4-Tier Neural Network based Model for Reliable WSNs

Mamta Katiyar
ECE Department
M. M. Engineering College,
M. M. University
Mullana, Ambala, Haryana,
India-133207

H.P. Sinha, Ph.D.
ECE Department
M. M. Engineering College,
M. M. University
Mullana, Ambala, Haryana,
India-133207

Dushyant Gupta, Ph.D.
Electronics Department,
University College,
KUK
Kurukshetra, Haryana,
India-13611

ABSTRACT

Fault tolerance is one of the attractive inherent features of neural networks. This feature facilitates to retrieve the information of interest despite of corrupted signal in presence of faults in the network. This paper presents a reliable and fault tolerant model for reliable transportation of information in Wireless Sensor Networks. Work presented in this chapter is an attempt to make the senor nodes intelligent enough to deliver the information of interest to the base station despite of the fact that wireless communication medium is noisy and information may be corrupted during transportation. Algorithm supporting the functionality of the proposed model has been analyzed through the example.

Keywords

ANN, BAM, NN, WSN.

1. INTRODUCTION

A Neural Network (NN) is a system consisting of processing components called neurons connected in parallel or distributed manner in a graph topology. In NN, neurons are connected with each other through weights called synaptic strength of connections. Weights connect the network input layer to output layer. Knowledge of NN is stored in its connections and there is no requirement of any additional storage. One of the difficulties with NNs is choosing of appropriate topology for the problem. Different methods for training the neural networks are inspired from biology science which determines the way NNs learn. In all most all of these networks, training is based on learning by example. In the problem definition, set of samples defining the correct input-output data are often given to the network and using these examples, the network weights are adjusted accordingly to produce the desired output for respective input patterns. The most important feature of a NN is its ability to recognize the data affected by noise or intentional change and to remove those variations after training.

Neural networks are efficiently used in WSNs for optimal use of energy of sensor nodes. Requirements of WSNs can be easily satisfied by the algorithms developed in neural network in application areas like parallel distributed computation, distributed storage, data robustness, fault tolerance and low computation. In a WSN platform, neural networks can help through dimensionality reduction, obtained simply from the outputs of the neural networks clustering algorithms, leads to lower communication cost and energy savings [1]. The other reason to use neural network based methods in WSNs is the similarity in the behaviour of WSNs and ANNs. Authors in [2] shown that ANNs exhibit exactly the same architecture as WSNs where behaviour of neurons is similar to sensor nodes and connections correspond to radio links. With this opinion, the whole sensor network can be seen as a neural network and within each sensor node inside the WSN.

2. ISSUES

In WSN, a packet may not be successfully transmitted to the destination because of the following reasons: (a) Probable forwarders inside the coverage are in good working condition but have not received the packet correctly because of noise etc. and (b) All the forwarders are in failure condition.

Our proposed model is an attempt to still improve the fault tolerance and reliability of packet delivery process in WSN. This model addresses the first reason of unsuccessful packet delivery in WSNs. Proposed model is well suited for the applications where the events that can occur in the network are from a predefined set of events. Assume that sensor node has to sense an event to receive a packet from a predefined set P = $\{p_1, p_2, p_3, p_4, p_5, \ldots,\}$; where the size of each packet is m-bit. Larger is the size of a packet, more will be the chances of faulty delivery of packet to the next probable forwarder/destination because of noise etc. Authors in [3] shown that if a packet is compressed before transmission, we can take the advantage of compression ratio but the price one may have to pay will be distortion ratio.

Our proposed model is based on encoding the associations using neural network. For each message (packet) sensed by a sensor node, a small packet (vector) associated to the original packet is forwarded towards the destination as we know that transmitting an associated vector of small size has better reliability and fault tolerance.

Authors in [4][5][6][7] presented models where an input pattern is presented to a memory and then the corresponding output pattern is presented to the transpose memory. After that the output is feed the memory and so on, until a stable state is reached. Most of these models work with bipolar and binary patterns. These models (for example Bidirectional Associative Memory) are not able to guarantee the recall of all patterns.

3. SYSTEM MODEL

Proposed 4-Tier neural network based model is four layered architecture as shown in Fig. 1. In this proposed model, 4-Layered neural network is embedded in each sensor of WSN. Signal sensed by the sensor node is presented at first layer of the neural network before transmitting to the next forwarder. Second layer is an auto associative Hopfield neural network and the input sensed at the first layer is applied to the second layer through the identity matrix and remains available on this layer until network gets stabilized. Output generated from the Hopfield neural net at the second layer is applied on the weight matrix W between layer second and third to encode the association between the two hetro associative vectors (signals) A and B where size of vector B is smaller than A.

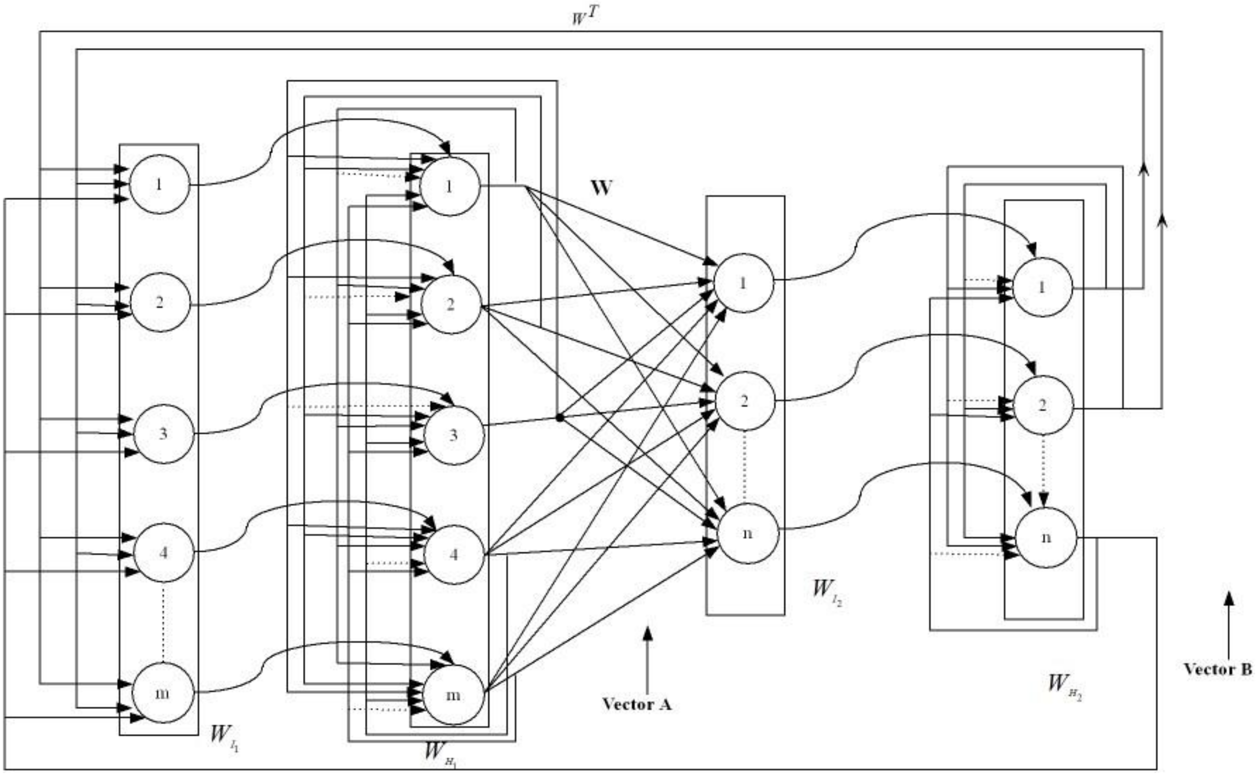

Fig 1: Four Tier Network Structure for Improved Reliability / Fault Tolerance of Data Transportation in WSN

Encoded signal generated at third layer is applied to the Hopfield neural network at forth layer and it remains available at this layer until this encoded signal gets stabilized. Now finally this encoded signal is sent to the next forwarder for onwards transmission towards the sink.

Hopfield neural networks at the second and forth layers are used to improve the inherent fault tolerance of the network where partially corrupted pattern is recovered bit by bit to strengthen forward association A→B. Functionality of second and third layer together is same as of Bi-Directional Memory (BAM). The output of the second and forth layers is updated in asynchronous manner (i.e. one bit is updated at one time) and thus partially corrupted noisy signal is updated first and then it is sent to the subsequent layers for recalling the associated pattern.

3.1 Notations Used

$$W_{m*n} = \sum_i (A'_i)^T (B'_i);$$

$$W^T_{n*m} = (W_{m*n})^T$$

$$\left.\begin{aligned} W_{I_i} &= [W_{ij}]_{m*m} \quad Where \left.\begin{matrix} W_{ij} = 1 \ When\ i = j \\ W_{ij} = 0 \ Otherwise \end{matrix}\right\rangle \\ W_{I_2} &= [W_{ij}]_{n*n} \quad Where \left.\begin{matrix} W_{ij} = 1 \ When\ i = j \\ W_{ij} = 0 \ Otherwise \end{matrix}\right\rangle \end{aligned}\right\} \begin{matrix} Identity\ matrix\ to\ copy\ bit\ by\ bit \\ data\ as\ input\ in\ subsequent\ layer\ neuron. \end{matrix}$$

$$W_{H_i} = [W_{ij}]_{m*m} = \sum_i (A'_i)^T (A'_i); \quad Where \left.\begin{matrix} W_{ij} = 0 \ When\ i = j \\ and\ W_{ij} = W_{ji} \end{matrix}\right\rangle$$

$$W_{H_2} = [W_{ij}]_{n*n} = \sum_i (B'_i)^T (B'_i); \quad Where \left.\begin{matrix} W_{ij} = 0 \ When\ i = j \\ and\ W_{ij} = W_{ji} \end{matrix}\right\rangle$$

Where A'_i, B'_i are bipolar representation of vectors A_i and B_i respectively and $(\mathrm{A}'_\mathrm{i})^T, (B'_i)^T$ are the transpose of vectors $\mathrm{A}'_\mathrm{i}\ and\ B'_i$.

3.2 Training Procedure

Activation functions used in the proposed 4-Tier neural network are as under:

A. For auto associative networks at layer second and forth:

$$\begin{aligned} \mathrm{Out}_\mathrm{j} = f(Net_j) &= 1;\ if\ \ Net_j\ \ is\ \ positive \\ &= 0;\ if\ \ Net_j\ \ is\ negative \\ &= previous\ output;\ if\ \ Net_j = 0 \end{aligned}$$

Where $\mathrm{N}et_\mathrm{j} = a_i + \sum_j Out_j W_{ji}$ and $\mathrm{a_i}$ is the $\mathrm{i^{th}}$ bit of input vector

B. For Bi-directional memory between layer-2 and layer-3
(Forward Association):

$$\begin{aligned} \mathrm{Out} = f(Net) &= 1;\ if\ \ Net\ is\ positive \\ &= 0;\ if\ \ Net\ is\ negative \\ &= previous\ output;\ if\ \ Net = 0 \end{aligned}$$

Net value is computed as under:

$$\mathrm{N}et = A.W_{m*n}$$

Where A: input signal sensed by the sensor node and stabilized after passing through the auto associative neural network at layer-2 and

$$W_{m*n} = \sum_i (A_i')^T.(B_i')$$

Forward Association:
Forward Association A→B is used at the source node sensing the event of interest.

Backward Association:
Backward association B→A is used at the base station to retrieve the actual event sent by the source node once encoded signal is received at the base station. Activation function used at the base station will be as under:

$$\begin{aligned} \mathrm{A} = \mathrm{Out} = f(Net) &= 1;\ if\ \ Net\ is\ positive \\ &= 0;\ if\ \ Net\ is\ negative \\ &= previous\ output;\ if\ \ Net = 0 \end{aligned}$$

Where $\mathrm{N}et = B.(W_{m*n})^T$ and $(W_{m*n})^T$ is the transpose of W_{m*n} .

3.3 Application Algorithm

Step1: Encode the associations for predefined event space:

$W_{m*n} = \sum_i (A_i')^T.B_i'$; for forward associations in BAM

$W_{mXn}^T = \sum_i (B_i')^T.A_i'$; for backward associations in BAM

$$W_{H_1} = \sum_i (A_i')^T.A_i';\ For\ auto\ associative\ network\ at\ layer-2\ Where\ w_{ii} = 0$$

$$W_{H_2} = \sum_i (B_i')^T.B_i';\ For\ auto\ associative\ network\ at\ layer\ -4\ Where\ w_{ii} = 0$$

Step 2: Initialization of identity matrix to facilitate copy operation to the subsequent auto associative network layer.

$$W_{I_1} = [I]_{m*m}\ and\ W_{I_2} = [I]_{n*n}$$

Step 3: Set the initial activation of layer-1 to the external input vector A.

$$Out_{layer-1} = A$$

While activation of the auto associative network at layer-2 are not converged do:

Step 4: Set the initial activation of layer-2 to the output of layer-1

$$[Out_i]_{layer-2} = A.I = a_i$$

Step 5: Do steps 6-8 for each unit out$_i$ at layer-2

Step 6: Compute net input:

$$net_i = a_i + \sum_j out_j.[W_{H_1}]_{ji}$$

Step 7: Determine activation (output):

$$[Out_i]_{layer-2} = \begin{cases} 1;\ if\ \ net_i\ \ is\ positive \\ 0;\ if\ \ net_i\ \ is\ negative \\ previous\ output;\ if\ \ net_i = 0 \end{cases}$$

Step 8: Broadcast the value of $[Out_i]_{layer-2}$ to all other units except itself.

Step 9: Test for convergence:

```
    If auto associative network at layer-2 stabilized
        then go to step 10;
    else
        go to step 5.
```

Step 10: Apply activation of layer-2 on the weight matrix W.

Compute the net input at layer-3

$$net = [out]_{layer-2}.W = A.W$$

Step 11: Determine the activation (output) of layer-3.

$$[Out_i]_{layer-3} = b_i = \begin{cases} 1;\ if\ \ net\ is\ positive \\ 0;\ if\ \ net\ is\ negative \\ previous\ output;\ if\ \ net = 0 \end{cases}$$

Step 12: Set the initial activation of layer-4 to the output of layer-3. $[out_i]_{layer-4} = B.I = b_i$

Step 13: Do steps 14-16 for each unit $[Out_i]_{layer-4}$.

Step 14: Compute net input:

$$[net_i]_{layer-4} = b_i + \sum_j [out_i]_{layer-4}.[W_{H_2}]_{ji}$$

Step 15: Determine the activation (output):

$$B = [Out_i]_{layer-4} = \begin{cases} 1;\ if\ \ [net_i]_{layer-4} is\ \ positive \\ 0;\ if\ \ [net_i]_{layer-4}\ \ is\ negative \\ previous\ output;\ if\ \ [net_i]_{layer-4} = 0 \end{cases}$$

Step 16: Broadcast the value of $[Out_i]_{layer-4}$ to all other units except itself.

Step 17: Test for convergence:

```
    If auto associative network at layer-4 stabilized
        then go to step 18;
    else
        go to step 13.
```

Step 18: Source node is ready to transmit the encoded signal to the next forwarder towards the base station.

At the base station, output of layer-4 is applied on W^T To recall the actual signal A.

$$A = f[net] = f(B.W^T) = \begin{cases} 1;\ if\ \ net\ \ is\ \ positive \\ 0;\ if\ \ net\ is\ negative \\ previous\ output;\ if\ \ net = 0 \end{cases}$$

4. SYSTEM ANALYSIS

This section presents the analysis of proposed model to show the functionality and performance of WSN in presence of noise. As wireless communication medium is error prone, two reasons have been considered in this analysis for erroneous delivery of signal at the base station as under:

i. Error in the signal during the transmission due to mistaken bits (altered bits).
ii. Error in the signal during transmission due to missing bits (unrecognized bits).

Functionality and performance of the proposed model has been illustrated through example as under considering bipolar representation of signals sensed by the sensor node (source node) for training the proposed neural network embedded in each sensor to differentiate between missing and mistaken bits and hence to differentiate between the two kind of errors during transmission in WSN. Analysis shows that proposed model has better fault tolerance for the transmission due to missing bits over the mistaken bits.

Example:
Consider the event space with the two samples A_1 and A_2 to be sensed by the sensor nodes and to be sent to the base station in a WSN as under:

$$A_1 = [1001011001101001]$$

$$A_2 = [1110001100111110]$$

As we know that event smaller in size has better fault tolerance during transmission, we have associated vectors B_1 and B_2 respectively with events A_1 and A_2 for reliable data transfer over the WSN. Associations are encoded and memorized in the connections (weights) of the neural network as under:

4.1.1 Encoding the Associations before Transmission:

Input sample spaces   associated vector for onward transmission

$$A_1 = [1001011001101001] \qquad B_1 = [100111100]$$

$$A_2 = [1110001100111110] \qquad B_2 = [111100111]$$

Bipolar versions:

$$A'_1 = [1\text{-}1\text{-}11\text{-}111\text{-}1\text{-}111\text{-}11\text{-}1\text{-}11]$$

$$B'_1 = [1\text{-}1\text{-}11111\text{-}1\text{-}1]$$

$$A'_2 = [111\text{-}1\text{-}1\text{-}111\text{-}1\text{-}111111\text{-}1] \qquad B'_2 = [1111\text{-}1\text{-}1111]$$

Weight matrix $W = W_{m*n} = \sum_i (A'_i)^T (B'_i);$

$$W = \begin{bmatrix} 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 2 & 0 & -2 & -2 & 0 & 2 & 2 \\ 0 & 2 & 2 & 0 & -2 & -2 & 0 & 2 & 2 \\ 0 & -2 & -2 & 0 & 2 & 2 & 0 & -2 & -2 \\ -2 & 0 & 0 & -2 & 0 & 0 & -2 & 0 & 0 \\ 0 & -2 & -2 & 0 & 2 & 2 & 0 & -2 & -2 \\ 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 2 & 0 & -2 & -2 & 0 & 2 & 2 \\ -2 & 0 & 0 & -2 & 0 & 0 & -2 & 0 & 0 \\ 0 & -2 & -2 & 0 & 2 & 2 & 0 & -2 & -2 \\ 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 2 & 0 & -2 & -2 & 0 & 2 & 2 \\ 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 2 & 0 & -2 & -2 & 0 & 2 & 2 \\ 0 & 2 & 2 & 0 & -2 & -2 & 0 & 2 & 2 \\ 0 & -2 & -2 & 0 & 2 & 2 & 0 & -2 & -2 \end{bmatrix}$$

$$W^T_{n*m} = (W_{m*n})^T =$$

$$W = \begin{bmatrix} 2 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & 2 & 2 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & 2 & 2 & -2 \\ 2 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & -2 & -2 & 2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & -2 & -2 & 2 \\ 0 & -2 & -2 & 2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & -2 & -2 & 2 \\ 2 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & 2 & 2 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 & 0 & 2 & 2 & -2 \end{bmatrix}$$

$$W_{H_2} = [W_{ij}]_{n*n} = \sum_i (B'_i)^T (B'_i); \quad Where\, W_{ii} = 0$$

$$W_{H_2} = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{bmatrix} +$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & -2 & -2 & 0 & 2 & 2 \\ 0 & 2 & 0 & 0 & -2 & -2 & 0 & 2 & 2 \\ 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & -2 & -2 & 0 & 0 & 2 & 0 & -2 & -2 \\ 0 & -2 & -2 & 0 & 2 & 0 & 0 & -2 & -2 \\ 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & -2 & -2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 & -2 & -2 & 0 & 2 & 0 \end{bmatrix}$$

Similarly

$$W_{H_i} = [W_{ij}]_{m*m} = \sum_i (A'_i)^T (A'_i); \quad Where\, W_{ii} = 0$$

4.1.2 Recalling the Actual Signal at the base Station from Encoded Vector:

Case-1: No Error during the transmission of encoded signal from source to destination (base station)-

Consider that the event $A'_1 = [1\text{-}1\text{-}11\text{-}111\text{-}1\text{-}111\text{-}11\text{-}1\text{-}11]$ is detected at the sensor node (source node) for onward transmission to the base station. Before transmission, it is applied on weight matrix W and related associated pattern B_1 is generated for onward transmission as under:

Net at layer-3 = $A'_1 * \mathrm{W}$

= [+ive, -ive, -ive, +ive, +ive, +ive, +ive, -ive, -ive]

Out at layer-3 = [1 0 0 1 1 1 1 0 0] = B_1

When vector B_1 is received at the base station, it is applied on W^T to generate original signal A_1 as under:

Net= B_1*W^T= [+ive, -ive, -ive, +ive, -ive, +ive, +ive, -ive, -ive +ive, +ive, -ive, +ive, -ive, -ive, +ive]

Out = [1 0 0 1 0 1 1 0 0 1 1 0 1 0 01]= A_1 ; Actual signal recalled.

Case-2: Error encounters during the transmission from source node to the destination (Base Station)-

(i) **Fault tolerance of the proposed model in case error occurs due to mistaken bits:-**

Recovery of one bit fault: Consider that signal received at the base station (at layer-4 of neural network) encounters **one bit fault (first digit as mistaken bit)** during transmission.

i.e. B_1 = [-**1** -1 -1 1 1 1 1 -1 -1].

Base station applies this vector on W^T to retrieve the original signal associated with this received signal as under:

Net= $B_1 * W^T$

= [+ive, -ive, -ive, +ive, -ive, +ive, +ive, -ive, -ive +ive, +ive, -ive, +ive, -ive, -ive, +ive]

Out = [1 0 0 1 0 1 1 0 0 1 1 0 1 0 01] = $A_{1;}$ recall the original signal.

Recovery of three bit fault: Consider that signal received at the base station (at layer-4 of neural network) encounters **three faulty bits (first 3-digits as mistaken bit)** during transmission.

i.e. B_1 = [-**1 1 1** 1 1 1 1 -1 -1].

Base station applies this vector on W^T to retrieve the original signal associated with this received signal as under:

Net= $B_1 * W^T$

= [+ive, -ive, -ive, +ive, -ive, +ive, +ive, -ive, -ive +ive, +ive, -ive, +ive, -ive, -ive, +ive]

Out = [1 0 0 1 0 1 1 0 0 1 1 0 1 0 01] = $A_{1;}$ recall the original signal.

Recovery of four bit fault: Consider that signal received at the base station (at layer-4 of neural network) encounters **4-faulty bits (first 4-digits as mistaken bit)** during transmission.

i.e. B_1 = [-**1 1 1 -1** 1 1 1 -1 -1].

Base station applies this vector on W^T to retrieve the original signal associated with this received signal as under:

Out # $A_{1;}$ failure to recover the original signal.

(ii). **Fault tolerance of the proposed model in case error occurs due to missing bits:-**

Recovery of three bit fault: Consider that signal received at the base station (at layer-4 of neural network) encounters **three bit fault (2nd, 3rd and 7th digits are missing)** during transmission.

i.e. B_1 = [1 **0 0** 1 1 1 **0** -1 -1].

Base station applies this vector on W^T to retrieve the original signal associated with this received signal as under:

Net= $B_1 * W^T$

= [+ive, -ive, -ive, +ive, -ive, +ive, +ive, -ive, -ive +ive, +ive, -ive, +ive, -ive, -ive, +ive]

Out = [1 0 0 1 0 1 1 0 0 1 1 0 1 0 01] = $A_{1;}$ recall the original signal.

Recovery of four bit fault: Consider that signal received at the base station (at layer-4 of neural network) encounters **four bit fault (2nd, 3rd, 7th and 8th digits are missing)** during transmission.

i.e. B_1 = [1 **0 0** 1 1 1 **0 0** -1].

Base station applies this vector on W^T to retrieve the original signal associated with this received signal as under:

Net= $B_1 * W^T$

= [+ive, -ive, -ive, +ive, -ive, +ive, +ive, -ive, -ive +ive, +ive, -ive, +ive, -ive, -ive, +ive]

Out = [1 0 0 1 0 1 1 0 0 1 1 0 1 0 01] = $A_{1;}$ recall the original signal.

Recovery of five bit fault: Consider that signal received at the base station (at layer-4 of neural network) encounters **five bit fault (2nd, 3rd , 7th ,8th and 9th digits are missing)** during transmission.

i.e. B_1 = [1 **0 0** 1 1 1 **0 0 0**].

Base station applies this vector on W^T to retrieve the original signal associated with this received signal as under:

Net= $B_1 * W^T$

= [+ive, -ive, -ive, +ive, -ive, +ive, +ive, -ive, -ive +ive, +ive, -ive, +ive, -ive, -ive, +ive]

Out = [1 0 0 1 0 1 1 0 0 1 1 0 1 0 01] = $A_{1;}$ recall the original signal.

Recovery of six bit fault: Consider that signal received at the base station (at layer-4 of neural network) encounters **six bit fault (2nd, 3rd ,6th, 7th ,8th and 9th digits are missing)** during transmission.

i.e. B_1 = [1 **0 0** 1 1 **0 0 0 0**].

Base station applies this vector on W^T to retrieve the original signal associated with this received signal as under:

Net= $B_1 * W^T$

= [+ive, -ive, -ive, +ive, -ive, +ive, +ive, -ive, -ive +ive, +ive, -ive, +ive, -ive, -ive, +ive]

Out = [1 0 0 1 0 1 1 0 0 1 1 0 1 0 01] = $A_{1;}$ recall the original signal.

Recovery of seven bit fault: Consider that signal received at the base station (at layer-4 of neural network) encounters **five bit fault** (**2nd, 3rd, 5th, 6th, 7th, 8th and 9th digits are missing)** during transmission.

i.e. B_1 = [1 **0 0** 1 **0 0 0 0 0**].

Base station applies this vector on W^T to retrieve the original signal associated with this received signal as under:

Net= $B_1 * W^T$

Out # $A_{1;}$ failure to recover the original signal.

5. RESULTS AND DISCUSSION

Application algorithm discussed in this chapter demonstrated through the example in support of the reliable communication is significance of improvement in the fault tolerance power of the network as the encoded signal is recovered in asynchronous manner (bit by bit recovery) before transmitting it to the next probable forwarder. Analysis of the 4-tired neural based model shows that network is capable to produce the desired signal despite of erroneous communication channel once the associations are encoded and stored in the neural network upto its maximum storage capacity. Analysis shows that there is significant improvement in the recovery of the corrupted signal if error occurs due to missing bits than that of mistaken bits. Proposed 4-tier model is useful for successful delivery of information with minimum consumption of energy as associated packet transmitted over the wireless communication channel is smaller in size and moreover transmitting a large packet consumes more energy than the energy used in computations involved in encoding the associations and in retrieval of the original event at the base station.

6. REFERENCES

[1] A. Kulakov, D. Davcev, G. Trajkovski, “Application of wavelet neural-networks in wireless sensor networks”, Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05), pp.262—267, 2005

[2] Oldewurtel, Frank, Mahonen, Petri, “Neural Wireless Sensor Networks”, International Conference on systems & Networks Communications, ICSNS’06, pp. 28-28 .

[3] Vijay Kumar, R. B. Patel, Manpreet Singh and Rohit Vaid, “ A Neural Approach for Reliable and Fault Tolerant Wireless Sensor Networks”, in International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 2(5), pp. 113-118, 2011.

[4] Y. F. Wang, “Guaranteed Record of all Training Pairs for Bidirectional Associative Memory ,” in IEEE Transactions on Neural Networks , vol. 1(6), pp. 559-567, 1991.

[5] L. Lee, W. J. Wangl, “Improvement of Bidirectional Associative Memories by using Correlation Significance”, in Electronics Letters, vol. 29(8), pp. 688-690, 1993.

[6] C. L. Leung, “Optimum Learning for Bidirectional Associative Memory in the sense of Capacity”, in IEEE Transaction on Systems, Man and Cybernetics, vol. 24(5), pp. 791-796, 1994.

[7] S. Chartier, M. Bouladoum, “A Bidirectional Hetero Associative Memory for Binary and Gray Level Patterns”, IEEE Transactions on Neural Networks , vol. 17(2), pp. 385-396, 2006.