Measurement: Sensors 24 (2022) 100485

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

# System identification using neuro fuzzy approach for IoT application

Rakesh Kumar Pattanaik [a], Srikanta Kumar Mohapatra [b], Mihir Narayan Mohanty [a], Binod Kumar Pattanayak [a,*]

[a] ITER, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, India
[b] Chitkara University Institute of Engineering & Technology, Chitkara University, Punjab, India

ARTICLE INFO

*Keywords:*
Adaptive neuro fuzzy system
Wireless sensor Network channel
Nonlinear dynamic system Identification
Internet of things
Autoregressive moving average

ABSTRACT

The Internet of Things (IoT) has become a popular application in recent years. However, it is the wireless communication mode. In such a scenario, the user would have to send information either nonlinear or dynamic data type in the form of a signal or an image, videos depending on the application. The proposed work is focused on this model identification that tends to nonlinear dynamic system identification for IoT applications. An Autoregressive Moving Average (ARMA) model represents model for IoT application. To verify the model supremacy, an ARMA bench mark system is used. The adaptiveness is proved through variation of weights and can be universally used for the next generation. In the first attempt, the Multilayer Perceptron model (MLP) is considered as the ARMA system and observed. Further, to improve its accuracy, the Adaptive Neuro-Fuzzy system (ANFIS) model is designed for system identification. It is shown in the result section that it identifies better than the MLP as well as traditional system identification techniques.

## 1. Introduction

The development of the wireless communication channel has many more implications for individuals and the world at large. Due to reflection and diffraction, electromagnetic waves travel along different paths when they travel through a typical wireless channel. The multipath channel lowers the quality of the signal that is received and causes serious multipath interference. To solve this problem, we need a good wireless channel identification algorithm. Also, in the Internet of Things (IoT), many sensors built into different environments put a lot of weight on recognising scenarios so that the wireless communication system can be designed, deployed, and managed in the right way and have better communication performance [1]. Because of this, it is hard to find the right wireless channel scenarios to meet the specific requirements of the wireless communication process. Urban macrocells, urban satellites, indoor hotspots, and other types of propagation environments by referring to them as "wireless channel scenarios" [2,3]. The two main types of propagation environments are line-of-sight (LoS) and non-line-of-sight (NLoS) conditions. Even if two scenarios are in the same environment, their wireless channel characteristics are usually very different from each other. This motivates the development of a wide range of empirical statistical wireless channel models for certain types of propagation conditions. In order to estimate the parameters of WSN channel, the received signal must be combined with prior knowledge of the transmitted sequence of symbols [4]. During a time period where it is expected that the channel will remain constant, block-based estimating approaches use a block of received symbols to estimate the average channel. In order to create a discrete FIR-channel, the channel's continuous impulse response is sampled in time and in delay. This gives snapshots of the channel as it changes over time. The estimation error in the channel samples is a function of the measurement noise, the transmitted symbols, the channel parameters, and the estimating technique. The goal of parameter estimation is to find, from a given class of models, the one that comes closest to the system's external properties. This is done by comparing the input and output data in general. There are two ways to figure out the system's mathematical model: system identification and excitation analysis. In the process of exception analysis, the task is to analyse and identify the systems in terms of physical and chemical laws. The second method is to analyse how the system works and how the experimental data is passed. System identification determines a system's mathematical model from input and output data. It is divided into two methods. The first one is parametric identification; the second one is non-parametric model identification. The parametric model is defined in the form of a mathematical model with differential

* Corresponding author.
*E-mail addresses:* rakeshpattanaik.888@gmail.com (R.K. Pattanaik), srikanta.2k7@gmail.com (S.K. Mohapatra), mihir.n.mohanty@gmail.com (M.N. Mohanty), binodpattanayak@soa.ac.in (B.K. Pattanayak).

https://doi.org/10.1016/j.measen.2022.100485
Received 9 September 2022; Received in revised form 23 September 2022; Accepted 26 September 2022
Available online 29 September 2022

equations and state equations. The mathematical model with parameters like impulse response, frequency response, and transfer function are non-parametric model representations [5,6]. There are several applications of parameter estimation and identifying the model's impulse response. In order to identify or estimate an unknown parameter or several unknown parameters in a massive MIMO WSN, the researchers in Refs. [7,8] have created unique and effective signal processing algorithms. The parameters of the Volterra series can be learned. Various parameter estimations for WSN channels are presented in Refs. [9–18]. Parameter estimation of WSN channel shall be considered as the problem of nonlinear system identification. Some studies related to nonlinear dynamic system identification are presented as the WLMS algorithm [19], Adaptive Algorithms [20], Wilcoxon algorithm [21], and FLLWNN [22]. Further, by implementing a novel kernel algorithm, system identification of a nonlinear system is presented in Refs. [23,24], a multi-layer recursive model [25]. Further, by using Lyapunov stability analysis and system identification, it is further presented in Ref. [26].

A WSN channel can be described as a time-varying ARMA or AR-process. The collection of random AR processes can be used as a single ARMA with the same first and second order statistics through spectral factorization. An AR-model can provide a good approximation to the dynamics when the channel's ARMA-model poles are located close to the unit circle. In this work, the ANFIS approach is used to build the IoT model. The adaptability is shown by the different weights, and it can be used by everyone in the next generation.

The rest of the sections are organised as follows: After introducing existing methods in Section 1, Section 2 presents the problem formulation part. In a further section, section 3, the methodology part is discussed in section. In Section 4 discussion of the suggested method's performance evaluation and comparison with other models is analyzed. In the final section, section 5, the conclusion of the work, is presented.

## 2. Problem formulation

### 2.1. Auto Regressive Moving Average (ARMA)

In general, physical systems are represented as simultaneous differential equations, and all of these equations can be summed up into a single Auto Regressive Moving Average equation:

$$\left(\beta^{N}+\nabla_{N-1}\delta^{N-1}+\ldots+\nabla_{1}\beta+\nabla_{0}\right)X_{t}=\left(\delta_{M}\beta^{M}+\delta_{M-1}\delta^{M-1}+\delta_{M-2}\beta^{M-2}+\ldots+\delta_{1}\beta+\delta_{0}\right)U_{t} \tag{1}$$

where the notations represented as *X(t)* is output of the system at time *t*, *U(t)* is the input of the system, β is difference operator $\beta\ (tX=X(t)\text{-}k)$, ∇, is represented as autoregressive coefficient and δ represented as coefficient of moving average. Since the inputs *U(t)* is unknown in the above equation, therefore the known outputs *X(t)* to estimate the system coefficients so that squares of *U(t),* is as low as possible. If we set all the values of $\delta$ except, $\delta_0$ to zero and set, $\delta_0$ to one in the above differential equation presented as an (AR) equation,

$$\left(\beta^{\mathrm{N}}+\nabla_{\mathrm{N}-1}\delta^{\mathrm{N}-1}+\ldots+\nabla_{1}\beta+\nabla_{0}\right)\mathrm{X}_{\mathrm{t}} \tag{2}$$

Similarly, a Moving Average (MA) equation is presented as

$$\left(\delta_{M}\beta^{M}+\delta_{M-1}\delta^{M-1}+\delta_{M-2}\beta^{M-2}+\ldots+\delta_{1}\beta+\delta_{0}\right)U_{t} \tag{3}$$

In general, an ARMA model's system transfer function is a pole-zero model, while an AR model's transfer function is an all-pole model, and a Moving average (MA) model's transfer function is an all-zero model. Let another example be considered to predict the ARMA model. where an ARMA (p, q) is given as

$$\nabla(t)U(t)=\beta(t)e(t) \tag{4}$$

In another form

$$X(t)+\alpha_{1}X(t)_{n-1}+\ldots+\alpha_{p}X(t)_{n-p}=e(t)+b_{1}e(t)_{n-1}+\ldots+b_{q}e(t)_{n-q}$$

where, *e(t)*is represented as white noise, variance $\sigma_e^2$, and $\nabla(t)$ is define as

$$\nabla(t)=1+\alpha_{1}t^{-1}+\alpha_{2}t^{-2}+\ldots.+\alpha_{p}t^{-p} \tag{5}$$

In the same way, a polynomial $\beta(t)$ is polynomial, in $t^{-1}$ with order $q$. If $\nabla(t)$has no poles for "$t$" on or outside the unit circle, the process is stationary. Model parameters$\nabla(t)^{'}$and $\beta^{'}(t)$ of orders $q^{'}$and$q^{'}$, which are not necessarily equal to $p$ and $q$, are commonly discovered by estimating parameters from N observations $X_n$, n = *1, N;* the observations can be a realisation of the process (1), but this is not required. With this model, future predictions of the process (1) can be formed by substituting those estimated parameters into new data *U(t),* which can be represented as

$$\beta^{'}(t)\widehat{e}(t)=\nabla(t)^{'}U(t) \tag{6}$$

where the signal $\widehat{e}(t)$ is the estimated model's output with U(n) as the input signal; the estimated$\beta^{'}(t)$ is employed as the (Autoregressive)AR portion and $\nabla^{'}(t)$ as the MA part in this derivation of $\widehat{e}(t)$. The prediction error squared PE is defined as the expectation $E[\widehat{e}_t^2]$where *U(n)* is a realisation of the process that is independent of the observations *X(t)* used to estimate the parameters. By substituting *U(t)* with $\nabla^{'}(t)$ and $\beta^{'}(t)$ ,the output $\widehat{e}(t)$ of the model with equation (4)

$$\widehat{e}(t)=\frac{\nabla^{'}(t)}{\beta^{'}(t)}\ U(n)=\frac{\nabla^{'}(t)\beta(t)}{\nabla(t)\ \beta^{'}(t)}\ e(t)=\frac{D(t)}{C(t)}e(t) \tag{7}$$

As a result, an ARMA provides the relationship between the model's error output $\widehat{e}(t)$ and the innovations $e(t)$ (6), that formed the genuine process of ARMA $p+q^{'},p^{'}+q$ process$\frac{D(t)}{C(t)}$ Consider (7) as a filtering operation [26].

## 3. Methodology

a. WSN Channel Model

Two devices communicate through a channel. The input signal *'U'* from the transmitting device appears as *'X'* for the receiving device. The channel acts as a band pass filter, whose transfer function *'T'* is represented

$$T=\frac{U_f}{X_f} \tag{8}$$

Ideally, the channel should be a linear time-invariant system. The system or channel here is a natural system and does not behave ideally. It is modelled as a dynamic and non-linear system. So, in this work, an ARMA model is used to identify the system's transfer function so that it closely approximates the natural channel.

Fig. 1 is the block diagram representation of the identification process. The operator *U (t)* indicates input to the WSN Channel via a transmitter device. The channel is denoted as *X (t)* through a receiving

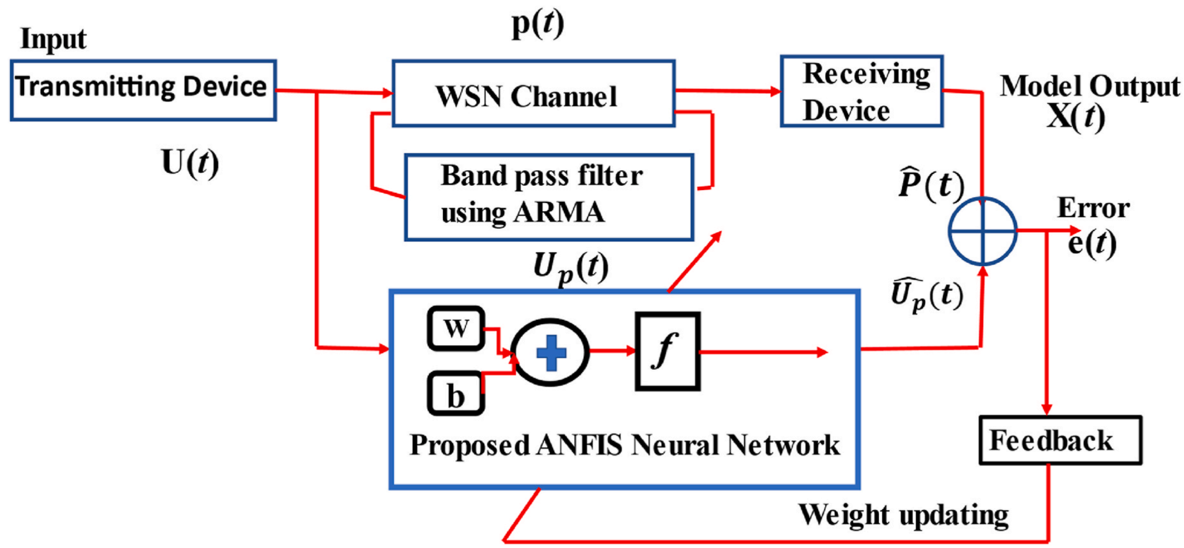


**Fig. 1.** Block diagram representation of Proposed Identification process.

device. The WSN channel is denoted as *P (t)*. This channel acts as a band pass filter and it is designed using the ARMA model. As discussed, the WSN channel is a nonlinear system. This inspires us to consider a nonlinear system identification problem. where the task is to identify the parameters of the unknown system. If prior information about the system is known, then the problem shall be considered a control problem. Whereas the system parameters are unknown, this shall be considered as the system identification problem. To identify the given system or WSN channel, a nonlinear ANFIS neural network-based model is designed as a replica model. In order to validate the proposed ANFIS model's output response, the WSN system's parameters are provided to the model. Further, the error between the two output has to be reduced using certain learning algorithm such as gradient decent, least mean square etc, until it tracks perfectly. It is known as the weight updating process. From the above figure, $U(p)_t$ represents the proposed model and output of the model is $\widehat{U}_P(t)$. $\widehat{P}(t)$ is an approximation output of X(t). Therefore, the aim is to create a neural network model with the same parameters as the WSN model.

$$\|\widehat{X} - \mathrm{X}\| = \|\widehat{P}(\mathrm{X}) - \mathrm{P}(\mathrm{X})\|, \leq \epsilon \;\; x\epsilon X \tag{9}$$

where, $\epsilon \leq 0$, and $\epsilon$ is some desired value, $\widehat{X}=\widehat{P}$ , $X=$ P and $\|.\|$is Euclidian norm. *'U'* and *'X'* are a subset of real number respectively for static system. Assuming a Lebesgue integrable function with the interval *[0, T]* or *[0 to ∞]* defines the bounds of a dynamic system. Error of the model is calculated as:

$$e = \|\mathrm{X} - \widehat{X}\| \tag{10}$$

b. Adaptive Neuro-fuzzy

Each layer of an ANFIS model consists of five sub-models, each of which is made up of nodes that stand for the same mathematical operator. Fuzzification of the input vector *X* signals is performed by Layer 1. This layer, also known as the IF-Part layer, calculates the degrees to which each input $U_i$ satisfies each of its fuzzy sets. These degrees are commonly referred to as membership degrees. To process the incoming

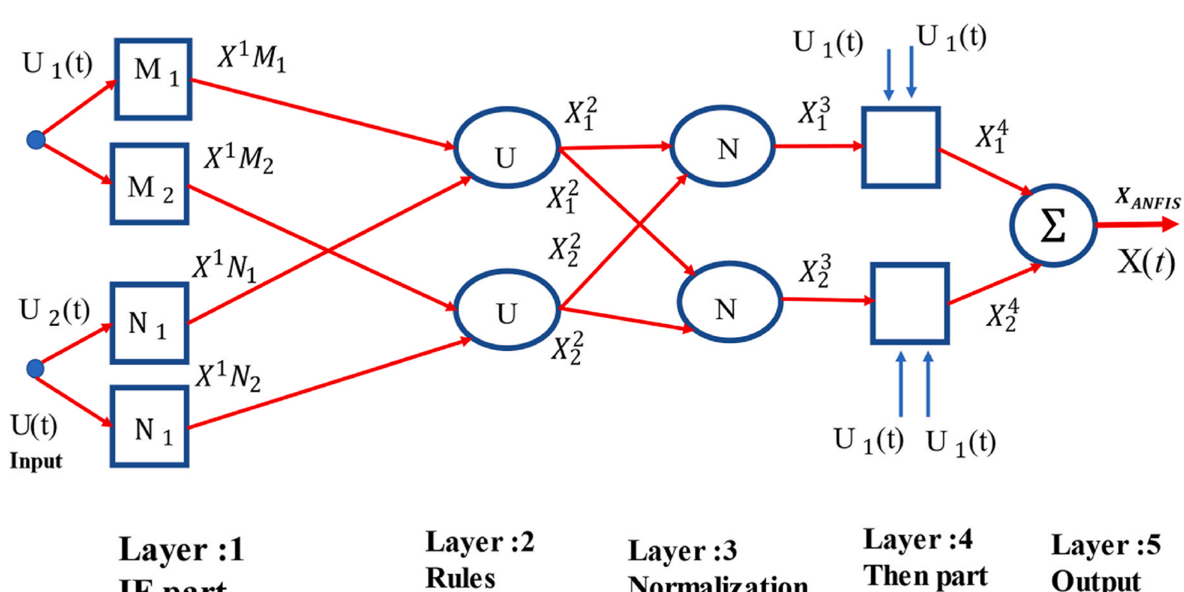


**Fig. 2.** Architecture of adaptive neuro-fuzzy process.

data, layers 2–3 do separate mathematical processes, layer 4 sends its signals to layer 5 and layer 5 adds them altogether. To illustrate the foundation of the ANFIS structure take the simple instance of a normal fuzzy inference with simply $U_1$ and $U_2$ as inputs and $X_{ANFIS}$ as output. Take the straightforward example of a regular fuzzy inference, where the inputs are just $U_1$ and $U_2$, the output is $X_{ANFIS}$, to demonstrate the underlying logic of the ANFIS structure. Fig. 2 depicts the ANFIS model is corresponding 5 layers. The notations are defined as IF < Premise>, THEN < Conclusion >. In this example two fuzzy IF-THEN rules for ANFIS for understanding purpose is presented [27]. According to the model requirements the rules shall be designed.

**Rule 1:** **IF** $U_1$ is $M_1$ and $U_2$ is $N_1$, THEN

$$f_1 = \alpha_1 \times U_1 + \beta_1 \times U_2 + C_1$$

**Rule 2:** **IF** $U_1$ is $M_2$ and $U_2$ is $N_2$, THEN

$$f_2 = \alpha_2 \times U_2 + \beta_2 \times U_2 + C_2$$

where, $\forall_i = \varepsilon\{1,2\}$, $M_i$ represents a fuzzy set for $U_1$ while $N_i$ represents a fuzzy set for $U_2$. At the training phase ANFIS models' parameters includes: $\alpha_i$, $\beta_i$, and $C_i$, $\forall_i = \varepsilon\{1,2\}$. A Gaussian fuzzy set is used in this example. The vector $p$ contains the input fuzzy set membership function parameters The elements in $p$ will include the means and standard deviations of the Gaussian membership $\sigma$ functions contained in $M_i$ and $N_i$ fuzzy sets, for instance, if $\forall_i = \varepsilon\{1,2\}$. In addition, it is not required that all inputs have the same number of fuzzy sets. As seen above, each fuzzy IF-THEN rule provides a fuzzy implication relation between certain input fuzzy sets (such as $M_i$ and $N_i$ in Rule 1) and their associated fuzzy outputs (such as $f_1$ in Rule 1), such that no two rules have the same premise portions. In this example, each input can accept two fuzzy sets, so the most fuzzy IF-THEN rules you can have are $2 \times 2 = 4$. In reality, the quest for the ideal ANFIS model along the fewest number of IF-THEN rules remains an active topic of study. In contrast, each fuzzy IF-THEN rule in ANFIS has a conclusion that is a first-degree polynomial $U$. IF-THEN rules' parameters, both for the conclusion and the premise's fuzzy sets, are included in the parameters.

In contrast to classical logic, the context of fuzziness defines the degree of membership of an element $U_i$ within a fuzzy set $\beta$ as a truth value between zero and one, denoted by $\varphi_\beta U_i$. Using the given notations and the two input signals $U_1$ and $U_2$, ANFIS starts by figuring out how much $U_1$ and $U_2$ belong to $M_i$ and $N_i$, respectively. According to $\varphi_\beta U_i = e^{\frac{-(U_i-m)^2}{2\sigma^2}}$ for a Gaussian membership function with the mean $m$ and the standard deviation σ. In addition, the number of the layer from which each intermediate output in Fig. 2 originates, is superscripted next to each such output. The generating node is shown by the subscript of the signal's output. Then, here are the outputs of layer 1:

$$X^1_{M_i} = \varphi_\beta M_i, \forall_i = \varepsilon\{1,2\} \tag{11}$$

$$X^1_{N_i} = \varphi_\beta N_i, \forall_i = \varepsilon\{1,2\} \tag{12}$$

where $\varphi_\beta M_i$ and $\varphi_\beta N_i$, $\forall_i = \varepsilon\{1,2\}$ are Gaussian membership functions the mean m and During the ANFIS training phase, the standard deviation of each of these functions is one of the things that the hybrid learning algorithm tunes. At the level of layer 1, the calculations are done. At the level of layer 2, the evaluation of the firing strengths per rule node starts

$$X^2_1 = X^1_{M_1} \times X^1_{N_1} \tag{13}$$

$$X^2_2 = X^1_{M_2} \times X^1_{N_2} \tag{14}$$

Then, layer 3 keeps going by adjusting the firing strengths so that:

$$X^3_1 = \frac{X^2_1}{\sum_{K=1}^{2} X^2_K} \tag{15}$$

$$X^3_2 = \frac{X^2_1}{\sum_{K=1}^{2} X^2_K} \tag{16}$$

Lastly, the outputs of layers 4 and 5 are figured out by using the following equations:

$$X^4_1 = X^3_1 \times f_1 = X^3_1(\alpha_1 \times U_1 + \beta_1 \times U_2 + C_1) \tag{17}$$

$$X^4_2 = X^3_2 \times f_2 = X^3_2(\alpha_2 \times U_1 + \beta_2 \times U_2 + C_2) \tag{18}$$

$$X_{ANFIS} = \sum_{K=1}^{2} X^4_2 \tag{19}$$

ANFIS models cannot be trained until the parameter vector is set up. Grid partitioning adds to the curse of dimensionality when an ANFIS model has many inputs, hence increasing the amount of fuzzy IF-THEN rules [28]. On the other hand, one of the best things about using FCM to initialise ANFIS [28,29] tends to make a small number of fuzzy IF-THEN rules.

## 4. Result and discussion

A benchmark model for identification is used to measure the effectiveness of the proposed method. In this section, a nonlinear system example is taken for identification and its performance is compared with the proposed model and the basic MLP model. The comparison between different models is done based on a differential matrix of Mean squared error (MSE) and Root Mean squared error, which is formulated as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(T_i - p_i)^2 \tag{20}$$

A differential equation with input and output relationships of a complex plant with relative degree −3 is given as [30]:

$$X(t+1)\frac{X(t)X(t-1)X(t-2)U(t-1)(X(t-2)-1)+U(t)}{1+X^2(t-2)+X^2(t-1)} \tag{21}$$

During training stage, the model considered an input having value uniformly distributed in an interval [−1,1]. 500 samples of data were generated using the plant model in simulation. With an input signal U at random and uniformly distributed in [−1.5, 1.5]. Five fuzzy rules are taken to construct the initial model of ANFIS using clustering the data $U \times X$. 100 learning epochs are used to train the model. The MATLAB Fuzzy Logic Toolbox (MATLAB, 2020) is used to generate the fuzzy rules:

If $U$ is $A'_1$ $THEN\ X = 4.12U + 0.035$
If $U$ is $A'_2$ $THEN\ X = 3.22U + 0.085$
If $U$ is $A'_3$ $THEN\ X = -3.27U + 1.053$
If $U$ is $A'_4$ $THEN\ X = 0.27U - 6.065$
If $U$ is $A'_5$ $THEN\ X = 5.27U + 0.065$

In total, the proposed rule extraction process has 5 rules, each having 5 Gaussian membership functions (MFs).

Fig. 3 represents the actual ARMA model for identification. The total number of samples are 1000, Among them 500 samples were used to train the model, and the remaining 500 samples were used for testing the model. After the training process of 100 epochs, the testing signal is used to determine the identification results.

$$X(t) = \begin{cases} 0.1\sin\left(\frac{2\pi n}{250}\right) + 0.5\cos\left(\frac{n}{25}\right) & if\ k \leqslant 300 \\ 0.02\sin\left(\frac{2\pi n}{25}\right) + 0.3\cos\left(\frac{2\pi n}{250}\right) & if\ 300 < k \leqslant 700 \\ 0.2\sin\left(\frac{2\pi n}{125}\right) & if\ 700 < k \leqslant 1000 \end{cases} \tag{22}$$

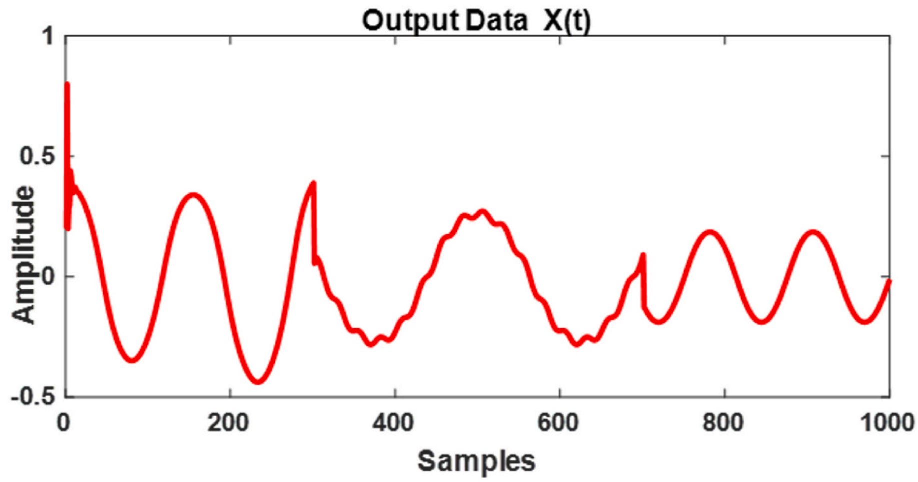


**Fig. 3.** Actual ARMA model.

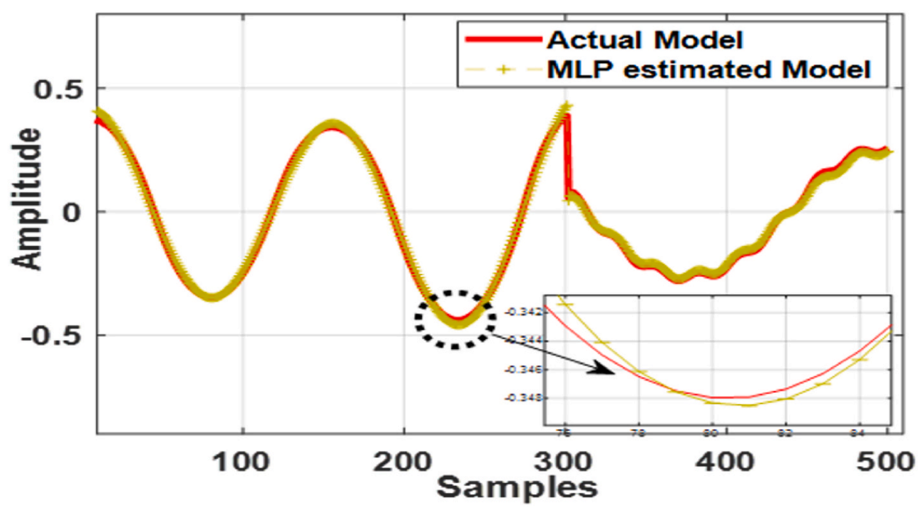


**Fig. 4.** Tracking of the MLP model over the actual model (Training output). The solid line represents the actual model and the dotted line represents the MLP estimated model.

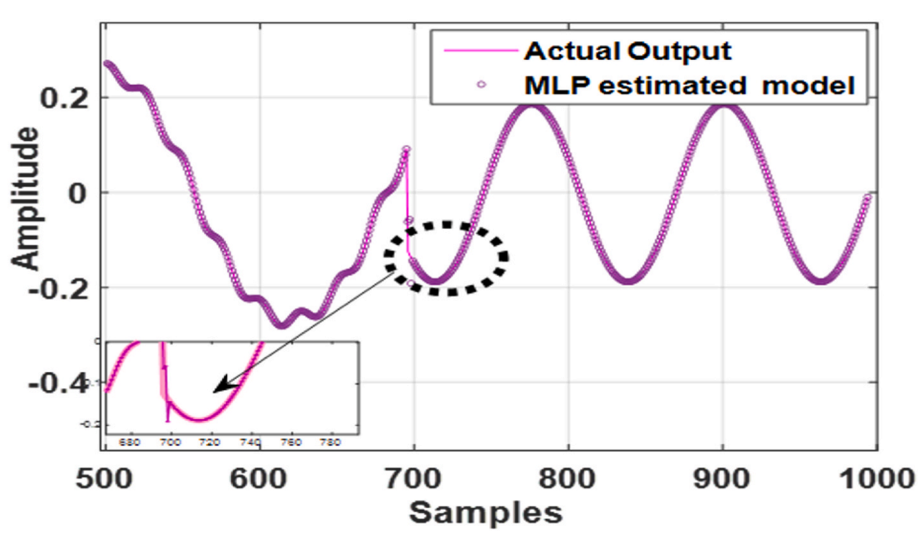


**Fig. 5.** Tracking of MLP model over actual model (Testing output). Circle represent the actual model and dotted line represent the MLP estimated model.

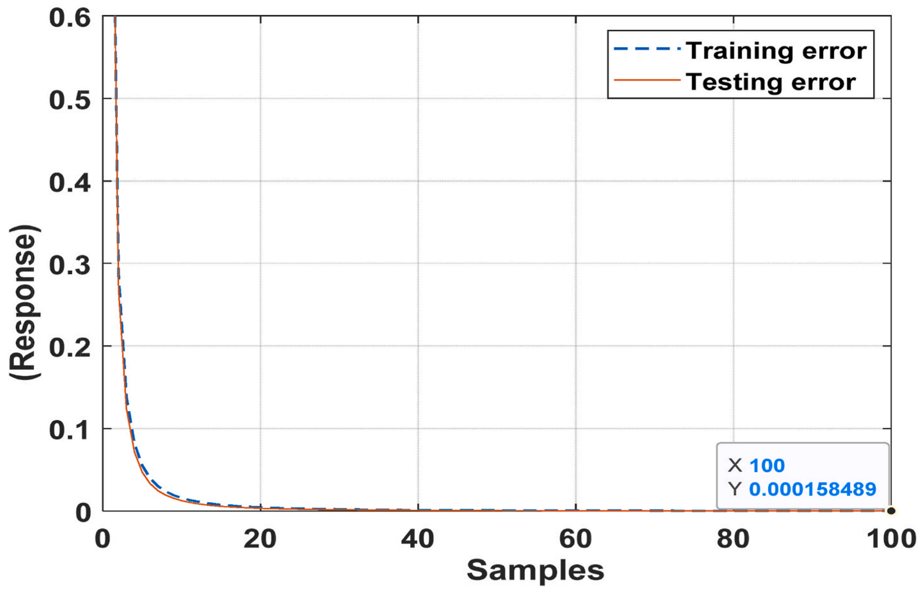


**Fig. 6.** Cost function of MLP model after Testing with max epoch of 100.

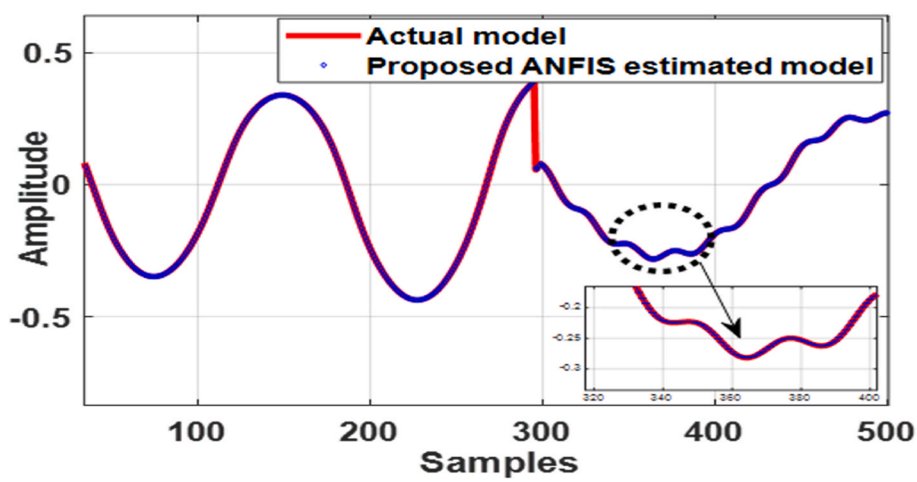


**Fig. 7.** Tracking of the ANFIS model over the actual model (Training output). The straight line represents the actual model and the diamond line represents the ANFIS estimated model.

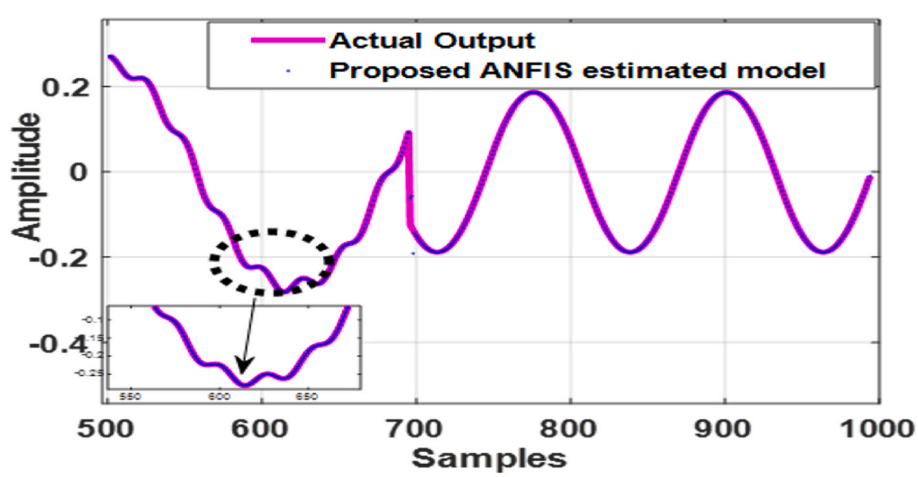


**Fig. 8.** Tracking of the ANFIS model over the actual model (Testing output). The straight line represents the actual model and the dotted line represents the ANFIS estimated model.

To verify the supremacy of the proposed model, the data was applied to multi-layered perceptron's. In the case when 40 epochs are used initially however the tracking result is not satisfactory, the number of epochs is increased linearly up to 100.Fig. 4 represents the identification of the actual model over the MLP model. The tracking of the two models is closer to each other.

Fig. 5 represents the testing output of the MLP model. At epoch 30, the model archives 0.0032 MSE at training and 0.0045 MSE at testing. The model performance also increased linearly after increasing the epoch size to 100. At epoch 100, the model achieve 0.00021 MSE as training and 0.00017 as testing MSE. The cost function of the model is depicted in Fig. 6 where the training and testing errors are plotted.

Fig. 7 represents the actual versus proposed ANFIS estimated model. The model is trained well, and the tracking of the two models is seen from the figure.

Fig. 8 represents the testing output of the proposed model. Five hundred samples were taken for model validation. The effectiveness of the model is evaluated after 20 training epochs are given. The model achieves 0.00061 MSE. The epoch size is extended to 100. After running for 100 iterations, the best result obtained is an MSE of 0.000012.

The overall comparison between the two models is depicted in Fig. 9. The mean squared error (MSE) of the proposed model is the closest it gets to being zero, as shown in the figure. The MLP model is an excellent training model, but the proposed model improves upon it by including adaptive learning approaches and efficient weight updates. Compared to other models, the proposed ANSFIS model's performance is shown in Table 1. Based on these results, the MLP model in Ref. [28] has a training error of 0.00056 and a testing error of 0.00026, while the RBF model in Ref. [29] has these values at 0.0037 and 0.0015, respectively. In its training phase, the NARAX model [31] records an MSE of 0.000082, and in its testing phase, it records an MSE of 0. 00043.The input and output parameters of the nonlinear system used for performance analysis of different models have also been use to verify and test the output of the proposed model. First, the MLP model requires use of the input and output data. Training MSE records at 0.00021 and testing MSE records at 0.000156. The ANFIS model is then given the input output data. The model achieves training MSE of 0.000039 and a testing MSE of 0.00012. In the ANFIS architecture, the first and fourth layers both have nodes that can change based on what they see. To optimise this parameter, one needs an algorithm for learning. There are two learning algorithms. Jang et al. [32] came up with hybrid learning. backpropagation, and algorithms. The least-squares method and the gradient descent method are both parts of the hybrid learning algorithm. This method has two steps: moving forward and moving backward. During the forward movement, the network input will move forward until the fourth layer. At that point, a least-squares method will be used to figure out the resulting parameters. In the backwards movement step, the error signal will be generated after computing the error. Backward propagation will fix the premise parameters by utilising a gradient descent technique.

## 5. Conclusion

A dynamic nonlinear WSN channel model based on the ARMA model is implemented in this work. The neural network model's dependence on the sensor model and network structure is analyzed. The objective of the work is to design an ANFIS neural network model to identify the parameters of the WSN channel. A standard benchmark ARMA nonlinear model is used to verify the model performance. To measure the effectiveness of a model, the mean squared error (MSE) is employed. The following table provides a comparison of different models such as the MLP, RBF, and NARAX. The ANFIS model achieve 0.000012 MSE in testing and 0.000039 MSE in training, according to the comparison table. The above analysis shows the performance of the proposed model is better compared to other models. In future the performance of more complex nonlinear plants would be analyzed.

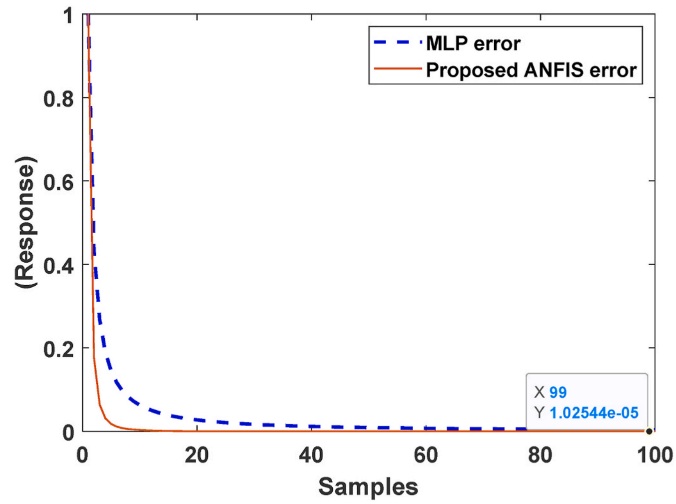


**Fig. 9.** Performance Analysis of the Models shows the MSE error between the two models.

**Table 1**
Performance evaluation of the proposed model.

| Sl. No. | MODEL | | MSE |
|---|---|---|---|
| 1 | MLP [28] | Train | 0.00056 |
| | | Test | 0.00026 |
| 2 | RBF Model [29] | Train | 0.0037 |
| | | Test | 0.0015 |
| 3 | NARAX [31] | Train | 0.000082 |
| | | Test | 0.00043 |
| 4 | **MLP** | Train | 0.00021 |
| | | Test | **0.000158** |
| 5 | **Proposed ANFIS** | Train | 0.000039 |
| | | Test | **0.000018** |

## CRediT authorship contribution statement

**Rakesh Kumar Pattanaik:** is the research scholar pursuing his Ph. D. work, He is working in the area of Signal Processing, So, his contribution in this paper is to execute the program and experiments with the guide. **Srikanta Kumar Mohapatra:** has verified with other algorithm. **Mihir Narayan Mohanty:** is supervising the scholar, His contribution in this work is to formulate the problem, Simultaneously observed the execution part of the work and finalised the paper from the work. **Binod Kumar Pattanayak:** has written the manuscript skeleton.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

No data was used for the research described in the article.

## References

[1] L. Xu, W. He, S. Li, Internet of Things in industries: a survey, IEEE Trans. Ind. Inf. 10 (4) (2014) 2233–2243.
[2] K. Guan, B. Ai, B. Peng, D. He, G. Li, J. Yang, Z. Zhong, and T. Kurner, "Towards realistic high-speed train channels at 5G millimeterwave Band-Part I: Paradigm, significance analysis, and scenario reconstruction,'' IEEE Trans. Veh. Technol, vol. 67, no. 10, pp. 9112-9128, 2018.
[3] T. Zhou, C. Tao, S. Salous, and L. Liu, "Geometry-based multilink channel modeling for high-speed train communication networks,'' IEEE Trans. Intell. Transp. Syst, vol. 21, no. 3, pp. 1229-1238, 2020.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Network. 38 (4) (2002) 393–422.
[5] S. Han, Identification in nonparametric models for dynamic treatment effects, J. Econom. 225 (2) (2021) 132–147.
[6] J. Man, M.D. Zielinski, D. Das, P. Wutthisirisart, K.S. Pasupathy, Improving non-invasive hemoglobin measurement accuracy using nonparametric models, J. Biomed. Inf. 126 (2022), 103975.
[7] H. Asplund A.A. Glazunov, J.-E. Berg, Statistical analysis of measured short-term impulse response functions of 1.88Ghz radio channels in Stockholm with corresponding channel model, in: *Proc. IEEE 50$^{th}$ Vehic. Tech. Conf. Fall*, 1999, pp. 107–111.
[8] J.-K. Hwang, J.H. Winters, Sinusoidal modelling and prediction of fast fading processes, in: Proc. IEEE Globecom, vol. 98, 1998, pp. 892–897.
[9] K.P. Rajput, A. Kumar, S. Srivastava, A.K. Jagannatham, L. Hanzo, Bayesian learning-based linear decentralized sparse parameter estimation in MIMO wireless sensor networks relying on imperfect CSI, IEEE Trans. Commun. 69 (9) (2021) 6236–6250.
[10] H.A. Alobaidy, M.J. Singh, M. Behjati, R. Nordin, N.F. Abdullah, Wireless transmissions, propagation and channel modelling for IoT technologies: applications and challenges, IEEE Access 10 (2022) 24095–24131.
[11] A.I. Moustapha, R.R. Selmic, Wireless sensor network modelling using modified recurrent neural networks: application to fault detection, IEEE Trans. Instrum. Meas. 57 (5) (2008) 981–988.
[12] R.A. do Prado, F. da Rocha Henriques, D.B. Haddad, Sparsity-aware distributed adaptive filtering algorithms for nonlinear system identification, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.
[13] K. Lee, S. Kim, K. You, Iterative regression-based hybrid localization for wireless sensor networks, Sensors 21 (no-1) (2021) 257.
[14] K.P. Rajput, M.F. Ahmed, N.K. Venkategowda, A.K. Jagannatham, G. Sharma, L. Hanzo, Robust decentralized and distributed estimation of a correlated parameter vector in MIMO-OFDM wireless sensor networks, IEEE Trans. Commun. 69 (10) (2021) 6894–6908.
[15] T. Olofsson, A. Ahlén, M. Gidlund, Modelling of the fading statistics of wireless sensor network channels in industrial environments, IEEE Trans. Signal Process. 64 (12) (2016) 3021–3034.
[16] M. Wei, K. Kim, Intrusion detection scheme using traffic prediction for wireless industrial networks, J. Commun. Network. 14 (3) (2012) 310–318.
[17] Q. Yu, L. Jibin, L. Jiang, An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks, Int. J. Distributed Sens. Netw. 12 (1) (2016), 9653230.
[18] S. Xu, B. Zeng, Network traffic prediction model based on auto-regressive moving average, J. Network. 9 (3) (2014) 653.
[19] S. Cho, J.W. Park, S.H. Sim, Decentralized system identification using stochastic subspace identification for wireless sensor networks, Sensors 15 (4) (2015) 8131–8145.
[20] S. Dash, M.N. Mohanty, Analysis of outliers in system identification using WLMS algorithm, in: International Conference on Computing, Electronics, and Electrical Technologies ICCEET, IEEE, 2012, pp. 802–806.
[21] S. Panda, M.N. Mohanty, Analysis of norms in adaptive algorithm on application of system identification, in: Information Systems Design and Intelligent Applications, Springer, 2016, pp. 579–588.
[22] S. Dash, M.N. Mohanty, Variable sign-sign Wilcoxon algorithm: a novel approach for system identification, International Journal of Electrical and Computer Engineering, IJECE 2 (4) (2012) 481–486.
[23] M.N. Mohanty, B. Sahu, P.K. Nayak, L.P. Mishra, Non-Linear dynamic system identification using FLLWNN with novel learning method, in: International Conference on Swarm, Evolutionary, and Memetic Computing, Springer, 2013, pp. 332–341.
[24] R.K. Pattanaik, M.N. Mohanty, Nonlinear system identification using robust fusion kernel-based radial basis function neural network, in: 2022 International Conference on Emerging Smart Computing and Informatics ESCI, IEEE, 2022, pp. 1–5.
[25] R.K. Pattanaik, B.K. Pattanayak, M.N. Mohanty, Use of multilayer recursive model for non-linear dynamic system identification, J. Stat. Manag. Syst. 1 (2022) 12.
[26] R. Kumar, S. Srivastava, A novel dynamic recurrent functional link neural network-based identification of nonlinear systems using Lyapunov stability analysis, Neural Comput. Appl. 33 (13) (2021) 7875–7892.
[27] M. Knaiber, L. Alawieh, Bayesian inference using an adaptive neuro-fuzzy inference system, Fuzzy Set Syst. (2022).
[28] M.B. Priestley, Spectral Analysis and Time Series: Univariate Series", vol. 1, Academic *press*, 1981.
[29] H. Liu, J. Chen, D. Hissel, H. Su, Short-term prognostics of pem fuel cells: a comparative and improvement study, IEEE Trans. Ind. Electron. 66 (8) (2018) 6077–6086.
[30] M. Rezakazemi, A. Dashti, M. Asghari, S. Shirazian, H2-selective mixed matrix membranes modelling using ANFIs, PSO- ANFIs, GA- ANFIs, Int. J. Hydrogen Energy 42 (22) (2017) 15211–15225.
[31] R. Kumar, S. Srivastava, A novel dynamic recurrent functional link neural network-based identification of nonlinear systems using Lyapunov stability analysis, Neural Comput. Appl. 33 (13) (2021) 7875–7892.
[32] J.M.P. Menezes Jr., G.A. Barreto, Long-term time series prediction with the Narx network: an empirical evaluation, Neurocomputing 71 (16–18) (2008) 3335–3343.