System (ANFIS).

ANFIS is one of the most important schemes that integrates the advantages of both neural and fuzzy systems in one model. In ANFIS, all parameters associated with linear and non-linear equations, number of rules and architecture of fuzzy system can be adapted by neural network algorithms. ANFIS algorithm has a very high learning speed and gives high accuracy in testing phase. Also, it has the ability to use human expressions (linguistic terms) when using to solve real life complex problems. It is simple to realize and can be used to control different application fields [12].

Normally ANFIS is used to map input pattern to membership functions (MFs), output of MF to a set of "if-then rules", rules to a set of output sets, output sets to output of linear MFs, and the output MFs to a single valued output or a decision associated with the output. A typical ANFIS structure is depicted in Figure 1.
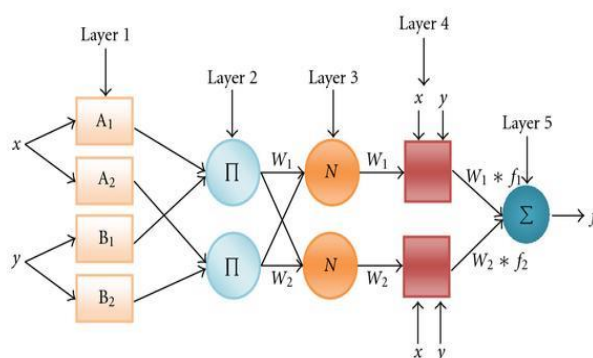

Figure 1. Adaptive Neuro-Fuzzy Inference System

According to figure 1, ANFIS structure normally has 5 layers of neurons [12]:

Layer 1: Each node generates the membership degree of linguistic terms that are given by the following Eq. 1 and Eq. 2:

$$Q_{1,i} = \mu_{Ai}(x) , \quad i = 1,2,3, \tag{1}$$

$$Q_{2,i} = \mu_{Bi}(y) , \quad i = 1,2,3, \tag{2}$$

Where "x" and "y" are the inputs, "I" is number of the nodes, and A and B are the linguistic terms. Eq. 3 is an example of a membership function that is bell-shaped function:

$$\mu_{Ai}(x) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2bi}} , \quad i = 1,2,3, \tag{3}$$

As we can see, the bell-shaped function has three parameters a, b, and c that are called premise parameters.

Layer 2: Each node calculates the firing strength of each rule using the "min" or "prod" operator. Normally, AND operation is the most commonly used operation in fuzzy systems. The output of this layer can be as the following Eq. 4:

$$Q_{2,i} = w_i = \mu_{Ai}(x) * \mu_{Bi}(y), \quad i = 1,2,3, \tag{4}$$

Layer 3: In this layer, normalized process is done. Every node in this layer calculates the proportion of each rule to the aggregate of all conditioning rules in previous layer. The number of the normalization rules is equal to the number of rules in layer 2. In general the formula in Eq. 5 can be used to represent the output of this layer.

$$\boldsymbol{Q_{3,i}} = \overline{\boldsymbol{w}}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2,3, \tag{5}$$

Layer 4: The outputs of this layer are the product of the normalized firing strength coming from layer 3. The nodes compute a parameter function on the layer 3 output. Parameters in this layer are called consequent parameters and can be represented as Eq. 6.

$$\boldsymbol{Q_{4,i}} = \overline{\boldsymbol{w}}\boldsymbol{f_i} = \overline{\boldsymbol{w}}(\boldsymbol{p_i}\boldsymbol{x} + \boldsymbol{q_i}\boldsymbol{y} + \boldsymbol{r_i}), \ i = 1,2,3, \tag{6}$$

Where "p", "q" and "r" are called consequent parameters.

Layer 5: In this layer, a single node aggregates all incoming signals from layer 4 to calculate a single output as in Eq. 7.

$$Q_5 = \sum \overline{w}f_i, \ i = 1,2,3, \tag{7}$$

Training and Testing in ANFIS begin by dividing the input data sets into training, testing and checking groups. The training data are normalized to be suitable for the training process using the Min, Max method. It's done by mapping each input value between binary 00, 01 and 10 numbers. After that, the forward path which is the first phase in hybrid algorithm will start to initialize and train the consequent linear parameters in layer 4 based on Linear Regression algorithm. After the Linear Regression process finished, the second phase in the hybrid algorithm which is based on back-propagation algorithm will begin training the nonlinear promise parameters which find in layer 2.

The finishing of training process for the ANFIS can be done by two methods. In the first method, training ANFIS ends when the threshold error is less than the specific limit which is defined at the beginning of the training phase. In the second method, ANFIS stops learning after a specific number of learning iterations. In our case, the ANFIS algorithm is stopped learning based on the limited number of iterations.

3. EXISTING STUDIES

A With the advancements of using reconfigurable FPGA and soft computing techniques within WSNs, many studies have been done on this field. The flexibility and efficiency of FPGA devices motivate the researchers to use them in many soft intelligent techniques such as [3]. The authors used FPGA to implement an adaptive neuro-fuzzy system based on radial basis neural network to reduce and optimize the hardware resources with the fixed-point and simplified floating-point arithmetic.

Another study by [8] shows the ability of implementing the meta-heuristic learning algorithms of neuro-fuzzy system (NFS) on the FPGA based on improved particle swarm optimization (iPSO). According to the authors, Virtex5 FPGA board was used to implement the neuro-fuzzy system. Particle swarm optimization technique was used to train and update neuro-fuzzy parameters. The researchers used this kind of integration to speed up the training algorithm and minimize the usage of hardware resources such as size of memory and number of multipliers.

Another FPGA implementation of Adaptive Neuro-fuzzy Inferences Systems (ANFIS) for controlling temperature and humidity data sensors inside a greenhouse using geothermal energy as a power source for heating system was proposed in [9]. Four techniques are used in this approach; artificial neural network, PI control, fuzzy logic control,

and adaptive neuro-fuzzy control to control the indoor temperature. The main advantages of using the hardware language approach are rapid prototyping and allowing usage of powerful synthesis controller through the use.

In [5], the implementation on reconfigurable hardware of a Sugeno type for adaptive Neuro-Fuzzy inference system is proposed. The pipeline and parallel pipeline architecture play an important role in modelling the algorithm for the FPGA based implementation. They used Matlab Simulink toolbox to simulate the proposed controllers, implement and simulate the adaptive Neuro-Fuzzy inference system algorithm using FPGA boards based on both high level synthesis tool and VHDL coding. The pipeline structure of the proposed hardware was designed in a way that permits the parameters to update in parallel. The hardware model's processing speed is very high, and allows the controller to be used in real-time applications.

In [19], the adaptive neuro-fuzzy model was implemented based on FPGA.In this article, High Level Synthesis (HLS) tool with C/C++ language from Xilinx was used to design the embedded soft core within FPGA fabric. The authors describe some of optimization methods to implement and realize the adaptive neuro-Fuzzy algorithm. In this article, The authors Use a high level language (C-language) instead of Hardware Description Language (HDL). So, the information about hardware details are not needed. According to the authors, using such kind of embedded IP core can be used for variety real time applications.

In addition to the control systems that appear in previous works, hybrid neuro-fuzzy can be used for power prediction system [13][1]. The authors proposed a power prediction system for wind energy generation. In this work, the sensor networks are used to perform some measurement and send the results to the main station. The main station performs many analysis to the sensing data based on data mining algorithms like Fuzzy C-Means algorithm (FCM) which is used to estimate the fuzzy rules for   hybrid neuro-fuzzy systems [1]. The prediction system can be used for Damage prediction for wind turbines using both wireless sensor and actuator networks [13]. Fuzzy control system is used to remove the effect of overheating by forecasting the damage in wind turbines. The authors in this work use the adaptive Neuro-Fuzzy inference system to build a real database based on real temperature, frequency, and the effect of temperature on the natural frequency values. Then, using this database to enhance the performance of Fuzzy control system.

In [7], an online learning recurrent Neuro-Fuzzy classifier algorithm is proposed for use in classification applications. The recurrent network is embedded in the RNFC by adding feedback connections in the second layer, where the feedback units act as memory elements. They show that effective neuro-fuzzy classifier should be able not only to adaptively adjust fuzzy membership functions but also to dynamically adapt fuzzy operators.

In [6], a complete hardware and software system was designed based on embedded FPGA device to develop a soft controller for intelligent environments. A group of embedded IP cores are used to build   neuro-fuzzy architecture which represents the hardware portion of the system. MicroBlaze core processor that represents the software part used to drive and control the operation for overall system. Their approach is simplifying an ANFIS model in order to reduce the computational costs of the algorithm and to promote the scalability and modularity of its HW implementation. In addition, we exploit the redundancy inherent in ambient-intelligence data in order to reduce the dimensionality of the system. A high speed, scalability and efficiency can be achieved by using FPGA devices compared with other implementation based soft computing mechanisms. Another implementation of neuro-fuzzy inference system based on FPGA Zynq System on Chip development board with a dual-core ARM Cortex A9 processor is presented in [18]. The ANFIS algorithm is designed and implemented on the FPGA based on the VHDL language. The algorithm is used to correct the corner displacements of the vehicle.

4. METHODS

The architectural flexibility of FPGA provides high efficiency such as parallelism, and the performance through the development of algorithms. So a new generation of reconfigurable, smart, and low level of energy consumption sensors is developed based on FPGAs. Figure 2 explains the general architecture of sensor node based FPGA [10].
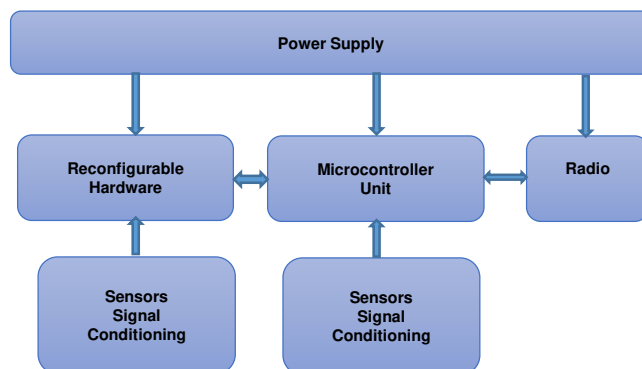

Figure 2. Architecture of FPGA with sensors

In our work, reconfigurable part within a wireless sensor node is used for implementation of multiple and different applications to control of sensing data, decision making, analysis and advanced data management based on ANFIS algorithm. ANFIS algorithm is the best Neuro-fuzzy algorithm that is used to control the nonlinear applications. That's why we will use the ensemble technique to design and build parallel embedded soft FPGA based MicroBlaze microcontrollers within the sensor reconfigurable hardware part. The block diagram in Figure 3 shows the proposed reconfigurable part:
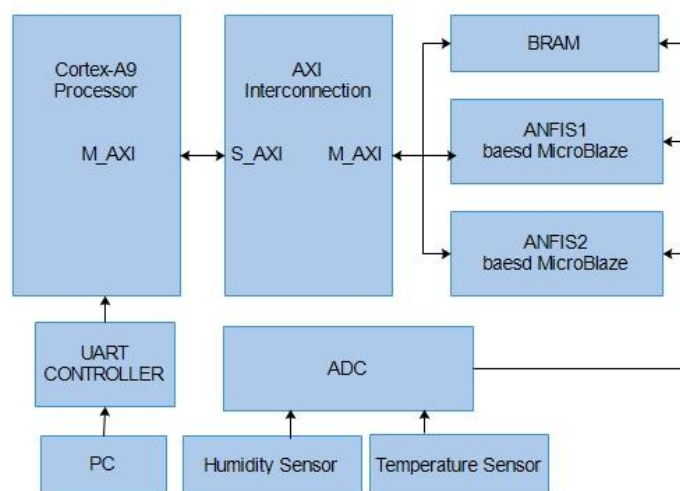

Figure 3. Proposed Reconfigurable Architecture (System)

The ensemble of the design system in figure 3 is composed of hardware and software part. The basic hardware part was designed using Vivado 2016.4 – Block Design which is an important tool from Xilinx to start creating the design system within Zynq®-7000 AP SoC ZC702 Evaluation Kit. The hardware part consists of two ANFIS algorithms based on MicroBlaze soft-core processors. The first ANFIS algorithm is connected to temperature sensor and other algorithm is connected to humidity sensor with their associated BRAM (Block Random Access Memory), UART (Universal Asynchronous Receiver Transmitter) controller, and ADC (Analog to Digital Convertor) controller.

The Xilinx Advanced eXtensible Interface (AXI) which is a kind of smart microcontroller buses within FPGA board is used to join and aggregate the outputs of hardware algorithms (slave AXI) to the main processor (master AXI).

The software part is done using Xilinx Software Development Kit (SDK 2016.4) is represented by compiler, libraries, and application programs. The hardware system is programmed using C language to implement the developed ANFIS algorithms in our hardware. Both hardware and software platforms are transferred into a bit file which is ready to be downloaded on the FPGA device to create the system. Figure 4 summarizes the sequence of operations that did by FPGA processor (Cortex-A9).
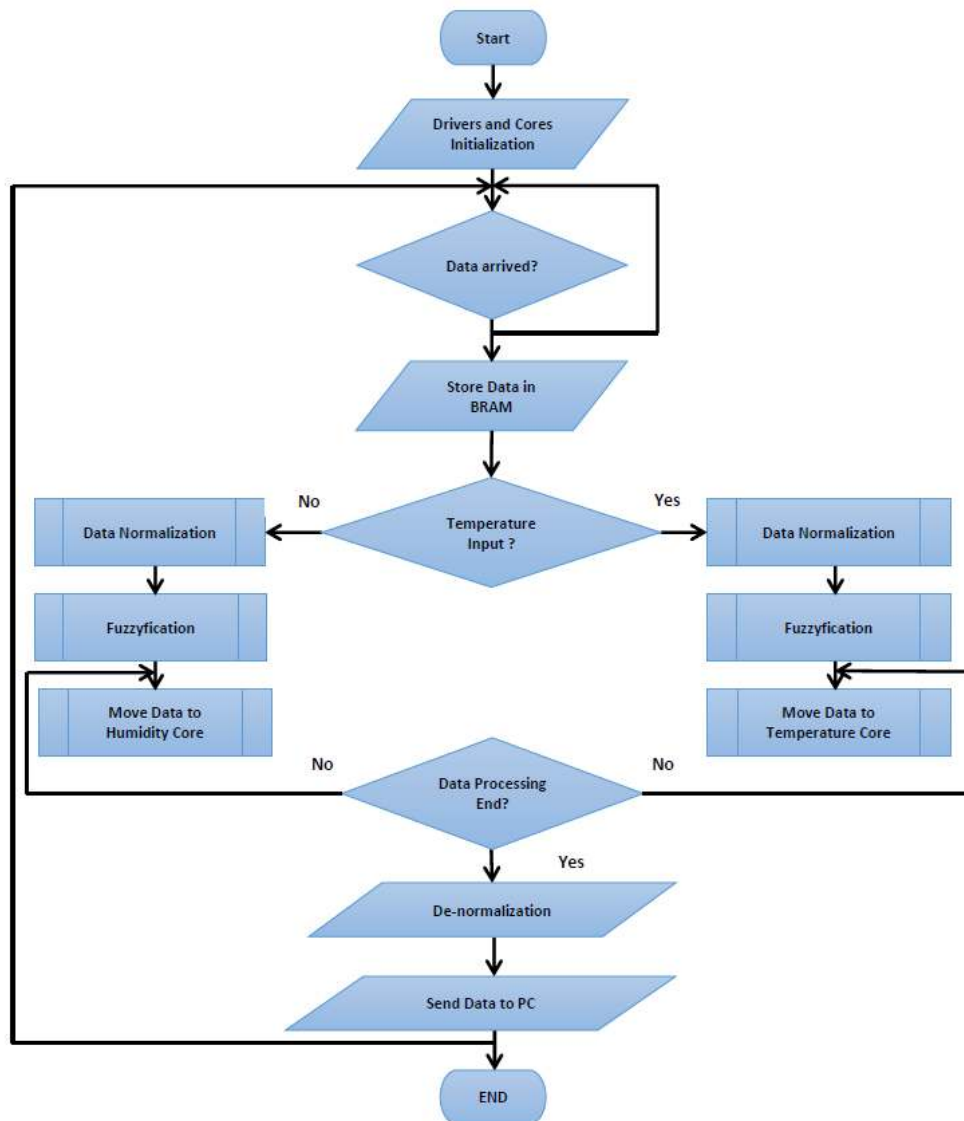
Figure 4. Sequence of operations main processor

The Cortex-A9 processor starts to initialize all drivers and hardware cores, stores the collecting data to BRAM, and then performs the data processing operations that involve:

- Data classification.

- Data normalization

- Data fuzzification

- Data routing: In next step, the processor routes each data set to the equivalent MicroBlaze cores and then, reading backs the results in order to send them to the host computer via serial communication.

5. FINDINGS

In Matlab environment, most of researchers try to use a graphical user interface (anfisedit-GUI) for ANFIS algorithm or a function provided by a Matlab Toolbox, which is "anfis-command" to construct and do the testing and training that makes it simple to use and easy to follow.

In our approach, we've modified ANFIS algorithm and combined it with gradient descent optimization algorithm during learning phase to minimize the error, prevent stuck in local minima and reach a global minima. This optimization algorithm is known as Momentum that speeds up the steps that is taken towards the optimal solution.

In Momentum, a fraction "m" of the previous weight update is added to the current weight. As a result, we provide a faster convergence and speed up the training process. This could not be realized if we apply ANFIS function provided by Matlab Toolbox. In figure 5, the pseudo code explains how to add the momentum factor into the ANFIS algorithm.

```
Set the type of membership function
(Gaussian function)
Load initial premise parameters (a,b,c)
Load initial consequent parameters (p,q,r)
Load initial momentum factor (m)
Input the sensor data
Normalize the data

// start forward path

While (error > threshold) then

    Generates the membership degrees for input
    Normalize and aggregate generated data
    Calculate the output of the algorithm
    Updata consequent parameters
    p_new=p_old + delta(error) * input data
    q_new=q_old + delta(error) * input data
    r_new=r_old + delta(error)

  // start backward path

// update premise parameters
    a_new=a_old + delta(error) + momentum factor (m)
    b_new=b_old + delta(error) + momentum factor (m)
    c_new=c_old + delta(error) + momentum factor (m)

End while loop
De-normalize the output data
```

Figure 5. Pseudo code of the proposed algorithm

Two types of data sets (temperature and humidity) taken from Beach Weather Stations - Chicago Park are used for both our modified code with momentum factor and Matlab Toolbox for ANFIS [20][21]. Each data set consists of 13917 samples from 5/22/2015 to 12/31/2015. Figure 6 shows the effective of adding the momentum factor to the ANFIS algorithm.

According to the Figure 6, the mean square of error with momentum factor is (0.9058) after 40 iterations, while as equal (23.0417) after the same number of iterations.
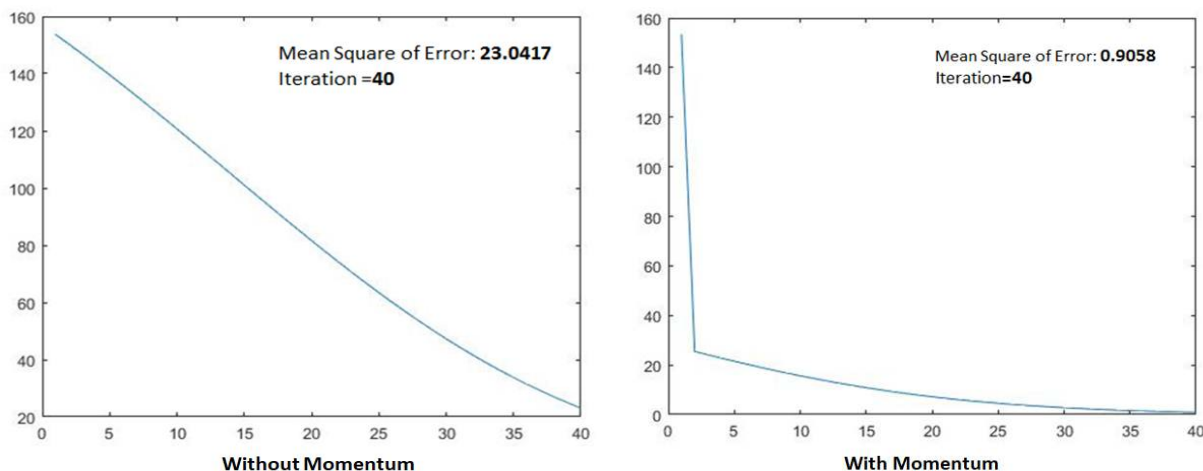

Figure 6. Training and Testing results for Temp. data set %70 training and %30 testing.

Table 1 and Table 2 show the error after training and testing the data sets.

Table 1. Training and Testing results for Temperature data set %70 training and %30 testing.

Technique type	Training	Testing
Matlab Toolbox	3.7802e-11	0.013204
Modified Algorithm	1.3943e-14	0.0010988

Table 2. Training and Testing results for Humidity data set %70 training and %30 testing.

Technique type	Training	Testing
Matlab Toolbox	9.3506e-11	0.062509
Modified Algorithm	7.4105e-15	0.001184

while the table 3 and 4 show the effect of adding the momentum factor on speed in training and testing phases.

Table 3. Training Speed (Execution Time) in Sec.

Execution Time in Sec (Training)		
Data Type	ANFIS Toolbox	Modified Algorithm
Temperature	3.675	1.1921
Humidity	3.6427	1.1895

Table 4. Testing Speed (Execution Time) in Sec.

Execution Time in Sec (Testing)		
Data Type	ANFIS Toolbox	Modified Algorithm
Temperature	0.0124	0.011
Humidity	0.0122	0.012

According to table 3, we can easily see that adding the momentum factor to the modified algorithm reduces the execution time for training phase about 60% compared with Matlab toolbox algorithm in both temperature and

humidity data sets. While the execution time in table 4 for testing phase is almost the same for both algorithms because there are no iteration processes happened in both algorithms under the test condition.

6. DISCUSSION AND CONCLUSIONS

In this paper, implementation of adaptive Neuro-Fuzzy algorithms using FPGA was proposed. Ensemble of the multiple ANFIS algorithm is used in order to improve the efficiency and accuracy for a sensor node. Because of using different ways to train and simulate Ensemble ANFIS within a single wireless sensor node, we generate a kind of intelligent system. In our proposed method, pre-processing the data collected from different sensors is possible and can be used in diverse applications such as prediction, data cleaning, data smoothing and data compression. As a result, we can eliminate unnecessary data and reduce the traffic on transmission channels.

On the other hand, two methods were applied to simulate the temperature and humidity data sets: a Matlab Toolbox and modified algorithm with momentum factor for ANFIS. Using modified algorithm with momentum provides a best and fast convergence during training process. Our modified algorithm is also more flexible and can be modified to combine with other optimization algorithms. The results have shown the efficiency of using the momentum factor with ANFIS algorithm. In the future, we will try to combine a modified ANFIS with different types of soft computation algorithms for the sensor node.

REFERENCES

[1] Ahmed, E.,Mohamed, S., Khaled, M., Ahmed, A., (2016), A hybrid neuro-fuzzy power prediction system for wind energy generation, International Journal of Electrical Power & Energy Systems,74, 384-395.

[2] Akyıldız, L., Sankarasubramaniam, Y., Su, W., Cayırcı, E. ,(2002), "Wireless sensor networks: A survey", Journal of Computer Networks, 38, 393-422.

[3] Andrzej, P., Meng, J. ,(2016), "The method of hardware implementation of fuzzy systems on FPGA", International Conference on Artificial Intelligence and Soft Computing, ICAISC 2016, 284-298.

[4] Brassai, S., Hajdu, S., Tamas, T., 4(2015), "Hardware implementation of a neuro-fuzzy controller using high level synthesis tool", Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics, 10.1515/macro-2015-0018.

[5] Brassai, S., Hajdu, Sz., Tamas, T., Bakó, L., (2015), "Hardware implemented adaptive neuro-Fuzzy system", Carpathian Control Conference, 10.1109/Carpathian CC.2015.7145046.

[6] Campo, I., Basterretxea, K., Echanobe, J., (2012), "A System-on-chip development of a neuro–fuzzy embedded agent for ambient-intelligence environments", IEEE Transactions on Systems, 10.1109/TSMCB.2011.2168516.

[7] Cheng-Jian, L., Chun-Cheng, P., (2014), "Classification using an efficient neuro-fuzzy classifier based on adaptive fuzzy reasoning method", Conference on Computer, Consumer and Control, 10.1109/IS3C.2014.34.

[8] Cihan, K.arakuzua, F., Mehmet, A. ,(2016), "FPGA implementation of neuro-fuzzy system with improved PSO learning", Journal of the International Neural Network Society,79- 2016, 128-140).

[9] Doaa, M., Hanaa, T., (2017), "Analysis and design of greenhouse temperature control using adaptive neuro-fuzzy inference system", Journal of Electrical Systems and Information Technology, 4(1), 34-48.

[10] François, Ph., (2014), "Runtime hardware reconfiguration in wireless sensor networks for condition monitoring", Universitäts - und Landesbibliothek, Darmstadt.

[11] Janusz, K., (2000), "Introduction to neuro-fuzzy systems", Advances in Soft Computing. Springer-Verlag, ISBN 978-3-7903-1852-9. Berlin/Heildelberg.

[12] Janusz, K. (2002), "Neuro-fuzzy architectures and hybrid learning", Springer-Verlag,ISBN 978-3-7903-1802-4, Berlin/Heildelberg.

[13] Maicon, M.,Luci, P., Silvana, R., Flavia, C., Claudio, M., Paulo F., Albert, Y., (2017), "Damage prediction for wind turbines using wireless sensor and actuator networks", Journal of Network and Computer Applications. 80,123-140

[14] Meena, S. & Krishna, N., (2014), "Simulation of dynamically reconfigurable wireless sensor node", International Conference on Electronics and Communication System, Coimbatore, 10.1109/ECS.2014.6892795. India.

[15] Melin, P., Soto, J., Castillo, O., Soria, J., (2013), "Time series prediction using ensembles of ANFIS models with genetic optimization of interval type-2 and type-1 fuzzy integrators", Journal of Hybrid Intelligent Systems, 10.3233/HIS-140196.

[16] Potdar, V., Sharif, A., Chang, E., (2009), "Wireless sensor networks: a Survey", International Conference on Advanced Information Networking and Applications Workshops, 10.1109/WAINA.2009.192. UK, Bradford.

[17] Rajasekaran, C., Jeyabharath, R. and Veena, P., (2014), "Hardware-software reconfigurable techniques for wireless sensor network", Journal of Applied Sciences, Engineering and Technology 8(17): 1855-1862.

[18] Rasika, A. and Pathan, M., (2015), "Design and implementation of ANFIS algorithm using VHDL for vehicular system", Journal on Recent and Innovation Trends in Computing and Communication 3, 2: 820-824.

[19] Sándor, T., Szabolcs, H., Tibor, T., (2015), "Embedded adaptive neuro-fuzzy inference system with hardware implemented real time parameter update", International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics, 10.1515/macro-2015-0021.

[20] Chicago. , (2015). "Beach Weather Stations - Automated Sensors - 2015 - Air Temperature | City of Chicago | Data Portal". [online] Available at:https://data.cityofchicago.org/Parks-Recreation/Beach-Weather-Stations-Automated- Sensors-2015-Air-/rsk3-iyyk

[21] Chicago. , (2015). "Beach Weather Stations - Automated Sensors - 2015 - Humidity | City of Chicago | Data Portal". [online] Available at: https://data.cityofchicago.org/Parks-Recreation/Beach-Weather-Stations-Automated-Sensors-2015-Humi/4tf5-5fw5