Application of Neural Network and Wavelet Transform Techniques in Structural Health Monitoring

Mahabubur Rahman

A Thesis

in

The Department

of

Building, Civil & Environmental Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Civil Engineering) at
Concordia University
Montreal, Quebec, Canada

September 2009

Your file Votre référence
ISBN: 978-0-494-63075-4
Our file Notre référence
ISBN: 978-0-494-63075-4

ABSTRACT

Application of Neural Network and Wavelet Transform Techniques in Structural Health Monitoring

Mahabubur Rahman

Structural Health Monitoring (SHM) has recently emerged as a useful tool for tracking the performance parameters of a structure such as strain, deflection, and acceleration through a series of sensors installed on them. The signals produced from these sensors are the main performance indicator of the structure. In assessing the condition of the structure, the proper analysis and the evaluation of changes in pattern of signals are the most important tasks in SHM. Another important aspect of SHM is the detection of the defective sensors. And it is very difficult to identify it manually from a series of sensors. Although it is an important task in SHM but no straightforward method exists currently to carry out this task. In this study, the sensor data from a Canadian bridge have been utilized here to develop Artificial Neural Network (ANN) and Wavelet Transform (WT) based methods for tracking the changes in sensor data pattern and detecting the defective sensors in SHM. The ANN structures are constructed with input nodes accepting data from selected strain gauges and a target selected from the remaining strain gauges. The data collected at different time periods are de-noised by WT and tested against the trained network to find the pattern of differences between the input and output data series. The proposed methods have been validated with the available data and are found to be effective in tracking the data patterns and detecting defective sensors.

ACKNOWLEDGEMENT

Firstly, I would like to thank my thesis supervisor Dr. Ashutosh Bagchi for his excellent guidance and directions throughout the thesis work. Without his advice and active support, it could have been a difficult task to organize and complete this thesis. His regular advice, enormous knowledge, expertise and directions on the development of the main idea have been effective towards the success of this thesis. Also he encouraged me by sparing his busy scheduled time and responding emails that helped to achieve the final target of this thesis.

Secondly, special thanks to Dr. Kinh H. Ha for his assistance and advice to accomplish this task.

Also I thank the authority of Concordia University and the librarians for their cooperation.

Besides this, I owed a great deal to my better half-Maksuda, daughters- Fariha, Maliha and Sara for their continuous moral support. And to my brother, friends and colleagues for the inspirations I received from them.

Finally, I would like to dedicate the whole work to the sweet memories of my father.

Table of Contents

List of figures

List of tables

List of symbols

$p =$ Network input vector

$t =$ Network target vector

$w =$ Network weight matrix

$n =$ Net input

$b =$ Bias of the network

$a =$ Network output

$x_k =$ Vector of current weight

$g_k =$ Current gradient

$\alpha_k =$ Learning rate

$\varepsilon =$ Strain

$P =$ Static load

$f'_c =$ Compressive strength of concrete

$A =$ Area

$E =$ Modulus of Elasticity

$db4 =$ Daubechies family of order four

List of Abbreviations

ANN = Artificial Neural Network

BMS = Bridge Monitoring System

BP = Back Propagation

CC = Correlation Coefficient

DLA = Dynamic Load Allowance

DAS = Data Acquisition System

EEG = Electroencephalography

ECG = Electrocardiography

FRP = Fiber Reinforced Polymer

FOS = Fiber Optic Sensor

FFT = Fast Fourier Transform

GFRP = Glass Fiber Reinforced Polymer

ISIS = Intelligent Sensing of Innovative Structures

MSE = Mean Squared Error

MTQ = Ministry of Transportation, Quebec

MR & R =Maintenance, Rehabilitation and Repair

SHM = Structural Health Monitoring

VBDD = Vibration Based Damage Detection

WT = Wavelet Transform

SG1_1 = Sensor Gauge-1 in direction -1 (1= Vertical)

SG1_2 = Sensor Gauge-1 in direction-2 (2= Horizontal)

SG2_1 = Sensor Gauge-2 in direction-1 (1 = Vertical)

SG2_2 = Sensor Gauge-2 in direction-2 (2 = Horizontal)

SG3_1 = Sensor Gauge-3 in direction-1 (1 = Vertical)

SG3_2 = Sensor Gauge-3 in direction-2 (2 = Horizontal)

SG4_1 = Sensor Gauge-4 in direction-1 (1 = Vertical)

SG4_2 = Sensor Gauge-4 in direction-2 (2 = Horizontal)

SG5_1 = Sensor Gauge-5 in direction-1 (1 = Vertical)

SG5_2 = Sensor Gauge-5 in direction-2 (2 = Horizontal)

SG6_1 = Sensor Gauge-6 in direction-1

(1 = Vertical)

SG6_2 = Sensor Gauge-6 in direction-2

(2 = Horizontal)

SG7_1 = Sensor Gauge-7 in direction-1

(1 = Vertical)

SG7_2 = Sensor Gauge-4 in direction-2

(2 = Horizontal)

SG8_1 = Sensor Gauge-8 in direction-1

(1 = Vertical)

SG8_2 = Sensor Gauge-8 in direction-2

(2 = Horizontal)

CHAPTER-1

Introduction

1.1 Background and Motivation

Structural Health Monitoring (SHM) is an important tool for condition assessment of a structure which can produce necessary information to alert us in case of any distress in a structure. It also helps us in taking immediate preventive measures to avoid any catastrophe that ultimately protects human lives and reduces huge financial losses. In addition, structural health monitoring ensures the smooth operation of the structure throughout its service life by detecting damages at its onset and taking instant actions to repair those damages. There are various diagnostic methods of structural damage in SHM. Some of them are data driven and the others are model based. A large volume of published work is available on damage detection techniques employing both the methods as mentioned above. But still there are some uncertainties in terms of accuracy and reliability of these methods.

One of the recent model based techniques is the vibration based damage detection method. In this method normally a model is needed to be developed to correlate damage in the structure with the changes in its dynamic properties such as natural frequencies and mode shapes (Fang et al., 2005). Kim and Stubs, 2003 developed a crack location /size model and studied the crack effect on natural frequencies which was in turn used to decide crack occurrence; Khiem and Lien, 2002 developed a dynamic stiffness matrix method for spectral analysis of forced vibration of a beam with multiple-damage by applying the equivalent rotational spring model of crack; Shi et al., 2000 developed the incomplete mode shapes for detection and localization of cracks. Bagchi et al., 2009 used the model based techniques for damage detection in a bridge using vibration properties. There are many examples on model based methods in SHM. The main idea of this method is the reference or baseline dynamic response of an intact structure model is established first. Then the difference between the baseline response and the response of the damaged specimen will be used as an indicator for damage occurrence, and damage identification algorithms will then be performed to determine the damage location and severity. Normally, all the damage assessment algorithms use a validated baseline model, which is often constructed from finite element method (Zang et al., 2001). Despite the model based methods have many attractive features; in practice the implementation of these methods may encounter difficulties. In a typical method using modal information, the baseline modal parameters are often obtained in terms of mathematical modeling of linear vibrating systems. However, local structural nonlinearities could pose great challenge for such procedure. Furthermore, the accuracy of modeling is not guaranteed. For example, damping factor, while affecting the structural response significantly, is very difficult to model accurately (Zou et al., 2000). In general, for these model-based methods, a complicated model updating procedure via correlated experimental and numerical analyses has to be taken first to establish the baseline (Bagchi, 2005). Additionally, commonly adopted damage assessment algorithms are complex and the damage identification is achieved through a costly and time-consuming inverse process, this ultimately could be a problem for real-time health monitoring applications. There are also uncertainties related to measurement noise, incomplete modal vectors and environmental factors (Humar et al., 2006).

On the other hand the data driven method of structural health monitoring is a direct approach in the assessment of structural condition by analyzing the signals produced from a series of sensors installed on it. In this method of monitoring the condition of the structure could be assessed from signals without creating any mathematical model of the structure. Although this method doesn't require developing any model but it is difficult to predict actual structural condition from pattern of signals. In most cases pure signal analysis can only solve the detection problem at primary level. But for the higher levels and detail assessment, additional information needed. Many research works have been performed and various pattern identification tools have been used to develop an effective data driven method. One of the successful works is "The novel event identification using unsupervised neural computation" (Card et al., 2004). In this research, attempt is made to develop a data driven method that can provide quick information about the damage by successful use of artificial neural network as a pattern recognition tool in combination with wavelet transform as a data de-noising tool. Because both linear and nonlinear structural systems can be treated with Neural Network based approaches (Adeli and Jiang, 2009) and data de-noising by wavelet technique before presenting to neural network has shown better performance (Marwala, T., 2000). In addition, this methodology is used for analyzing the field monitoring data of a Canadian Bridge to assess its applicability.

1.2 Recent cases of bridge failure

It has been observed that the consequences of Bridge failure are huge in terms of lives, people sufferings as well as financial investments. Some examples of such failure, their severity and possible reasons are illustrated shortly here. Investigations revealed that these failures could be avoided if a proper monitoring system were in place. These are few of many examples of recent failure incidents of our infrastructure that has primarily raised the concern and encouraged the governments and owners to think on how we could protect our valuable infrastructure as well as prevent loss of lives by developing an appropriate monitoring system.

1.2.1 Sgt. Aubrey Cosens V.C. Memorial Bridge failure

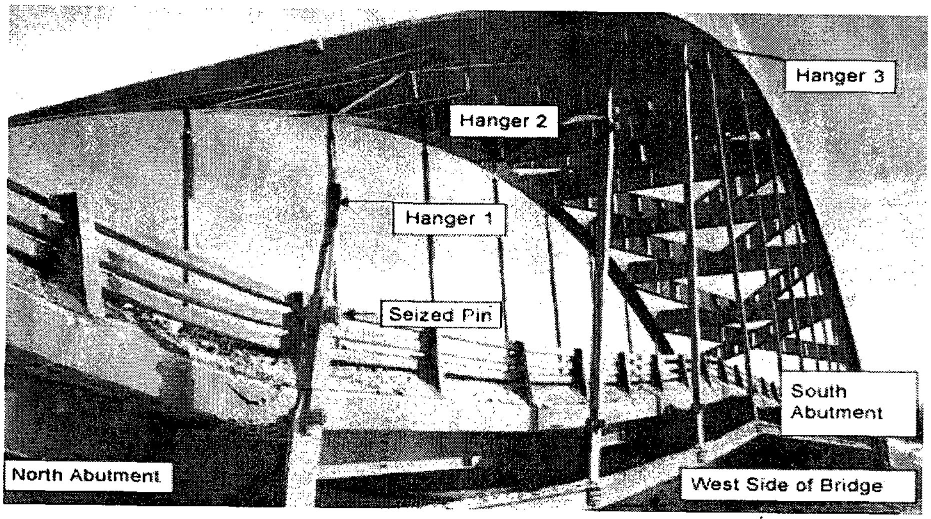

Figure-1.1: Picture of the partial (N-W corner) failure of the Sgt. Aubrey Cosens VC Memorial Bridge (Bagnariol, 2003)

A partial (N-W corner) failure of the Sgt. Aubrey Cosens V.C. Memorial Bridge over the Montreal River at Latchford, ON is shown in Figure-1.1. The failure occurred on January 14, 2003 at 3:00 p.m. (Bagnariol, 2003).

Description of the Bridge: The Sgt. Aubrey Cosens V.C. Memorial Bridge is located on Highway 11, between Temagami and New Liskeard. It is a 110 meter single span steel through arch bridge with vertical hangers. A concrete deck, supported by 12 vertical hangers on each side, is connected to 12 floor beams. The top portion of each hanger has a threaded rod that extends through the bottom plate of the steel arch and is held in place by two nuts. Each hanger also has an eyelet/pin system (top and bottom) to allow free rotation about a transverse axis. A series of steel stringers (6 per bay) tie the floor beams and the concrete deck together. The bridge was built in 1960, rehabilitated in 1992 and the structural steel painted in 1998 (Bagnariol, 2003).

Description of failure: As a Southbound tractor-trailer crossed the bridge; the concrete deck deflected approximately 2 metres at the Northwest corner due to the failure of 3 hanger rods. The bridge was immediately closed to traffic. Structural engineers determined that the deck failed due to fractures in two of the hanger rods which connected the deck to the overhead arch. Initially, only two hanger rods failed, but their failure triggered an adjacent hanger to fail as well. The MTO's engineers concluded that the main contributors to the fatigue induced fracture of the hanger rods include i) hanger pins that had seized, ii) defects introduced in the hanger threads during construction and

iii) steel that did not remain ductile in very cold temperatures. The failure was progressive over a period of several years (Bagnariol, 2003).

From the discussion it is obvious that although the failure was progressive over a period of several years but no earlier maintenance actions have been taken for the protection of the structure due to the lack of a suitable monitoring system.

1.2.2 De La Concorde Overpass failure

The failure of the de la Concorde Overpass occurred on September 30, 2006 (Johnson et al., 2007). A picture of the failure is shown in Figure-1.2. This failure causes serious injuries as well as loss of lives to the user of it.

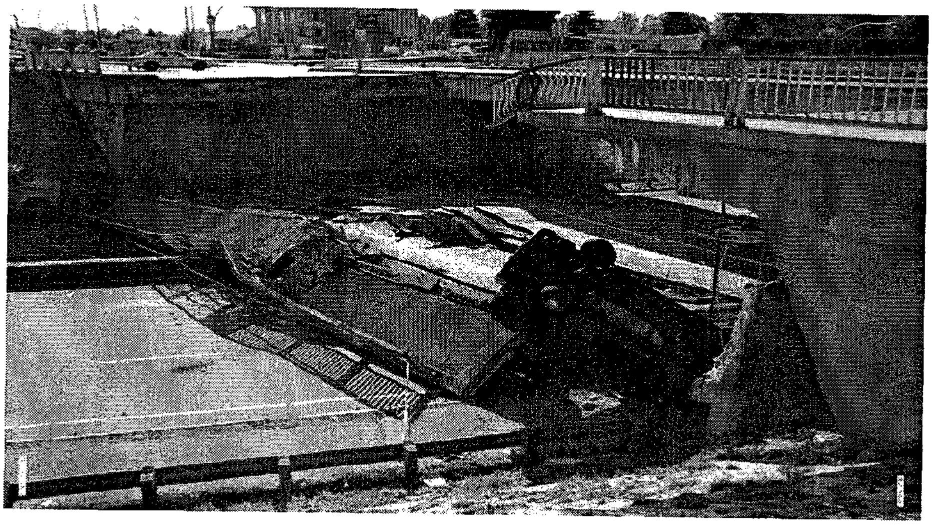

Figure-1.2:.Picture of failure of the de la Concorde Overpass over highway 19 at Laval, Quebec (Johnson et al., 2007).

Description of the Bridge: On September 30, 2006, part of an overpass (65 foot section of a three-lane overpass) collapsed in Laval, Montreal on Boulevard de la Concorde running over Auto route 19. The collapse crushed two vehicles under it, killing five people and seriously injuring six others. The overpass on the Boulevard de la Concorde (Concorde Boulevard) was built in 1970 and was expected to have a life span of 70 years, of which it only lasted 36 (Johnson et al., 2007).

Description of failure: The main cause of failure is the shear failure in the abutment of the Bridge. This was due to a horizontal plane fracture that had slowly grown over the years. Although a regular visual inspection was conducted by MTQ (Ministry of Transportation Quebec) but it couldn't identify any major damage that might cause the failure of the structure even before 30 minutes of the failure (Johnson et al., 2007).

From the above discussion it was clear that only visual inspection was not able to provide any insight information regarding the initiation of the crack of the abutment that ultimately causes this failure. Therefore, as an alternative to the visual inspection, the instrumentation of the Bridge with modern SHM system might be more effective and would protect this Bridge from sudden failure.

1.2.3 I-35W Mississippi River Bridge collapse

The failure of the I-35W Mississippi River Bridge occurred on August 1, 2007. This Bridge was located in Minneapolis, Minnesota, USA. The failure caused severe casualties including the human lives. The picture of the failure of this Bridge is shown in Figure-1.3 (Article on collapse, 2007).

Description of the Bridge: It was an eight lane steel truss arch bridge. The construction of the Bridge started on 1964 and opened in 1967. The central span of the bridge suddenly collapsed into the river followed by adjoining spans. A number of vehicles and people were the victim of this failure. Thirteen people, about 8 males and 5 females lost their lives (Article on collapse, 2007).

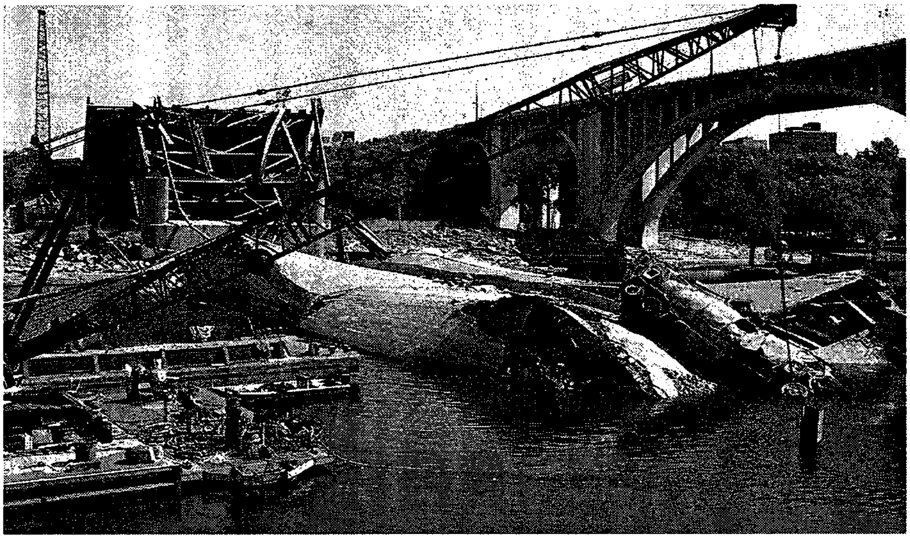

Figure-1.3: Picture of the collapse of I-35W Mississippi River Bridge in Minneapolis, Minnesota, USA (Article on collapse, 2007)

Description of failure: Since 1993 the bridge was brought under periodic visual inspection each year. The federal government declared the bridge "Structurally deficient" in 1990 indicating significant corrosion in its bearings. According to a 2001 study by the civil engineering department of the University of Minnesota, cracking had been previously discovered in the cross girders at the end of the approach spans. Although the report concluded that the bridge should not have any problems with fatigue cracking in the foreseeable future, the bridge instrumentation by strain gauges and continuous structural health monitoring had been suggested (Article on collapse, 2007).

The above incidents clearly points out the deficiency of current maintenance practice for infrastructure and need for more sophisticated structural evaluation using regular monitoring. Following observations are made from the above examples:

- It has been observed that in most of the cases visual inspection has been adopted to assess the condition of the structure at primary level. Although it has noticed some preliminary damage of the structure, but it was not sufficient to notify the damage inside the structure as well as the need of immediate maintenance and rehabilitation.
- In some cases Structural Health Monitoring has been recommended by the expert i.e.Civil Engineering Department of the University of Minnesota, as an alternative and more reliable monitoring system. But no significant measures had been taken for the implementation of it.
- From the above discussions of the failure it has been realized that only visual inspection and periodic monitoring are not sufficient to protect the structures from future damages; it requires a modern and more cost effective continuous monitoring system like SHM that will guarantee the safety as well as protection of valuable infrastructures.
- In addition to these, structural safety and protection from sudden failure is now great concern for engineers and owners. After thorough investigations into it, the engineering experts have agreed that development of an early warning system and accordingly taking precautions could only prevent such a major catastrophe.

1.3 Objectives

The objectives of this thesis are:

a) to develop a practical technique for assessing and tracking structural conditions using the monitoring data; and

b) to develop techniques for assessing the reliability of sensors in a structural health monitoring system.

1.4 Organization of the thesis

The thesis has been divided into seven chapters each of which is then subdivided into sections and subsections. The chapters are arranged in the following sequences:

The background and objective of the thesis is presented in this Chapter-1. A literature review on Bridge Management Issues, Role of SHM, Statistical and Neural Network approach is presented in Chapter-2. A Brief description on Neural Network, training with backpropagation and Wavelet technique is presented in Chapter-3. The methodologies and their explanations with flow chart and block diagram are presented in Chapter-4. A Case study and SHM system of a Canadian Bridge is presented in Chapter-5. The application of the proposed methodology with step by step analysis of the field monitoring data is presented in Chapter-6. Finally, the discussions on the results of analyses, conclusions and future recommendations are presented in Chapter-7.

CHAPTER-2

Literature Review

2.1 Bridge Management Issues

The global developments of communication system are mainly contributed by the construction of road networks and Bridges. As many Bridges are aging; the maintenance and rehabilitation of Bridges are now a great concern for many countries around the world. The term Bridge Management implies to the maintenance, rehabilitation and repair (MR& R) of Bridges by optimizing the use of resources and funds for maintenance and rehabilitation. According to the Commission Report (Johnson et al., 2007), half of Quebec's bridge will need structural works within the next five years. Quebec's transportation ministry aims to have 80 percent of its structures in 'good shape' within 15 years. The commission also calls for Bridge rehabilitation to become a national priority. Further the commission noticed some shortcomings of Ministry of Transportation, Quebec (MTQ) i.e. the inadequate management of structure.

Usually most of the structures are affected with major problems developed within a few years of construction. This ultimately causes the rapid growth of maintenance needs to an unmanageable proportion. This needs to be given special attention. On the other hand statistics show that some of the bridges have become sub-standard due to severe damage which couldn't be repaired by regular maintenance; they need to be taken under rehabilitation program. These situations represent major financial and logistical problems for the authorities concerned (Das, 1999). To deal with this situation we need to develop a technical solution by finding the reasons and taking preventive measures ahead of the catastrophic failure of infrastructures. Some of the effective preventive measures are: a) the use of FRP instead of regular reinforcements to avoid corrosion of steel due to de-icing salt and adverse effect of weather b) adoption of innovative and cost effective monitoring technique i.e. SHM and c) proper utilization of artificial neural networks (ANN) as a prediction tool. Finally for addressing all these issues and taking appropriate preventive measures at the time of inception of any damage, the need for the development of a modern Bridge Management System (BMS) came into the consideration of the researchers. According to the Intermodal Surface Transportation Efficiency Act (ISTEA) and the American Association of State Highway and Transportation Officials (AASHTO) guidelines, a model Bridge monitoring system (BMS) should include four basic modules: a database, a deterioration model, a cost model and an optimization model. These modules are arranged in the following Figure-2.1 according to their dependency. The arrows point to the dependent module (Morcous, 2000).

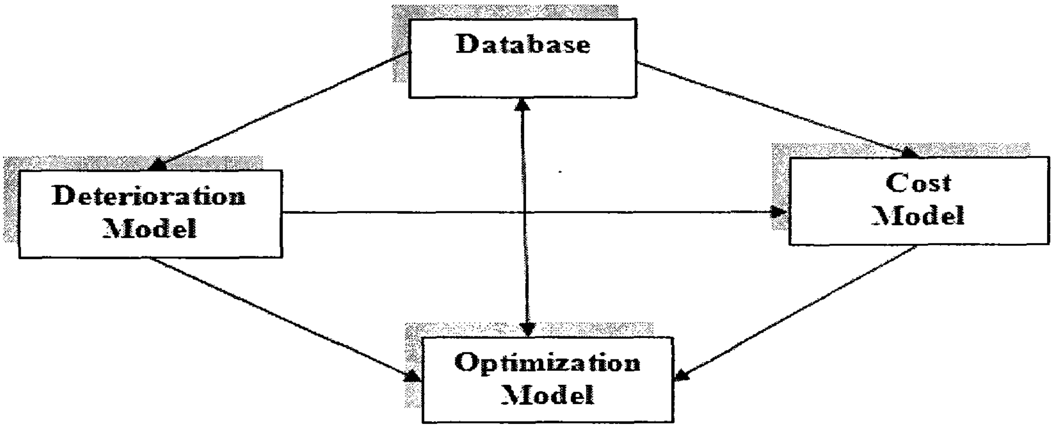

Figure-2.1: Modules of a Model BMS (Czepiel, 1995).

The database module is the heart of any BMS. This module involves mainly the collection and storage of inventory, inspection, appraisal and maintenance data. These data are used by the other BMS modules as a basis for all decisions and actions analyzed by the BMS (Hudson et al., 1987).

The deterioration model is defined as the link between measures of bridge condition and a vector of explanatory variables (Ben-Akiva and Gopinath, 1995). A measure of bridge condition in its simplest form is an assessment of the extent and the severity of a specific damage. More complex measures combine the extent and severity of different damage types in one index. The explanatory variables are defined as the factors affecting bridge deterioration and can be observed or measured. The function of this model is to predict the future condition of different bridge components. This prediction facilitates the determination of the optimal maintenance strategy and the estimation of future funding requirements (Morcous, 2000).

The cost model typically estimates two types of costs: agency cost and user cost (Johnston et al, 1994). Agency cost is estimated to determine the expenditures incurred (labor, materials and equipment) to perform a specific maintenance action. User cost measures the impact of the deterioration on the user as a function of the bridge condition. User costs can also be generated due to bridge deficiencies such as narrow width, low clearance, poor alignment and low load capacity (Morcous, 2000).

The optimization model determines the least-cost MR&R strategies for different bridge components. This model may use life cycle cost analysis to account for the maintenance performed on a bridge throughout its entire service life. This analysis may suggest some maintenance actions to be made before the bridge condition worsens and the cost of maintaining it increases. A BMS may take either a top-down or bottom-up approach to optimization (Czepiel, 1995). The top-down approach determines the desired goals for the entire network and then selects bridges that achieve these goals. The bottom-up approach determines the optimum action for each individual bridge and then selects actions that optimize the network condition. The bottom-up approach provides more meaningful results but it is slower with large bridge populations (Morcous, 2000).

In this thesis we are not going to discuss elaborately Bridge Management System. Rather we will highlight mainly on how SHM could be an important part of the Bridge Management System.

2.2 Role of SHM

SHM as a monitoring system has a potential role in maintaining proper functionality and safety of the structure throughout the whole service life of it. Bridge monitoring and inspection are expensive, yet they are essential tasks in maintaining a safe infrastructure. The primary method used to monitor bridges is visual inspection. During a typical bridge inspection, the various components of a bridge are examined at close range by trained inspectors, who evaluate the condition of the components and give them a condition rating. This rating is a qualitative evaluation of the current condition based on a set of guidelines and on the inspector's experience. For many situations, this type of evaluation is appropriate and effective. However, due to the subjective nature of this evaluation, ratings of the conditions of similar bridge components can vary widely from inspector to inspector, and from state to state. Moreover, there is no assurance of inspections is being performed at the appropriate or critical times (Abudayyeh et al., 2004).

A visual inspection often may not reveal hidden defects, and the quality of the inspection is highly dependent on training, personnel experience, available equipment, and in service environment (Moore et al., 2001). Thus, other nondestructive test methods with structural health monitoring (SHM) concepts have been a major topic of research for maximizing and efficiently using available economic resources, reliable bridge condition evaluations and early detection of possible failures (Alampalli and Ettouney, 2006).

Structural health monitoring, if used properly, has a high potential for bridge management applications such as scheduling maintenance activities just-in-time, improving durability, extending service life, delaying/scheduling rehabilitation and replacement, quality control/assurance of constructed projects, and improving the accuracy of deterioration models for better prediction of future bridge conditions (Alampalli and Ettouney, 2006)

Since its inception, SHM has diverse applications in evaluating the condition of structures. Generally for SHM of bridges there are various kinds of test methods adopted to assess the condition of it. Some of these are discussed below to have preliminary idea of their evolutions:

2.2.1 Static Field Testing

Static field test method determines the load carrying capacity of a structure. It also provides data about a structure's behavior and ability to sustain live loads. These early tests typically measured only deflection, and as a result the data could only be used to determine a bridge's stiffness in flexure and safety in the short term. More recently, various departments of transportation, most notably in Ontario, have instituted programs of static load testing of highway bridges to examine structural behavior and health. These programs have provided a wealth of information on the in-situ behavior of real structures (ISIS, 2004). There are three basic types of static load tests: behavior, diagnostic and proof (ISIS, 2004).

2.2.2 Dynamic Field Testing

Dynamic field tests are used to assess the behavior of structures subject to moving loads. These types of tests are most applicable to bridges, since vehicle loads are generally moving (unless a traffic jam occurs on the bridge). In a typical dynamic field test, a *test vehicle* moves across a "bump" of a pre-determined size on the bridge being tested. The test is usually carried out several times with the test vehicle traveling at a range of velocities. The vehicle hitting the bump introduces an impulsive dynamic load into the structure, which excites the bridge's dynamic response. There are four types of dynamic field tests: stress history tests, dynamic load allowance (DLA) tests, ambient vibration tests, and pull-back tests (ISIS, 2004).

2.2.3 Periodic Monitoring

Structural health monitoring of civil engineering structures can be either periodic or continuous. Periodic SHM is conducted to investigate any detrimental change that might occur in a structure (or in a repair that has been made to the structure). By monitoring the behavior of a structure at specified time intervals (weeks, months, or years apart), changes in the behavior can be detected and these changes may be used as an indication of damage or deterioration (ISIS, 2004)

2.2.4 Continuous Monitoring

Continuous monitoring, as the name implies, refers to monitoring of a structure for an extended period of time (weeks, months, or years). In continuous monitoring, data acquired at the structure are either collected or stored on site (logged) for transfer, analysis, and interpretation at a later time, or they are continuously communicated to an offsite (remote) location. In the most sophisticated of these types of SHM applications, field data are transmitted remotely to the engineer's office for *real-time monitoring* and interpretation. Customarily, continuous monitoring is only applied to those structures that are either extremely important or if there is a doubt about their structural integrity. The latter might be the case if the structure is likely to be exposed to extreme events, such as severe earthquakes and hurricanes, or if its design includes an innovative concept that does not have a history of performance to prove its long-term safety. The SHM system of the Portage Creek Bridge is such an example (ISIS, 2004).

2.3 Statistical & Neural Network approach in pattern recognition

2.3.1 Introduction

Recently, pattern recognition is widely used in solving the problems in a variety of engineering and scientific discipline such as artificial intelligence and remote sensing. Watanabe, 1985 defines a pattern “as opposite of a chaos; it is an entity, vaguely defined, that could be given a name”. For example, a pattern could be a fingerprint image, a handwritten cursive word, a human face, or a speech signal. Given a pattern, its recognition/classification may consist of one of the following two tasks (Watanabe, 1985*)*: 1) supervised classification (e.g., discriminant analysis) in which the input pattern is identified as a member of a predefined class, 2) unsupervised classification (e.g., clustering) in which the pattern is assigned to a previously unknown class (Jain et al., 2000).

Interest in the area of pattern recognition has been renewed recently due to emerging applications which are not only challenging but also computationally more demanding. These applications include data mining (identifying a pattern, e.g., correlation, or an outlier in millions of multidimensional patterns), document classification (efficiently searching text documents), financial forecasting, organization and retrieval of multimedia databases, and biometrics (personal identification based on various physical attributes such as face and fingerprints (Jain et al., 2000).

The design of a pattern recognition system essentially involves the following three aspects: 1) data acquisition and preprocessing, 2) data representation, and 3) decision making. The problem domain dictates the choice of sensor(s), preprocessing technique, representation scheme, and the decision making model. It is generally agreed that a well-defined and sufficiently constrained recognition problem (small interclass variations and large interclass variations) will lead to a compact pattern representation and a simple decision making strategy. Learning from a set of examples (training set) is an important and desired attribute of most pattern recognition systems. The four best known approaches for pattern recognition are: 1) template matching, 2) statistical classification, 3) syntactic or structural matching, and 4) neural networks (Jain et al., 2000). Statistical and Neural Network approaches are discussed in the following subsections:

2.3.2 Statistical Approach

In the statistical approach, each pattern is represented in terms of 'd' features or measurements and is viewed as a point in a d-dimensional space. The goal is to choose those features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in a d-dimensional feature space. The effectiveness of the representation space (feature set) is determined by how well patterns from different classes can be separated. Given a set of training patterns from each class, the objective is to establish decision boundaries in the feature space which separate patterns belonging to different classes. In the statistical decision theoretic approach, the decision boundaries are determined by the probability distributions of the patterns belonging to each class, which must either be specified or learned (Devroye et al., 1996).

One can also take a discriminant analysis-based approach to classification: First a parametric form of the decision boundary (e.g., linear or quadratic) is specified; then the "best" decision boundary of the specified form is found based on the classification of training patterns. Such boundaries can be constructed using, for example, a mean squared error criterion. The direct boundary construction approaches are supported by (Vapnik, 1998) philosophy: If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem.

Statistical pattern recognition has been used successfully to design a number of commercial recognition systems. In statistical pattern recognition, a pattern is represented by a set of d features, or attributes, viewed as a d-dimensional feature vector as we mentioned here. Well-known concepts from statistical decision theory are utilized to establish decision boundaries between pattern classes. The recognition system is operated in two modes: training (learning) and classification (testing) (Figure-2.2). The role of the preprocessing module is to segment the pattern of interest from the background, remove noise, normalize the pattern, and any other operation which will contribute in defining a compact representation of the pattern. In the training mode, the feature extraction/selection module finds the appropriate features for representing the input patterns and the classifier is trained to partition the feature space. The feedback path allows a designer to optimize the preprocessing and feature extraction/selection strategies. In the classification mode, the trained classifier assigns the input pattern to one of the pattern classes under consideration based on the measured features.

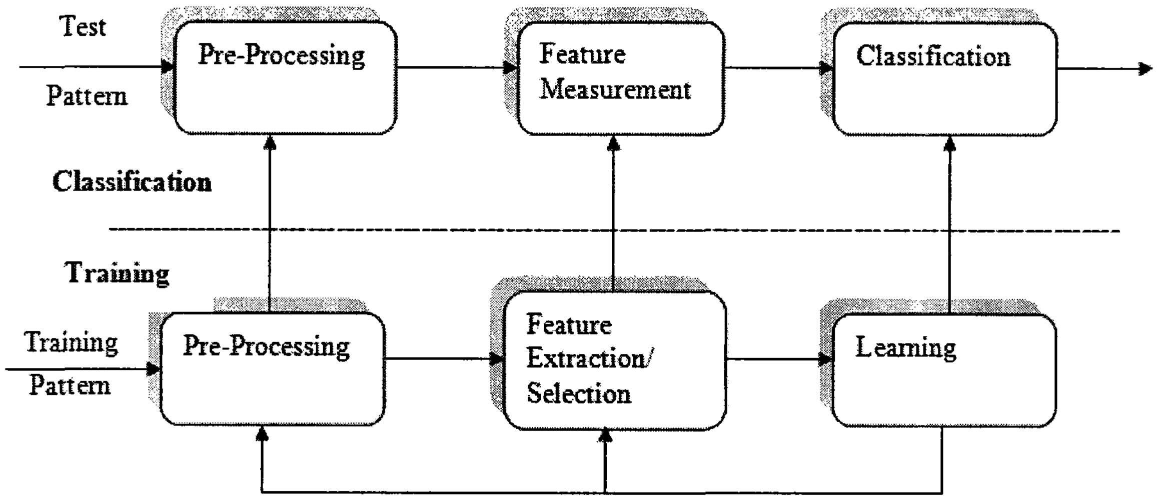

Figure-2.2: Model for statistical pattern recognition (Jain et al., 2000)

2.3.3 Neural Network Approach

Neural networks can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. Neural network models attempt to use some organizational principles (such as learning, generalization, adaptivity, fault tolerance and distributed representation, and computation) in a network of weighted directed graphs in which the nodes are artificial neurons and directed edges (with weights) are connections between neuron outputs and neuron inputs. The main characteristics of neural networks are that they have the ability to learn complex nonlinear input-output relationships, use sequential training procedures, and adapt themselves to the data (application of this procedure has been discussed elaborately in chapter-4 & 5).

The most commonly used family of neural networks for pattern classification tasks (Jain et al., 2000) is the feed-forward network, which includes multilayer perceptron and Radial-Basis Function (RBF) networks. These networks are organized into layers and have unidirectional connections between the layers. Another popular network is the Self-Organizing Map (SOM), or Kohonen-Network (Kohonen, 1995), which is mainly used for data clustering and feature mapping. The learning process involves updating network architecture and connection weights so that a network can efficiently perform a specific classification/clustering task. The increasing popularity of neural network models to solve pattern recognition problems has been primarily due to their seemingly low dependence on domain-specific knowledge (relative to model-based and rule-based approaches) and due to the availability of efficient learning algorithms for practitioners to use.

Neural networks provide a new suite of nonlinear algorithms for feature extraction (using hidden layers) and classification (e.g., multilayer perceptrons). In addition, existing feature extraction and classification algorithms can also be mapped on neural network architectures for efficient (hardware) implementation. In spite of the seemingly different underlying principles, most of the well known neural network models are implicitly equivalent or similar to classical statistical pattern recognition methods. Ripley, 1993 also discusses this relationship between neural networks and statistical pattern recognition. Anderson et al., 1990 pointed out that “neural networks are statistics for amateurs. Most neural networks conceal the statistics from the user”. Despite these similarities, neural networks do offer several advantages such as, unified approaches for feature extraction and classification and flexible procedures for finding good, moderately nonlinear solutions.

2.4 Pattern recognition using ANN

Pattern recognition is one of the most dominant algorithms of the Artificial Neural Network (ANN) and it has created attention of most of the researchers. Numerous works have been done on data analysis of SHM system especially by using pattern recognition algorithm. Many researchers have achieved satisfactory results by decomposing and de-noising data using the wavelet technique before using them as input to the neural networks. Some relevant works those encouraged to develop the main idea of this thesis are shortly described below:

Wenyuan et al., 2007 explained how BP (backpropagation) Neural Network and Wavelet Transform can successfully identify the damage in a structure. They used wavelet transform to decompose acceleration response of a structure in six layers, selecting damage feature of damaged sensitivity based on acceleration, thus recognizing damage time of structure and realizing the surveillance on time of structural damage. Through wavelet package decomposition of the acceleration response they obtained the Eigen vector of energy in all frequency ranges and constructed characteristic parameter input of BP neutral network and then constructed a group of sample network output related to the input. The output can not only indicate location of structural damage and degree of structural damage, but also greatly reduce time spent on training and emulation of network.

Pittner and Kamarthi, 1999 dealt with the assessment of the value of process parameters from the wavelet coefficients of a measured process signal. Since a direct assessment from all wavelet coefficients will often turn out to be tedious or leads to inaccurate results, a preprocessing routine that computes robust features directly correlated to the process parameters is highly desirable. They presented a new efficient feature extraction method based on the fast wavelet transform. This method divides the matrix of computed wavelet coefficients into clusters equal to row vectors. The important frequency ranges have a larger number of clusters than the less important frequency ranges. The features of a processed signal are provided by the Euclidean norms of each such vector. The effectiveness of this new method has been verified on a flank wear estimation problem in turning processes.

The study aimed at describing a preprocessing method for reducing the size of input patterns to neural networks and increasing the estimation or classification performance. The results show that wavelet expansions represent sensor signals from turning experiments efficiently and allow extracting important features about the flank wear state for use in neural network models.

Datta et al., 2007 applied a multi-resolution wavelet analysis coupled with a neural network based approach in the problem of fault diagnostics of industrial robots. The multi-resolution analysis implements discrete wavelet transforms with filters and decomposes the signal in various levels. The approximate and detailed coefficients of the decomposed signals are then used for training a feedforward neural network whose output determines the state (faulty or normal) of the robot. The neural network classifier was then implemented and monitored in a Matlab-Simulink environment using a state-flow model. Validation of the method was performed offline using experimental data obtained from an industrial robot manipulator used in the semi-conductor industry.

Karibasappa and Patnaik, 2004 developed an ANN based algorithm for face recognition, which is insensitive to variation of lighting and facial expression. This method uses the wavelet transform coefficients for feature extraction, which produces the feature components in a low dimensional subspace of an image, even under moderate variation in lighting and facial expression. These coefficients are obtained using fast wavelet transform and preprocessed using principal component analysis. These are compared with reference coefficients for matching purpose. Experimental results show that the performance of the proposed method is better than that of the other existing methods, particularly when facial images have variations in lighting and expressions.

2.5 Summary

From the above descriptions, it has been observed that wavelet analysis has been used as a data processor either by extracting features or by removing noise from the signals. After the preprocessing, the signals have been used as input to the artificial neural network (ANN) which was trained with backpropagation algorithm. For each case the result of the analysis was found satisfactory.

Another important issue in SHM is that there might be some sensor malfunctioning or existence of defective sensor in a series of sensors which needs to be replaced or repaired. It is very difficult to manually identify those defective sensors and no systematic approach currently exists to detect the problem in the sensing system itself and assess its reliability. It is clearly seen that data processing by wavelet transforms prior to applying it to the network gives more accuracy in pattern recognition. Attempt is made in this thesis to combine WT and ANN in the above mentioned way to develop an effective algorithm that can successfully identify defective sensors in addition to track the changes in data pattern.

CHAPTER-3

Neural Network and Wavelet Techniques

3.1 Neural Network

Neural Networks are the computational models inspired by the neuron architecture and operation of the human brain. It is an assembly of layers with highly connected neurons and it works as a processing unit. The strength of the connections between the neurons is represented by weights, **W** (Balageas et al., 2006).

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input /target pairs are needed to train a network (Demuth et al., 2008).

Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems. Neural networks can also be trained to solve problems that are difficult for conventional computers or human beings (Demuth et al., 2008).

An Artificial Neural network (ANN) is a computer program that can recognize patterns in a given collection of data and produce a model for that data.

Artificial Neural Networks resemble the human brain in the following two ways (NeuroDimension, Inc., 2009):

- Knowledge is acquired by the network through a learning process (trial and error).

- Interneuron connection strengths known as synaptic weights are used to store the knowledge.

The brain consists of a large number (approximately 10^{11}) of highly connected elements (approximately 10^4 connections per element) called neurons. For our purposes these neurons have three principal components: the dendrites, the cell body and the axon (Figure-3.1). The dendrites are tree-like receptive networks of nerve fibers that carry electrical signals into the cell body. The cell body effectively sums and thresholds these incoming signals. The axon is a single long fiber that carries the signal from the cell body out to other neurons. The point of contact between an axon of one cell and a dendrite of another cell is called a synapse. It is the arrangement of neurons and the strengths of the individual synapses, determined by a complex chemical process that establishes the function of the neural network. Figure-3.1 is a simplified schematic diagram of two biological neurons (Hagan et al., 1996).

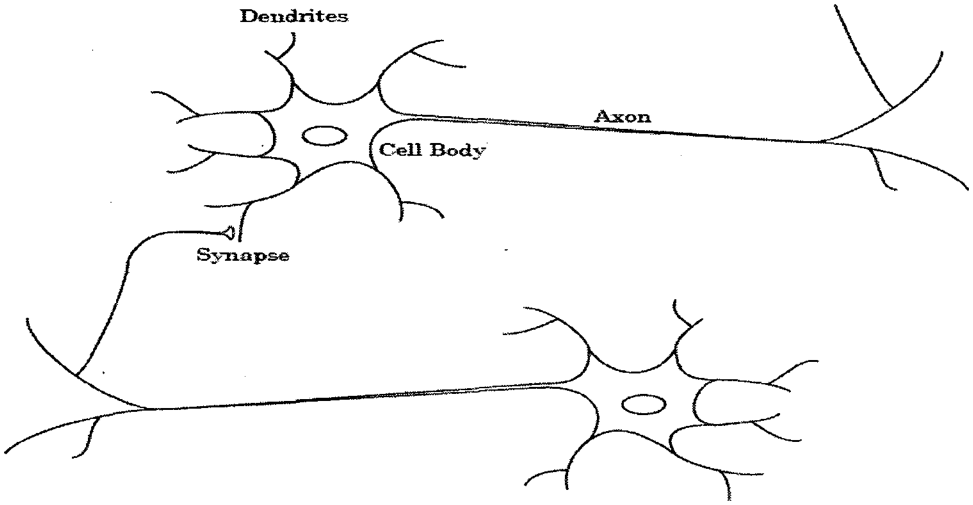

Figure-3.1: Schematic diagram of two biological neurons (Hagan et al., 1996)

Neural structures continue to change throughout life. These later changes tend to consist mainly of strengthening or weakening of synaptic junctions. For instance, it is believed that new memories are formed by modification of these synaptic strengths. Thus, the process of learning a new friend's face consists of altering various synapses.

Artificial neural networks do not approach the complexity of the brain. There are, however, two key similarities between biological and artificial neural networks. First, the building blocks of both networks are simple computational devices (although artificial neurons are much simpler than biological neurons) that are highly interconnected. Second, the connections between neurons determine the function of the network (Hagan et al., 1996).

The true power and advantage of neural networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics (NeuroDimension, Inc., 2009).

3.2 Historical Background of Neural Network

Some of the background works in the field of neural networks done in the late 19th and early 20th centuries. This works mainly focused on interdisciplinary work in physics, psychology and neurophysiology by some of the scientists such as Hermann von

Helmholtz, Ernst Mach and Ivan Pavlov. This early work emphasized general theories of learning, vision, conditioning, etc., and did not include specific mathematical models of neuron operation (Hagan et al., 1996).

In the 1940s the work of McCulloch et al., 1943 introduced the modern view of neural networks. They showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function. Their work is often acknowledged as the origin of the neural network field.

Interest in neural networks become weaker during the late 1960s because of the lack of new ideas and powerful computers were required to experiment. During the 1980s both of these impediments were overcome, and research in neural networks increased dramatically. New personal computers and workstations, which rapidly grew in capability, became widely available. In addition, important new concepts were introduced.

Two new concepts were most responsible for the rebirth of neural networks. The first was the use of statistical mechanics to explain the operation of a certain class of recurrent network, which could be used as an associative memory. This was described in a seminal paper by physicist (Hopfield, 1982).

The second key development of the 1980s was the backpropagation algorithm for training multilayer perceptron networks, which was discovered independently by several different researchers. The most influential publication of the backpropagation algorithm was by (Rumelhart and McClelland, 1986).

These new developments rekindled the field of neural networks. In the last ten years, thousands of papers have been written, and neural networks have found many applications. The field is buzzing with new theoretical and practical work.

3.3 Applications of Neural Network:

Recently neural networks are being applied in many fields. Some of the important applications are shortly described below (Demuth et al., 2008):

Aerospace

High performance aircraft autopilots, flight path simulations, aircraft control systems, aircraft component simulations, aircraft component fault detectors

Automotive

Automobile automatic guidance systems, warranty activity analyzers

Banking

Check and other document readers, credit application evaluators

Defense

Weapon steering, target tracking, radar and image signal processing

Electronics

Integrated circuit chip layout, process control, chip failure analysis, nonlinear modeling

Entertainment

Animation, special effects, market forecasting

Financial

Real estate appraisal, loan advisor, mortgage screening, corporate bond rating, credit line use analysis, corporate financial analysis, currency price prediction

Insurance

Policy application evaluation, product optimization

Manufacturing

Manufacturing process control, product design and analysis, process and machine diagnosis, welding quality analysis, paper quality prediction

Medical

Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant times, hospital quality improvement, and emergency room test advisement

Oil and Gas

Exploration

Robotics

Trajectory control, forklift robot, manipulator controllers, vision systems

Speech

Speech recognition, speech compression, vowel classification, text to speech synthesis

Securities

Market analysis, automatic bond rating, and stock trading advisory systems

Telecommunications

Image and data compression, automated information services, real-time translation of spoken language, customer payment processing systems

Transportation

Truck brake diagnosis systems, vehicle scheduling, routing systems

3.4 Neural Network Structure and Working principle

Generally a neural network consists of one input layer, one or multiple hidden layers and one output layer. Each layer has number of neurons (Demuth et al., 2008).

3.4.1 Neuron Model

The diagram of a neuron model is shown in Figure-3.2. It has one input element **p** , one weight matrix **w** , one bias **b** , net input to the transfer function **n** and output element **a**. We can model a neuron without a bias too (Demuth et al., 2008).

When a network is represented with an input element, the input matrix and the weight matrix multiplied and added to the bias. Finally the output **a** is the function of the total value of this quantity. It can be written in the following form (Demuth et al., 2008).

$$\mathbf{a} = f(\mathbf{w}\mathbf{p} + \mathbf{b}) \text{ --- } (1)$$

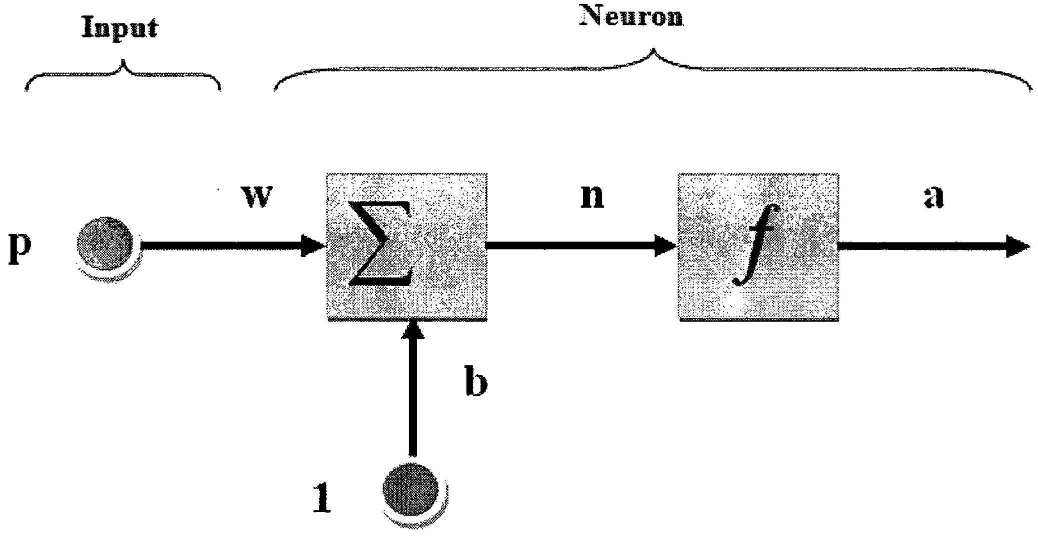

Figure-3.2: A schematic diagram for working principle of a Neuron (Demuth et al., 2008)

3.4.2 Single Layer Network

A single layer network consists of one layer in between input and output layer (Figure-3.3). Here **p** is an R length input vector, **W** is a weight matrix of size **SXR** and **a** and **b** are **S** length vectors. The neuron layer includes the weight matrix, the multiplication operations, the bias vector **b**, the summer, and the transfer function boxes **f**. Here the superscript 1 with all notations indicates layer one. Now the network output can be expressed in accordance with the following formula (Demuth et al., 2008):

$$\mathbf{a}^1 = f^1(\mathbf{IW}_{1,1} \times \mathbf{p} + \mathbf{b}^1) \text{ -- } (2)$$

Here IW= Input weights

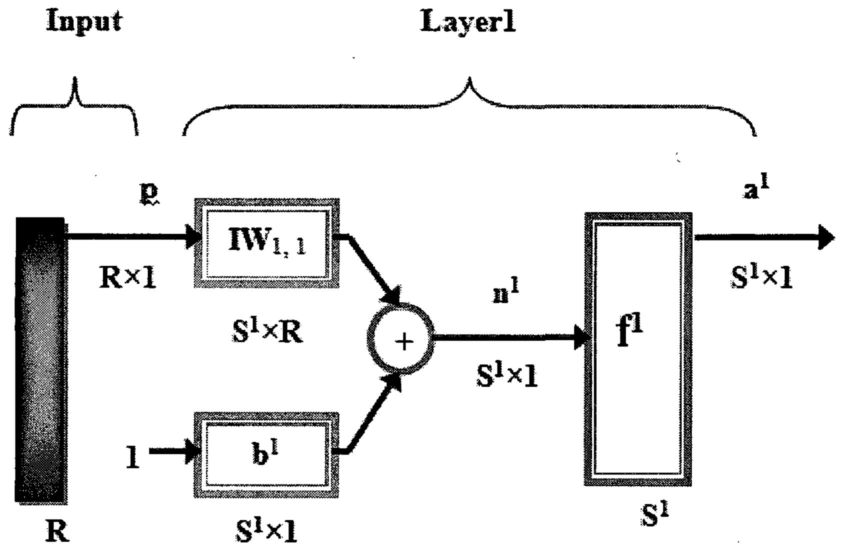

Figure-3.3: A model of the single layer network (Demuth et al., 2008)

3.4.3 Multiple Layers Network

A multiple layer consists of more than one layer as hidden layer in between input and output layer. Each layer has a weight matrix **W**, a bias vector **b**, and an output vector **a**. The network shown in Figure-3.4 has **R** inputs, $\mathbf{S^1}$ neurons in the first layer, $\mathbf{S^2}$ neurons in the second layer, etc. It is common for different layers to have different numbers of neurons. A constant input 1 is fed to the biases for each neuron. Note that the outputs of each intermediate layer are the inputs to the following layer. Thus layer 2 can be analyzed as a one-layer network with $\mathbf{S^1}$ inputs, $\mathbf{S^2}$ neurons, and $\mathbf{S^2XS^1}$ weight matrix **W2**. The input to layer 2 is $\mathbf{a^1}$; the output is $\mathbf{a^2}$. Now that we have identified all the vectors and matrices of layer 2, we can treat it as a single-layer network on its own. This approach can be taken with any layer of the network. The layers of a multilayer network play different roles. A layer that produces the network output is called an output layer. All other layers are called hidden layers. The two-layer network shown in Figure-3.4 has one output layer (layer 2) and one hidden layer (layer 1). If we consider another layer 3 in this network, $\mathbf{a}^2$ will be the input for this network, $\mathbf{S}^3 \times \mathbf{S}^2$ will be the weight matrix $\mathbf{W}_3$ and **a3** will be the output. The multiplication procedure can be expressed step by step according to the following formula (Demuth et al., 2008).

1st layer output, $\mathbf{a}^1 = f^1(\mathbf{IW}_{1,1}\mathbf{p} + \mathbf{b}^1)$ -- (3)

2nd layer output, $\mathbf{a}^2 = f^2(\mathbf{IW}_{2,1}\mathbf{a}^1 + \mathbf{b}^2)$ -- (4)

3rd layer output, $\mathbf{a}^3 = f^3(\mathbf{IW}_{3,1}\mathbf{a}^2 + \mathbf{b}^3)$ --- (5)

Combining equations 3, 4 and 5 we can write the following formula for output of a three layer network:

$\mathbf{a}^3 = f^3\left\{\mathbf{IW}_{3,1} f^2(\mathbf{IW}_{2,1} f^1(\mathbf{IW}_{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3\right\}$ -- (6)

A Multi-layer networks is the most powerful network. For instance, a network of two layers, where the first layer has sigmoid function and the second layer has linear function, can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well (Demuth et al., 2008).

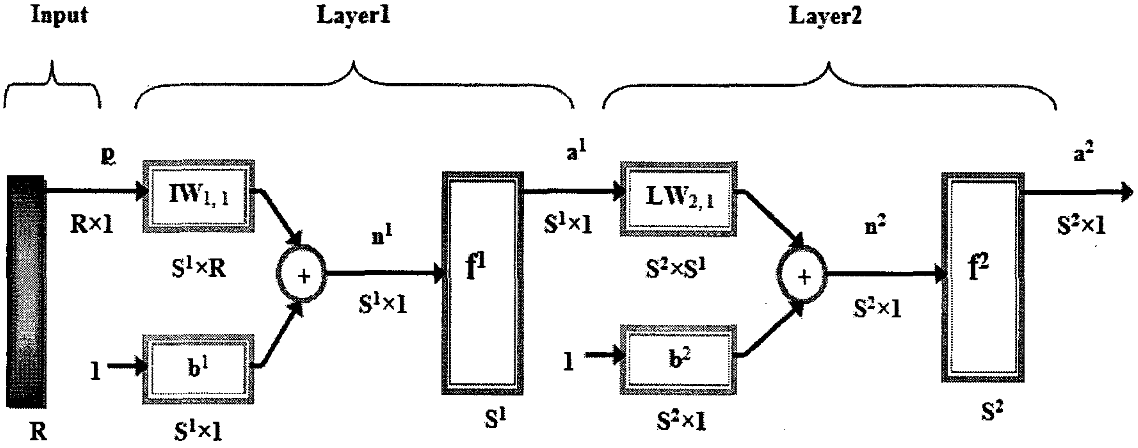

Figure-3.4: A model of the multiple layer network (Demuth et al., 2008)

3.5 Transfer Functions

Generally transfer functions controls the limits of input and output values of the networks. For example if we want to constrain the input or output values of a network then we need to use the sigmoid transfer function. If we don't want to constrain the input or output values then we need to use the purelin or linear transfer function. Some commonly used sigmoid and linear transfer functions have been introduced below (Demuth et al., 2008).

3.5.1 Tan-Sigmoid transfer function

This transfer function is commonly used in multi layer networks. It can constrain the output value between +1 to -1 (Demuth et al., 2008)

Here, $a = \tan sig(n)$

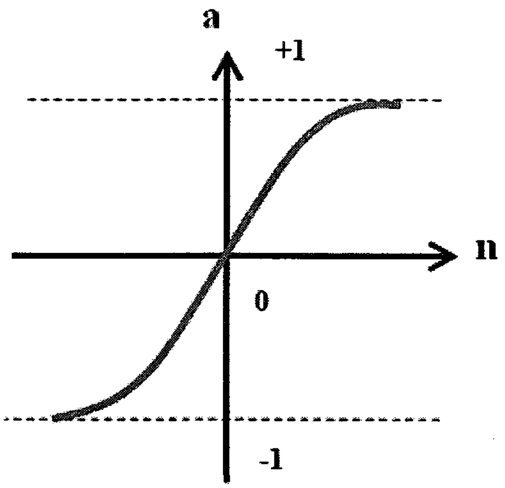

Figure-3.5: A Tan-Sigmoid transfer function

3.5.2 Log-Sigmoid transfer function

This transfer function is also used in multilayer networks. It can constrain the output value between 0 and +1 (Demuth et al., 2008).

Here, $a = \log sig(n)$

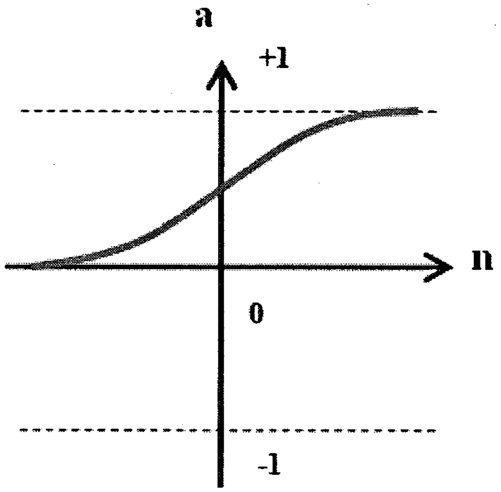

Figure-3.6: A Log-Sigmoid transfer function

3.5.3 Linear transfer function

This transfer function can take any value without any limit. It can produce output values beyond the range +1 to -1 (Demuth et al., 2008).

Here, $a = purelin(n)$

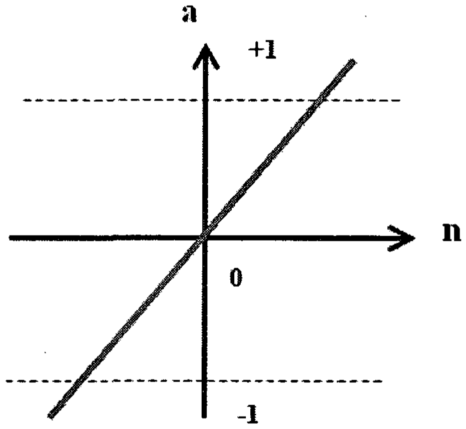

Figure-3.7: A Linear transfer function

3.6 Feed Forward Network

The Feed-Forward Networks have one or more hidden layers of sigmoid transfer function and one output layer with linear transfer function. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network produce values beyond the range −1 to +1.

On the other hand, if we want to constrain the outputs of a network, then the output layer should use a sigmoid transfer function (such as *logsig* for 0 to 1 and *tansig* for -1 to +1). For multiple-layer networks the number of layers determines the superscript on the weight matrices. Figure-3.8 shows an architecture of the two layer Feed forward Network with appropriate notations, *tansig* function in the hidden layer and *purelin* function in the output layer (Demuth et al., 2008).

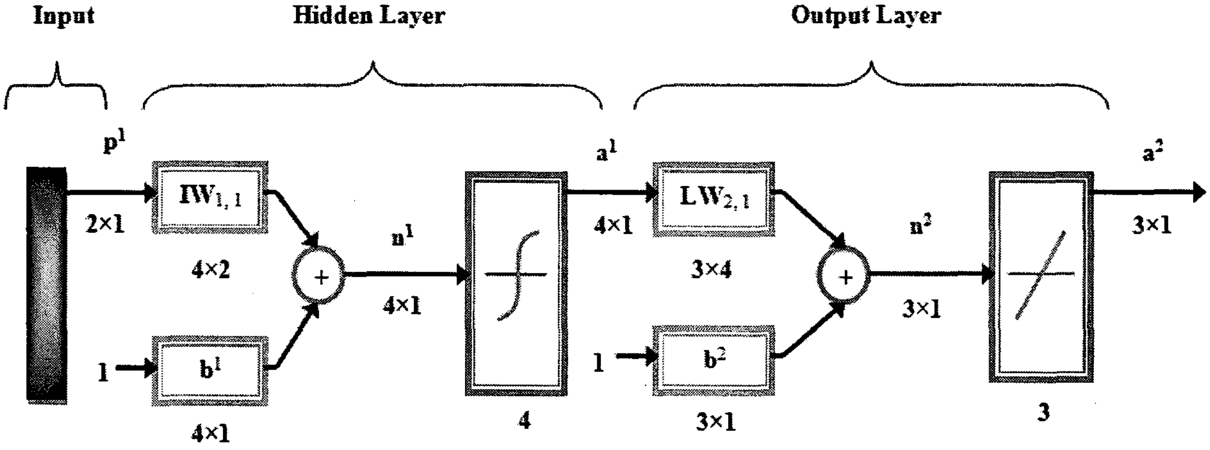

Figure-3.8: A model of a Feed Forward network (Demuth et al., 2008)

This network can be used as a general function approximator. It can approximate any function with a finite number of discontinuities arbitrarily well, given sufficient neurons in the hidden layer.

As MATLAB (R2008b) is used in the current study, some of the basic functions are discussed in the following subsections:

3.6.1 Creating a Feedforward Network in MATLAB

In MATLAB, the function *newff* creates a feedforward network. It requires three arguments and returns the network object. The first argument is a matrix of sample R-element input vectors. The second argument is a matrix of sample S-element target vectors. The sample inputs and outputs are used to set up network input and output dimensions and parameters. The third argument is an array containing the sizes of each hidden layer. (The output layer size is determined from the targets.)

If we want to create a feed-forward network with three neurons in one hidden layer, the following command in MATLAB is enough to create it (Demuth et al., 2008).

net = newff (**p** ,**t**, *3)*

Here, *p* is input matrix, *t* is target of the network.

This command creates the network object and also initializes the weights and biases of the network; therefore the network is ready for training. There are times when we might want to reinitialize the weights, or to perform a custom initialization. The following sub section explains the details of the initialization process.

3.6.2 Initializing Weights (init)

Before retraining a feedforward network, we must reinitialize the weights and biases. The *newff* command automatically initializes the weights, but for retraining of the network we have reinitialized them with this function *init* in MATLAB. This function takes a network object as input and returns a network object with all weights and biases initialized. Here is the command that we have used to reinitialize the network for retraining (Demuth et al., 2008):

net = init (net)

This command in MATLAB reinitializes the network weights and biases

3.6.3 Simulation (sim)

The function *sim* simulates a trained network whenever represented with new inputs. It takes the network input and the trained network and returns the network outputs (Demuth et al., 2008).

a = sim (net, p)

Where, *net* is the trained network, *p* is the network input and *a* is the network output.

3.7 Training of a Feedforward Network with Backpropagation

The training of the network by using the backpropagation algorithm has been discussed in the following subsections:

3.7.1 Backpropagation Algorithm

In the simplest implementation of backpropagation learning, it updates the network weights and biases in the direction in which the performance function decreases most rapidly. One iteration of this algorithm can be written as follows (Demuth et al., 2008):

$$x_{k+1} = \mathbf{x}_\mathrm{k} - \alpha_k g_k \text{ ---} (7)$$

Where, x_k is a vector of current weights and biases, g_k is the current gradient, and α_k is the learning rate.

3.7.2 Training with backpropagation

Backpropagation is the generalization of the Widrow and Hoff (1960) learning rule to multiple-layer networks and nonlinear differentiable transfer functions (tansig or logsig). In this method of training the input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined (Demuth et al., 2008).

Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

Once the network weights and biases are initialized, the network is ready for training. The network can be trained for function approximation (nonlinear regression), pattern association, or pattern classification. The training process requires a set of examples of proper network behavior—network inputs **p** and target outputs **t**. During training the weights and biases of the netwok are iteratively adjusted to minimize the network performance function. The default performance function for feedforward networks is mean square error (MSE)—the average squared error between the network outputs *a* and the target outputs *t*. It should be noted that properly trained backpropagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, if a trained network is represented with the new inputs those are similar to the training inputs then the outputs for these new inputs become very closer to the output during training.

3.7.3 Assessing the degree of fitness

A number of metrics, such as coefficient of determination R^2 (Cox, 1987), mean squared error MSE (Timothy, 2005) and correlation coefficient CC (Cox, 1987) can be used for comparing the actual and simulated output data to assess the performance of the network. The coefficient of determination R^2 close to 1 means the fitting of the curves is good. Negative R^2 means that remodeling or retraining is required. On the other hand MSE gives us the idea about the average error between the actual and output. The Correlation coefficient is the measure of correlations between two variables. If the Correlation Coefficient is close to either +1 or -1, it means the correlation is strong.

The statistical formulas for these coefficients are as follows:

Coefficient of determination, R-squared (Cox, 1987):

$$R^2 = 1 - \frac{SS_{error}}{SS_{total}} \text{------------------------------------} (8)$$

Where, $ss_{error} = \sum (actual - output)^2$ and $ss_{total} = \sum (actual - meanactual)^2$

Mean-Squared-Error (Timothy, 2005):

$$MSE = \frac{1}{N}\sum (actual - output)^2 \text{------------------------------------} (9)$$

Correlation Coefficient (Cox, 1987):

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{(n-1)S_x S_y} \text{ -- (10)}$$

In the above equation, x and y are two variables, i =1 to n S_x, S_y = standard deviation of x and y.

3.8 Wavelet

A wavelet is a waveform of effectively limited duration that has an average value of zero. If we compare wavelets with sine waves, which are the basis of Fourier analysis, sinusoids do not have limited duration — they extend from minus to plus infinity. And where sinusoids are smooth and predictable, wavelets tend to be irregular and asymmetric (Figure-3.9 a and b) (Misiti et al., 2008).

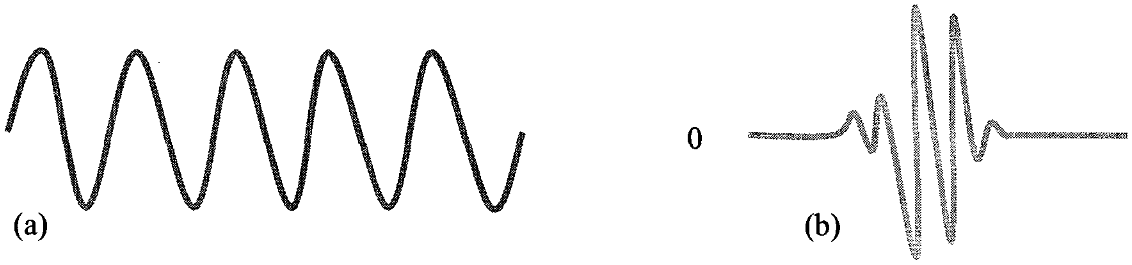

Figure-3.9: Example time histories: (a) Sine wave, (b) Wavelet (db10)

Fourier analysis consists of breaking up a signal into sine and cosine waves of various frequencies. Similarly, wavelet analysis breaks up a signal into shifted and scaled versions of the original (or mother) wavelet.

Fourier analysis has a serious drawback. In transforming to the frequency domain, time information is lost. When looking at a Fourier transform of a signal, it is impossible to tell when a particular event took place.

Wavelet technique can be expressed as a windowing technique with variable-size regions. Wavelet analysis allows the use of long time intervals where one wants more precise low frequency information, and shorter regions where one wants high frequency information. One major advantage afforded by wavelets is the ability to perform local analysis, that is, to analyze a localized area of a larger signal. Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss aspects like trends, breakdown points, and discontinuities. Furthermore, because it affords a different view of data than those presented by traditional techniques, wavelet analysis can often compress or de-noise a signal without appreciable degradation. Indeed, in their brief history within the signal-processing field, wavelets have already proven themselves to be an indispensable addition to the analyst's collection of tools (Ovanesova and Suarez, 1999).

3.8.1 Applications

Wavelet has various applications in the field of data analysis. Some of the important fields that are making use of wavelets include astronomy, acoustics, nuclear engineering, sub-band coding, signal and image processing, neurophysiology, music, magnetic resonance imaging, speech discrimination, optics, fractals, turbulence, earthquake-prediction, radar, human vision, and pure mathematics applications such as solving partial differential equations (Graps, 1995).

3.9 Different types of wavelets

There are a number of wavelets available including Haar, Daubechies, Biorthogonal, Coiflets, Symlets, Morelet, Mexican Hat and Meyer etc. (Misiti et al., 2008). Data can be denoised using any one of them. Here, the daubechies wavelet has been used for data denoising for their better accuracy as compared to the other wavelets, while introducing minimum changes in the original data patterns in a time series. A brief introduction on daubechies and its effectiveness on denoising in comparison to haar wavelet has been given below (Figure-3.11and Figure-3.12):

3.9.1 Daubechies:

This wavelet was invented by Ingrid Daubechies (Daubechies, 1992). They are called compactly supported orthonormal wavelet. The daubechies families of wavelets are written as dbN where N is the order and db is the family name of the wavelet. The daubechies wavelet function that we have used to decompose data is shown in the following Figure-3.10 (Misiti et al., 2008).

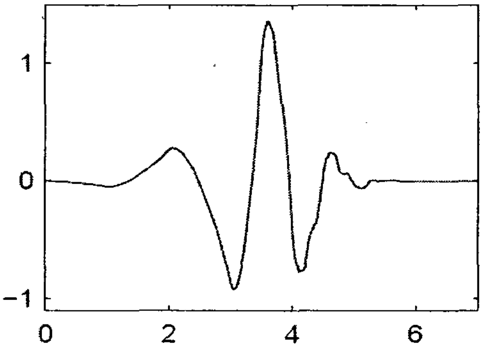

Figure-3.10: Daubechies (db4) wavelet function (Misiti et al., 2008)

3.10 Data de-noising by Wavelet

Data de-noising means the removal of high frequency data and it helps to detect the actual data pattern (Misiti et al., 2008). The data collected from various sensors of a SHM system are likely to contain extraneous information and noise which actually do not have any important information. These noisy data needs to be removed or preprocessed to achieve more accuracy during analysis. This preprocessing should be able to remove most of the noise that is unrelated to the signal structure without impairing our ability to detect structural changes (Misiti et al., 2008). There are other data denoising procedures e.g. Filtering, Fourier analysis etc.

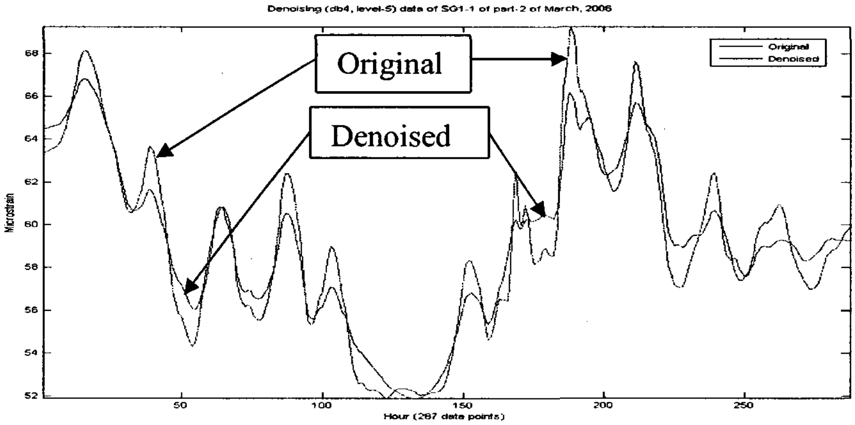

Figure-3.11: An example of denoising hourly data of SG1_1 of 2nd part of March, 2006 using db4 at level-5

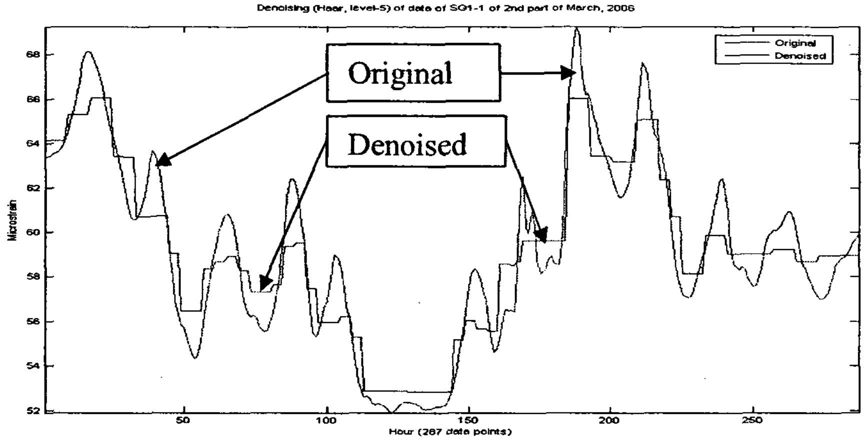

Figure-3.12: An example of denoising hourly data of SG1_1 of 2^{nd} part of March, 2006 using haar at level-5

3.10.1 One-Dimensional Discrete Wavelet Analysis

This is a MATLAB based graphical user interface tool. This can be used to de-noise one signal at a time. It has been used to de-noise the single sensor data mainly the target data of the network. Although there are number of options available for fine tuning the de-noising algorithm, we have used the defaults of *soft, fixed form threshold* and *unscaled white noise*. It should be mentioned that for data de-noising we have used *daubechies* wavelet (db4) at level-5. Then we saved de-noised signal and represent it to the trained network (Misiti et al., 2008).

3.10.2 One-Dimensional Multi-signal Wavelet Analysis

This is also MATLAB based graphical user interface tool. The main advantage of this tool is that we can de-noise more than one signal at a time. We have de-noised sixteen sensor data at a time by using this tool before representing them into the trained network as input. Here we also used daubechies wavelet (db4) at level-5 and the defaults of *soft,* *fixed form threshold and unscaled white noise.* Then we saved the de-noised data in a separate file and represent it to the trained network (Misiti et al., 2008).

3.11 Summary

In this chapter, the Neural Networks and Wavelet Transform including their various fields of applications are introduced. For neural network, the network training using the backpropagation is discussed. Training, initialization and simulation process of the network using MATLAB are also explained. In addition to this different transfer functions of the neural networks and their working principles are explained.

For wavelet analysis, the functions, working principles and its advantage over Fourier series are discussed. Different wavelet families including the daubechies family of wavelet that has been applied in decomposing the data are described. In addition to this, the method of data de-noising using both one dimensional discrete wavelet analysis and one dimensional multi-signal analysis are discussed.

The whole chapter discussed the background review of the basic principles that has been applied in the present study.

CHAPTER-4
Methodologies

4.1 Introduction

The main goal of this thesis is to develop a process by using Neural Networks and Wavelet technology that can be applied in tracking the changes of data pattern of a SHM system as well as in detecting the defective sensor from a series of sensors installed on a structure. Normally it is very difficult to identify the defective sensor from a series of sensors with visual inspection of the data only. As a result of it, an attempt is made to develop a technique that has been found very effective in addressing this important task.

In establishing such methodology first, some small problems related to our objectives have been considered, solved them with different methodologies, verified the results and finally selected the most effective one. Then this methodology has been applied to analyze the raw data collected from the SHM data base of Portage Creek Bridge in Victoria BC. This data base has been developed to store the monitoring data from all the sensors installed on the bridge for future analysis. In this study we have considered the strain gauge data from one of the instrumented column of the Bridge. These sensors produce 16 outputs both in horizontal and vertical directions. In addition, one temperature data has been considered from the same instrumented column.

Data of different periods have been collected to train up Neural Networks and finally we used this trained neural network to identify the changes in data pattern as well as the defective sensors from which the actual condition of the structure has been determined. In this analysis, firstly, we selected data of different time periods and represented these data to the trained networks to find out an appropriate algorithm; secondly, we applied these algorithms for analyzing the field monitoring data of Portage Creek Bridge to verify its applicability from step by step investigations as summarized below. It should be noted that the initial detection of defective sensor might be an indication of either sensor malfunctioning or any damage in the structure which needs to be investigated further. A description of ANN modeling, flow charts of the whole procedure are presented in the following sections.

4.2 ANN modeling

The field monitoring data of the Portage Creek Bridge has been collected from the SHM data base of ISIS Canada (ISIS , 2009). The short columns C1 & C2 of Pier-2 (Figure-5.1) have been retrofitted with GFRP wrap and instrumented with eight bi-directional (one vertical and one horizontal strain) strain gauges. There is also one temperature gauge with each column. This data base has storage of monitoring data from the year 2003 to 2006 at different time intervals i.e. 1/32 sec, 1 sec, 10 sec and 1 minute. We have picked up the 16 strain gauge and one temperature gauge readings of column-2 of 2006 at different time intervals.

In modeling the network, the input neurons have been selected with respect to the number of sensors data considered as input and the output neuron with respect to the sensor data considered as target. In this way we have modeled different networks with different input-output combinations as required. In Figure-4.1 such a network model has been shown which consists of sixteen neurons in the input layer receiving 16 strain gauge data as input and one neuron in the output layer receiving 1 strain gauge data as target. The number of hidden neurons of the network varies in accordance with the network training performance. Thus the final network model is selected with fine tuning the network model by changing the number of hidden neurons and checking the training performance until it shows the best performance. While training a network in MATLAB, its performance can also be checked at any time.

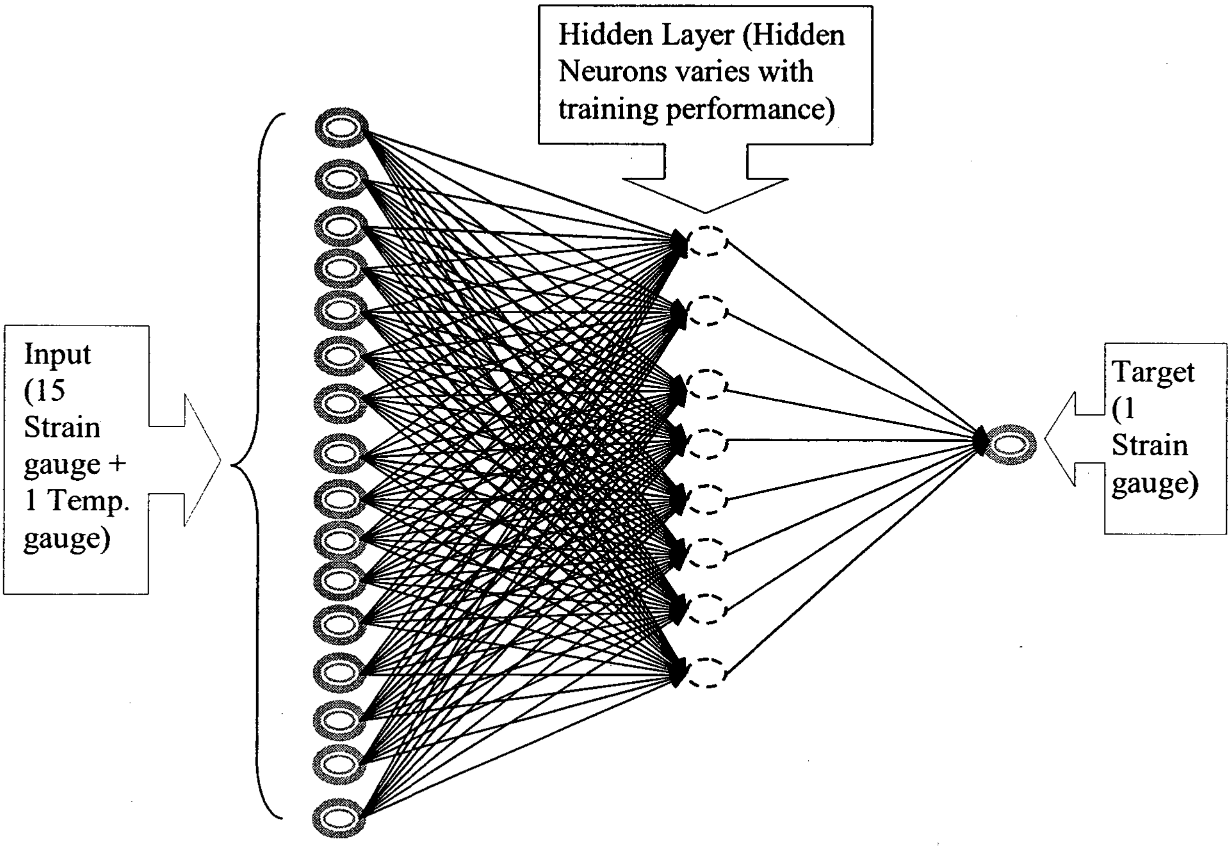
Hidden Layer (Hidden Neurons varies with training performance)
Input (15 Strain gauge + 1 Temp. gauge)
Target (1 Strain gauge)

Figure-4.1: Sample of a network model

4.3 Method of tracking the data pattern

The method of tracking the changes in data pattern in a SHM system has been explained by a Flow Chart as shown in Figure-4.2. The steps in the flow chart are shortly described below:

1. Data harvesting: In this step data are harvested from SHM data base
2. Training Input and Target: In this step Input and Target data are separated from harvested data for training of the networks (ANN)
3. New Inputs: In this step test data are separated from harvested data for testing the network
4. Data de-noising using wavelet transform: In this step all (training and testing) data are de-noised by using one-dimensional discrete wavelet analysis and one-dimensional multi-signal wavelet analysis
5. Data Normalization using MATLAB: In this step all data (training and testing) are normalized using MATLAB
6. Processed Training Inputs and target: In this step the processed (de-noised and normalized) training inputs and target data are separated to train a Feed forward network
7. Feed forward back propagation ANN model: At this step a feed forward back propagation network model is prepared
8. Training of the network and check training performance: In this step the feed forward network is trained and its performance is checked from MATLAB.
9. Initialize and retrain network: If the training performance of the trained network is not ok then initialize and retraining done iteratively until the performance is ok

10. Trained network: In this step the trained network is selected when the performance is good and it is ready to accept the processed test data as new inputs.
11. Processed new inputs: In this step the processed test data are separated for presenting it to the trained network
12. Network Output: In this step the output of the network for the test inputs (new inputs) are collected
13. Plot Actual vs. Output: In this step we plot network output (for test input) with respect to the actual target
14. R^2, MSE and CC: In this step we calculate these metrics to assess the degree of fitting

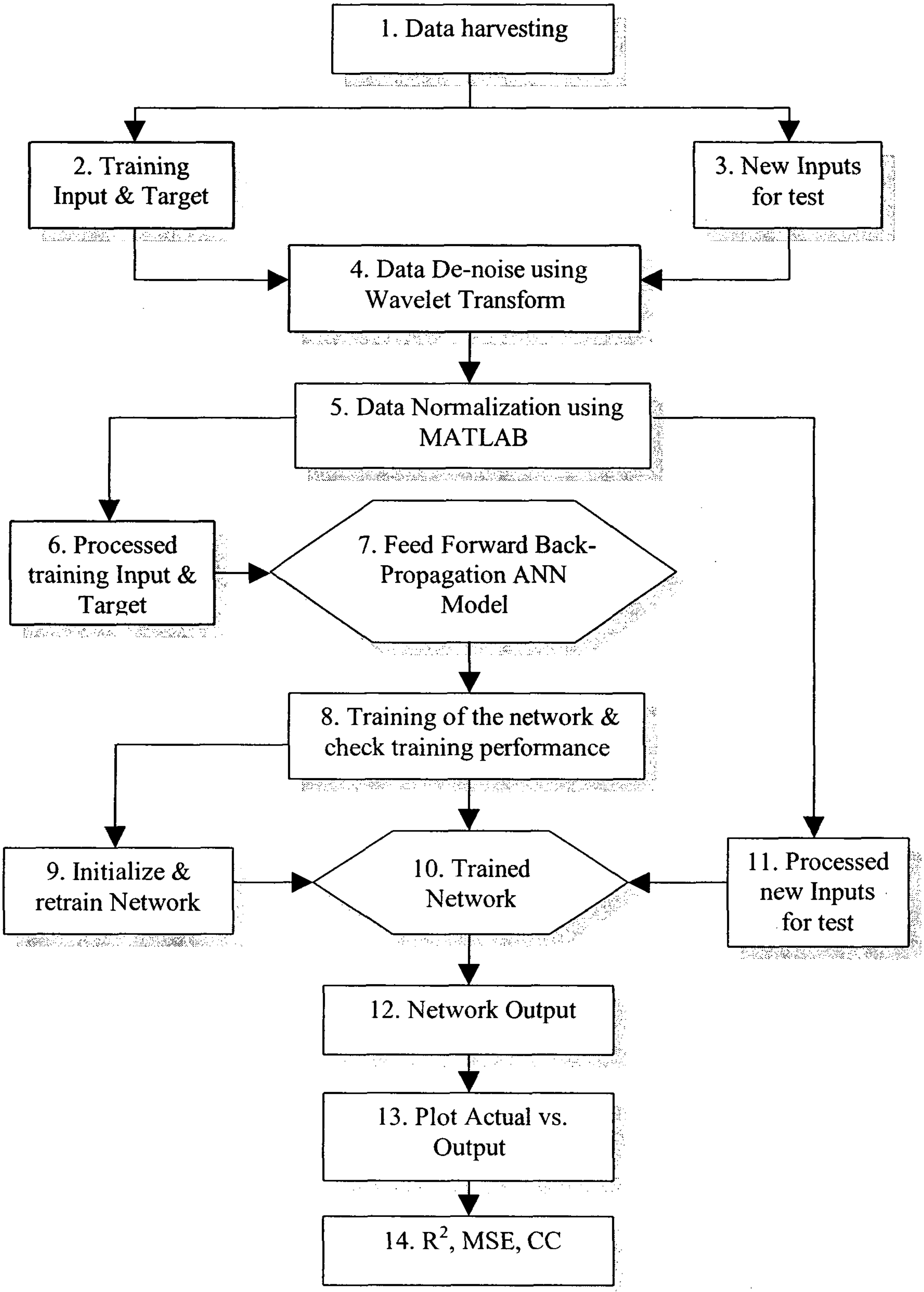
1. Data harvesting
2. Training Input & Target
3. New Inputs for test
4. Data De-noise using Wavelet Transform
5. Data Normalization using MATLAB
6. Processed training Input & Target
7. Feed Forward Back-Propagation ANN Model
8. Training of the network & check training performance
9. Initialize & retrain Network
10. Trained Network
11. Processed new Inputs for test
12. Network Output
13. Plot Actual vs. Output
14. R^2, MSE, CC

Figure-4.2: Flow Chart for tracking the changes in data pattern by Wavelet based ANN method

4.4 Method of identifying defective sensors

Two methods have applied for detecting the defective sensors from a series of sensors in a SHM system. One is Sequential Search method (Figure-4.3) and the other is Binary Search method (Figure-4.4). The steps of these methods are explained by block diagrams as follows:

4.4.1 Sequential Search method

The steps in this method are briefly described below (Figure-4.3):

1. Selected Sensors: In this step sensors have been selected for sequential search
2. Training of networks by sequentially taking out one sensor each time: In this step one sensor gauge has been taken as target from the selected sensors and the networks have been trained considering inputs from remaining sensors each time eliminating one sensor from the inputs.
3. Testing the trained networks: In this step each trained network has been tested with new inputs from similar sensor each time

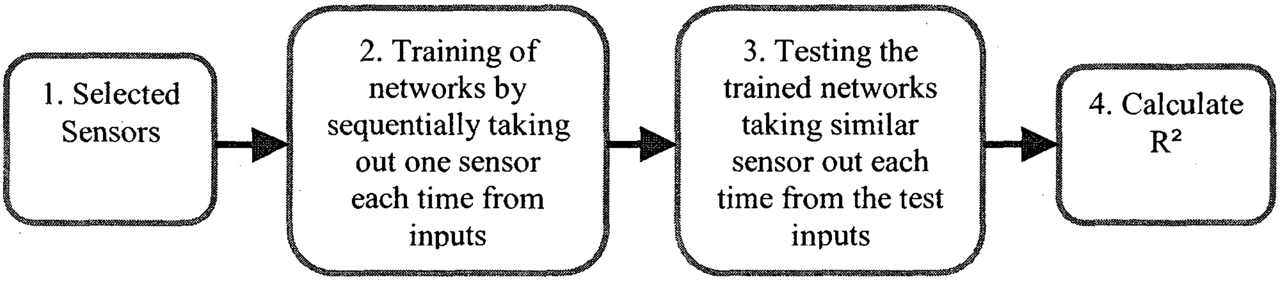
1. Selected Sensors
2. Training of networks by sequentially taking out one sensor each time from inputs
3. Testing the trained networks taking similar sensor out each time from the test inputs
4. Calculate R²

Figure-4.3: Block diagram for Sequential Search method

4. Calculate R-squared: In this step R-squared have been calculated from plots of network output vs. target. Then the R-squared have been plotted and the sensor at the maximum value of R-squared is the indicator of the defective sensor.

4.4.2 Binary Search method

The following steps have been followed in binary search method of the defective sensors (Figure-4.4):

1. Sensor Data: In this step data from the selected strain gauges are collected for processing
2. Process by Wavelet: In this step the collected data are processed by de-noising using Wavelet analysis
3. Binary Search: In this step the total number of strain gauge is divided into two group
4. Sensor Group-1: In this group 50% of the total sensor data are selected
5. Sensor group-2: In this group other 50% of the total sensor data are selected
6. Trained Network-1: In this step 1st network was trained considering the similar sensors as in Group-1 that was collected immediately before the time of collection of test inputs of Group-1
7. Trained Network-2: In this step 2nd network was trained considering the similar sensors data as in Group-2 that was collected immediately before the time of collection of test inputs of Group-2
8. Output-1: In this step the outputs of the network-1 due to the test inputs of sensor group-1 are plotted

9. Output-2: In this step the outputs of the network-2 due to test inputs of sensor group-2 are plotted
10. Compare pattern: In this step the pattern of these two outputs are compared
11. Defective Sensor Group: In this step the apparently defective sensor group is selected from comparison of data pattern and R^2 values
12. Sequential search: In this step sensors are eliminated one by one from the apparently defective sensor group and tested in the respective trained networks
13. Trained Networks: In this step the networks are trained by eliminating similar Strain gauges as test data of apparently defective sensor
14. Outputs: In this step the network outputs are plotted against the actual target values
15. Sensor at point of max. R^2: In this step the sensor at the point of maximum R^2 value has been identified as the defective sensor

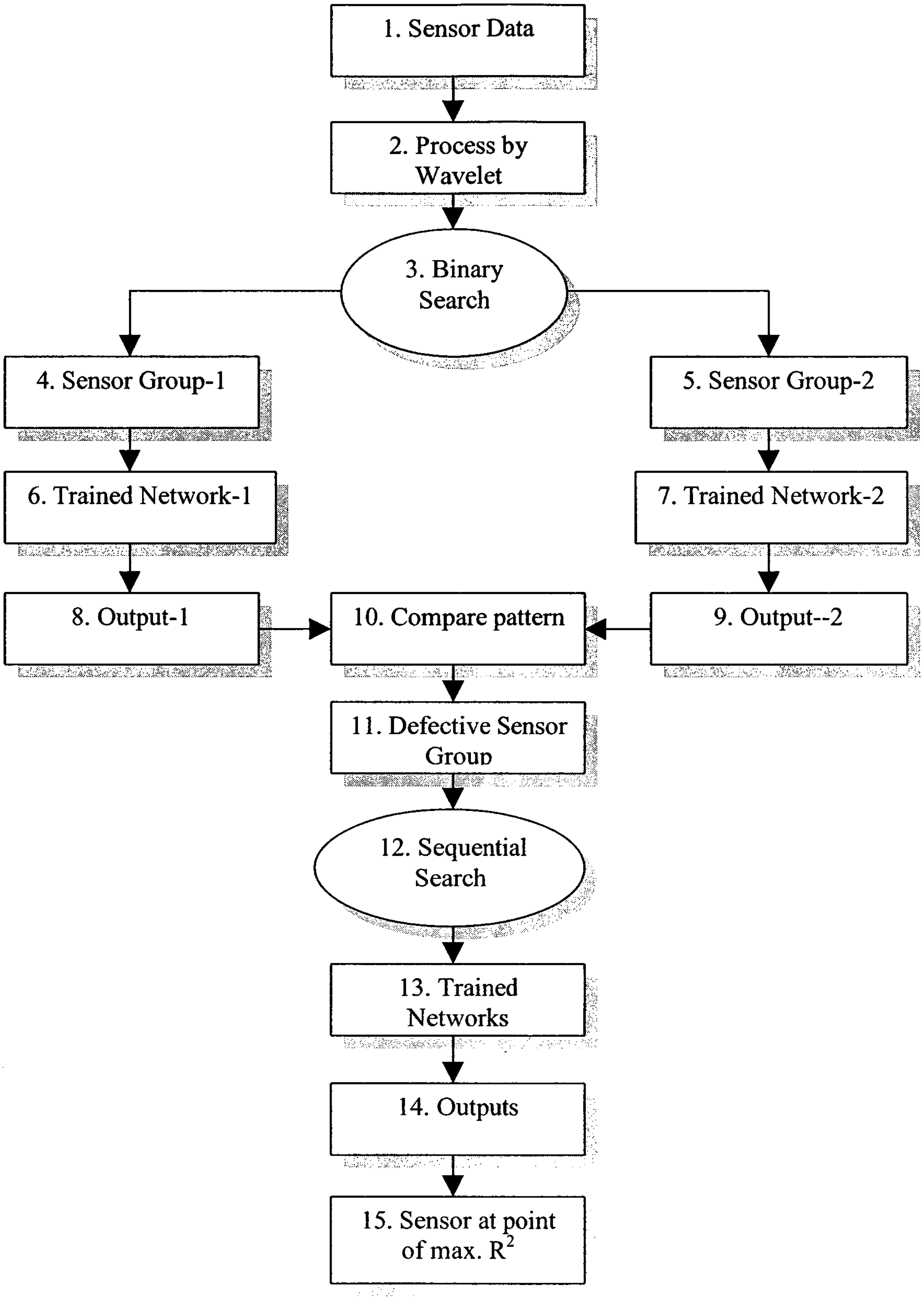

Figure-4.4: Block diagram for detection of defective sensor by the binary search method

4.5 Method of condition assessment from sample data blocks

In this section an alternative approach for identification of damage state of the structure has been discussed. In this method several sample data blocks for selected strain gauge readings are considered. The first block of each series of a particular strain is considered the reference block and rest of the blocks are called test data blocks for comparison. From observation of signal patterns from the blocks of signals at a particular time interval for different time span like each month or week or day might give us a clear picture of the strain variation with time. For example, if we identify the steady and non-steady part of a signal block, then the maximum picks of the non steady portion would be due to the impact from live load or moving loads of the vehicles. Generally, these maximum pick values would represent the maximum strain of the structure at this point of time. Finally these maximum strain values have been compared to the theoretical strain values calculated due to the static load using of a truck on the corresponding column of the Portage Creek Bridge (Noman, 2008)

CHAPTER-5

SHM system of the Portage Creek Bridge

5.1 Portage Creek Bridge

The Portage Creek Bridge is located in the city of Victoria, British Columbia and the bridge crosses Interurban road and Colquitz River at McKenzie Avenue. It is a 124m long, three-span structure with a reinforced concrete deck supported on two reinforced concrete piers, and abutments on H piles. The deck has a roadway width of 16.2 m (53 ft) with two 1.98 m (6′-6″) sidewalks and aluminum railings. The bridge was designed in 1982 prior to the introduction of current bridge seismic design codes and construction practices. Therefore, it was not designed to resist the earthquake forces as required by

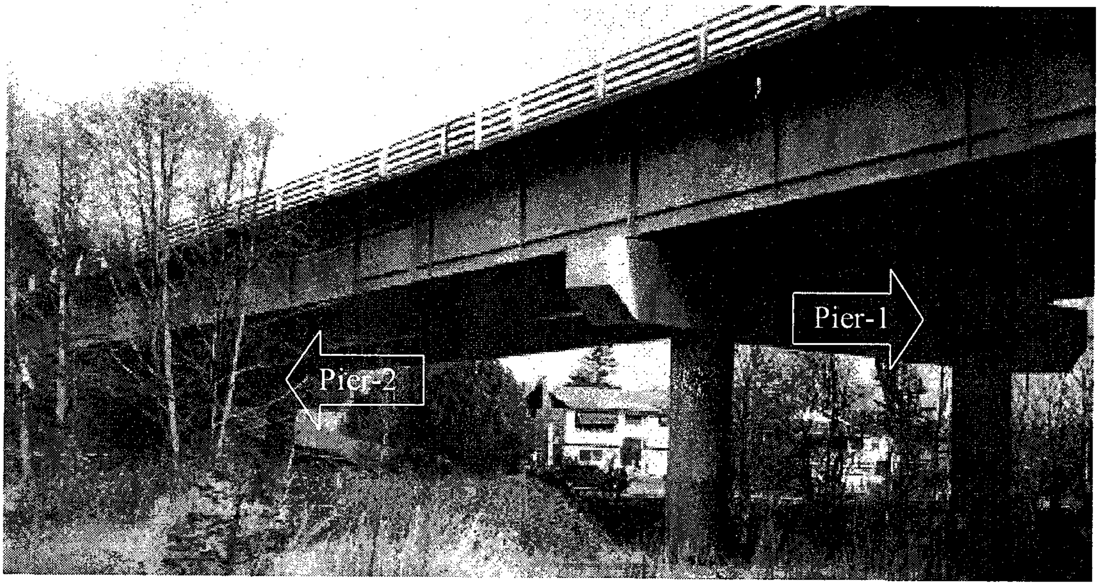

Figure-5.1: Portage Creek Bridge at Victoria, British Columbia (Huffman et al., 2006) today's standards (FHWA, 1995), (CAN/CSA-S6-88, 1988), (CAN/CSA-S6-00, 2005). It does appear, however, that consideration was given to some seismic aspects, as evidenced by a review of the drawing details. The bridge is classified as a Disaster Route bridge and was retrofitted to prevent collapse during a design seismic event, with a return period of 475 years (i.e., events having 10% probability of occurrence in 50 years). This Bridge was considered for the seismic evaluation of the as-built structure of Portage Bridge, design of the GFRP wrap system for strengthening the deficient columns, and development of an integrated structural health monitoring system for the bridge.

Here conventional materials and methods were used to retrofit most of the bridge. Based on dynamic analysis of the bridge, it was determined that the two tall columns of Pier No. 1 as shown in Figures-5.1, would form plastic hinges under an earthquake. As a result of these hinges, the short columns of Pier No. 2 would attract additional shear due to redistribution of internal forces. A non-linear static pushover analysis indicates that the short columns would not be able to form plastic hinges prior to failure in shear. Therefore, it was decided that Fiber Reinforced Polymer wraps (FRPs) should be used to strengthen the short columns for shear without increasing the moment capacity. The FRP wraps and the monitoring system were designed and implemented under the technical leadership of ISIS (Intelligent Sensing for Innovative Structures) Canada (Huffman et al., 2006).

5.2 SHM System

This bridge is one of 36 demonstration projects across Canada sponsored by ISIS Canada Research Network to assess the performance of FRP and the use of FOS (Fiber Optic Sensors) for structural health monitoring (SHM). The use of a SHM system is investigated for the purpose of determining the structural performance of the bridge for the various traffic loads and seismic loads carried by the bridge. This is also one of the practical examples of the implementation of intelligent sensing for the remote health monitoring of the seismic strengthened pier of the Portage Creek Bridge. The two columns of the bridge Pier-2 were strengthened with GFRP (Glass Fiber Reinforced Polymer) wraps with a total thickness of 5 mm. In order to monitor the response of the pier under an earthquake, eight bi-directional rosette type strain gauges and four long gauge fiber optic sensors have been attached to the outer layer of the GFRP wrap on each of the two columns as shown in Figure-5.2. In addition, two 3-D Crossbow accelerometers were installed on the pier cap above the columns. A durable weatherproof aluminum box with a safety lock has been installed on the pier cap to house a remote monitoring system. The National Instrument data acquisition system facilitates the remote transfer of data from the instrumented bridge. Through cable modem, the recorded data are transferred to engineer's office. The data collected so far has proved the reliability of the instrumentation system (Huffman et al., 2006).

This project assesses the performance of FRP for seismic strengthening. Fiber optic sensors (FOS) integrated with strain gauges and accelerometers are being used for structural health monitoring (SHM) remotely. Measured data from the various sensors that are integrated into the SHM system determine the structural performance under in-service conditions, diagnose faults, and quantify the risk of failure. Seismic strengthening with FRP is an effective and economical method of retrofitting for improvement in ductility and base shear resistance of existing structures. The seismic strengthening and instrumentation for remote monitoring was done in collaboration with the bridge consultant, Sargent and Vaughan Engineering Ltd., Victoria, and the bridge owner, the BC Ministry of Transportation. Using its innovative technology, ISIS Canada assisted with the retrofit involving GFRP wraps to strengthen the short columns.

The remote monitoring system provides current sensor status, and downloads measured data for analysis via internet. It also issues a warning when the allowable limit is exceeded. The SHM system of this bridge including the smart sensors and internet communication technology gives an example of how to monitor the health of large civil structures remotely.

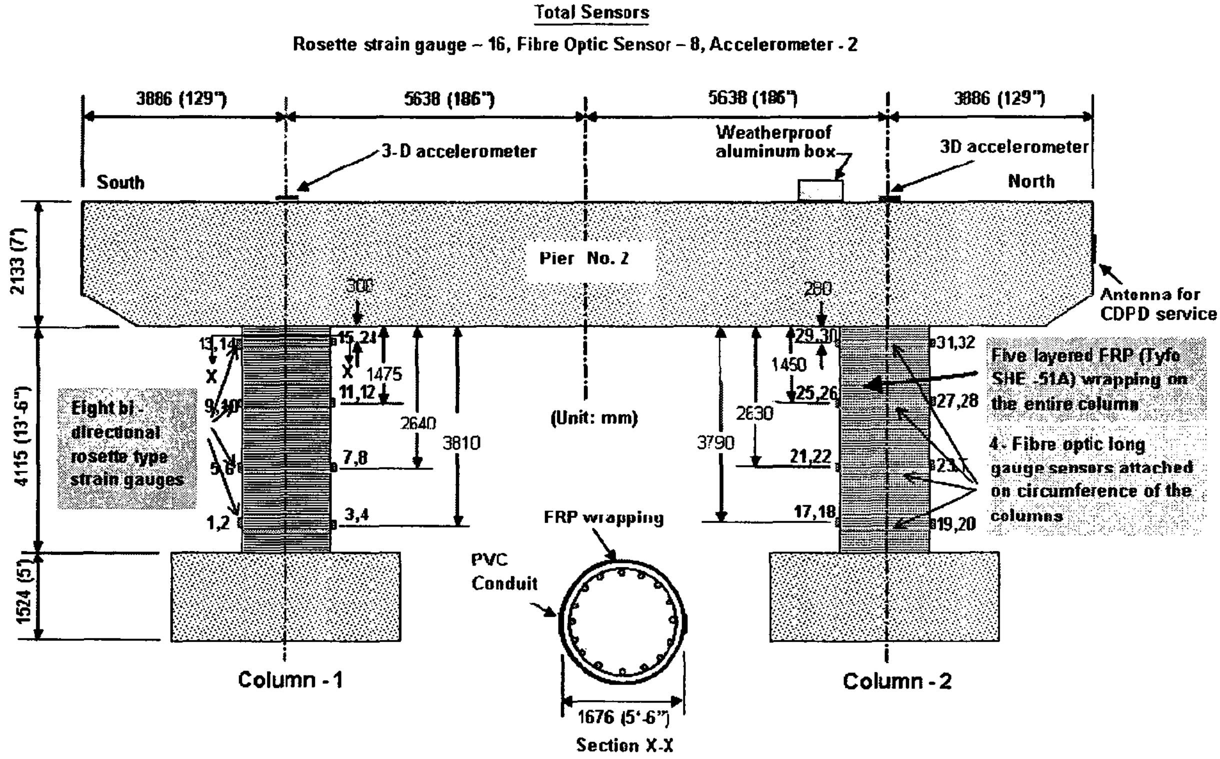

Figure- 5.2: Elevation of Pier no.2 (Short Columns) with sensor locations (Huffman et al., 2006)

5.2.1 Instrumentation for Remote Monitoring

ISIS Canada has made significant progress in the past seven years in the field of remote monitoring. Conventional sensors along with fiber optic sensors enhance the accuracy of the measurement factors. The following sensors were installed in the Portage Creek Bridge for remote structural health monitoring (Mufti *et al.*, 2000):

- Eight bi-directional electrical strain gauge rosettes on each column
- Four long gauge fiber optic sensors on each column
- One 3-D accelerometer on top of the pier cap of each column.

An elevation view of the instrumented Pier No.2 is shown in Figure-5.2 with sensor locations. Initially, eight Data Dolphin Data Loggers (Optimum Instruments Inc., 2000) each with four precision channels were installed as a multi-drop unit to collect strain data from the sensors. Optimum Instruments Inc., who supplied the data loggers, was, unfortunately, unable to adapt their system to the accelerometers and fiber optic long gauges. Other limitations of the system were:

- Limited data storage capacity
- Minimum data record interval of 10 seconds which is inadequate for dynamic events such as earthquakes
- Only one data logger can be connected at a time, so only four channels of data can be viewed at a time
- Significant download burden and system down time.

A high speed internet line was installed and a continuous data feed was established to the ISIS central server. A traffic camera has also been installed to synchronize the traffic picture with the strain data. A thick-walled lockable and weatherproof aluminum box houses all necessary instruments and connecting cables. The box is 450 (18″) × 440 (17.5″) × 1400 (56″) mm (width × height × length) and sits on the north end of the pier cap. The BC Ministry of Transportation has installed a power supply outlet, and the cables and wires are protected with PVC conduits and junction boxes.

As shown in Figure 5.2, the accelerometers were placed on the pier-head as the center of mass of the bridge deck and the pier-head is close to this position. The actual center of mass is at a higher elevation where the accelerometer cannot be placed due to practical constraints. The current location provides accessibility to the accelerometers and proximity to the data logger, which is placed on the pier-head. An onboard computer is also connected to the data logger in order to connect to the internet and to manage it remotely. Two accelerometers on the pier-head over each column are sufficient to capture the lateral vibration characteristics of the retrofitted columns and also to provide additional information about the torsion effects of the deck on the columns.

5.2.2 Sensor Description and Installation

Fiber Optic Sensors:

Fiber optic long gauge sensors developed by Fox-Tek Inc., Toronto, in collaboration with ISIS Canada were installed on the bridge. A 25 mm wide band along the periphery of the column at four levels was rubbed clean and all traces of dust removed. As is shown in Figure 5.2, five-layers of GFRP wrap were applied and allowed to harden. Then strain gauges and fiber optic sensors were attached to the outer surface of the wrap.

The long gauge system measures path displacement between two points on the structure by means of two mirrors formed on a fiber optic lead. This system works on the principle of low coherence interferometry using a light emitting diode. The light from the diode is split in two, travels two different path lengths and is then recombined at a photo detector. The measurement obtained is the total displacement over the gauge length of the sensor. First, a smooth surface was prepared by wiping with alcohol to remove all impurities. The sensor coil was gradually opened and taped to the prepared surface on the column. M-Bond adhesive with epoxy resin was used to bond the sensor to the FRP wrapping at predetermined locations.

Strain Gauge Rosettes:

A strain gauge rosette is meant to measure normal strains along two orthogonal directions. Two element 90° Micro-Measurements CEA-Series strain gauges of 350Ω resistance were installed on each column, as shown in Figure 5.2. A special clamping system was devised to ensure proper bonding between the FRP and the strain gauges. The device was comprised of a steel wire rope, turnbuckles, and wooden pressure pads. The wire was wound around the column. After the gauge was bonded to the column surface, the pressure pad was placed on the top of the gauge and the clamping pressure was applied with turnbuckles. Each strain gauge came with two pairs of 20 mm long pre-soldered short lead wires and sufficient extension wires. Although electrical strain gauges are not durable and unsuitable for long term monitoring, they are often used along with fiber optic sensors for the initial cross validation. Fiber optic sensors are difficult to install because of their brittleness and fragility. Since it was one of the early projects employing the integrated SHM system, the bidirectional electrical strain gauges were installed along with the long gauge fiber optic sensors along the circumference.

Accelerometers:

M Series Cross Bow 3-D accelerometers were selected for this installation. The M Series sensing element is a silicon micro-machined capacitive beam that is extremely small in die area and rugged. The capacitive beam is held in force balance for full-scale non-linearity of less than 0.2%. The M Series accelerometer operates on a single +5 V, DC power supply. Aluminum base plates were glued to the center of the columns at the top of the pier cap concrete surface. The accelerometer and cables were covered with a junction box and connected to the main conduit through a PVC pipe.

5.2.3 Data Acquisition & Harvesting

The Data Acquisition System is PC-based and equipped with National Instrument SCXI signal conditioning system and a NI A\D card (Figure 5.3). The sensing signals are noise compressed and amplified by the signal conditioning system and multi-scanned by the A\D card. A Lab VIEW application program is continuously running on the PC as a data collector and publisher. The data is collected at a rate of 32Hz and published to the data socket server every 5 seconds. The data socket server is located in the on-site PC and has the same IP (Internet Protocol) address as the publisher. On the central control site, the data server is a Dell SC1400 Windows 2000 server. A Lab VIEW application program is running continuously to subscribe the data from the publisher. Every 5 seconds the new data is transferred and real time signal processing is applied to the data. FFT is implemented to analyze the output from the accelerometers. The data server generates a data file every day for this monitored structure. These data are preserved in the ISIS Canada website. For the analysis purpose we harvested strain gauge readings for the period 2003 to 2006 depending on the availability of the data. First of all we considered the data of 16 strain gauges and one temperature gauge installed it on column C2 of pier2. From these data sets we made one input data sets and one target data set. Finally we used these data sets for training of the neural networks.

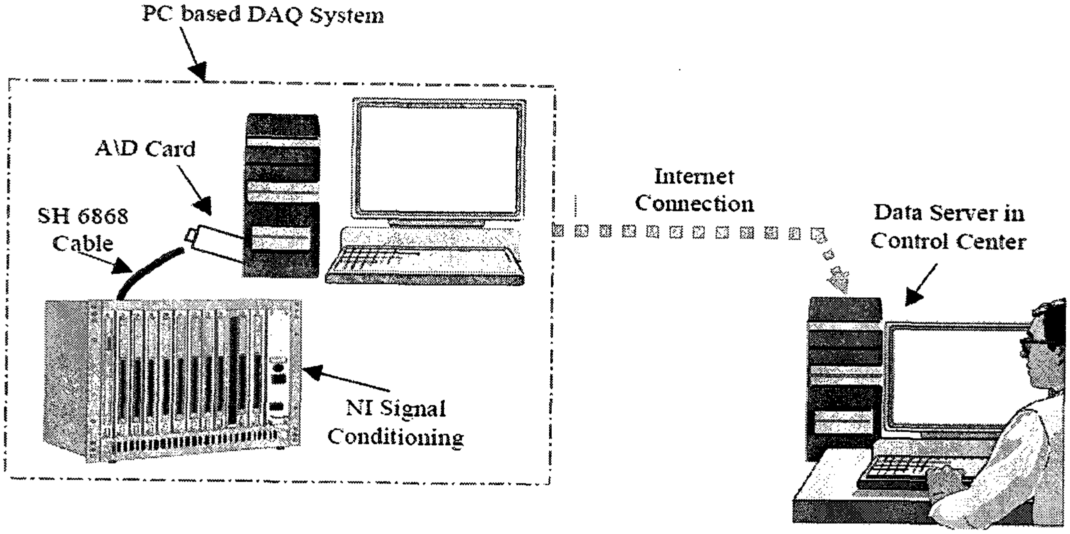

Figure-5.3: Diagram of the data acquisition and remote monitoring system (Huffman et al., 2006)

CHAPTER-6

Application of the proposed methods to the Portage Creek Bridge

6.1 Tracking the changes in data pattern

For observation of tracking the changes in data pattern, different neural networks have been trained and represented them with yearly data, seasonal data and monthly data following the methods as described in Section 4.3. Detail descriptions of these activities are described in the following subsections. This verification helped us to develop an idea about the training and tracking performance of the networks and identifying the pattern of sensor data.

6.1.1 Yearly changes in sensor data pattern

The objective of this analysis is to find out how ANN can track the changes in data pattern in a continuous monitoring system. Monitoring data for a year (2006) from the Portage Creek Bridge is divided into different segments of 3 to 4 days each. Data from 15 strain gauges and one temperature gauge are used as inputs and the remaining sensor data is used as target. All these sensors are located in one of the instrumented columns (Column C-2 of pier-2) of the bridge (Figure-5.2). The per minute data have been used for training and simulating the networks. The methodology has been explained in section 4.3. One network has been trained taking 16 sensors (including one temperature gauge) as input (4800 data for each sensor of January 1-5, 2006) and one sensor as target (4800 data of January 1-5, 2006). Then this trained network has been represented with new inputs from similar sensors data of January 6-9 (4122 data), January 9-12 (4051 data), February 9-12 (4320 data), February 12-15 (3116 data), March 1-5 (4752 data), March 12-16 (5199 data), April 6-8 (3568 data), May 1-4 (4320 data), May 5-8 (4295 data) and June 1-4 (4320 data). Plots of these network outputs and the actual target data are shown in Figure-6.1a to 6.1j as follows.

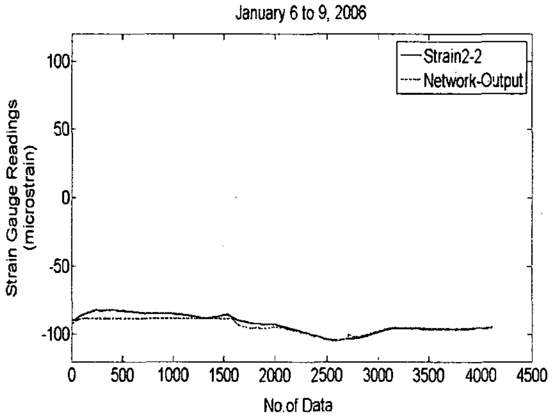

Figure-6.1a: Tracking changes in data pattern from January, 6-9 ($R^2 = 0.826$)

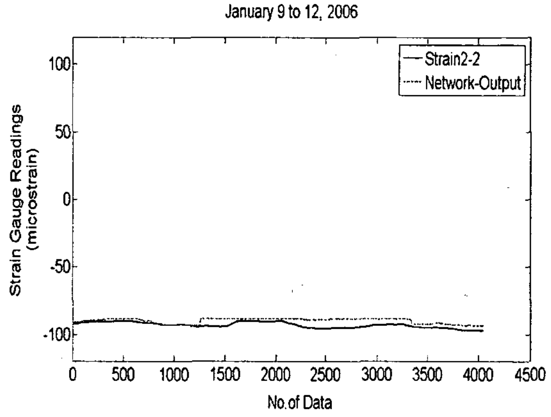

Figure-6.1b: Tracking changes in data pattern from January 9-12 ($R^2 = -1.968$)

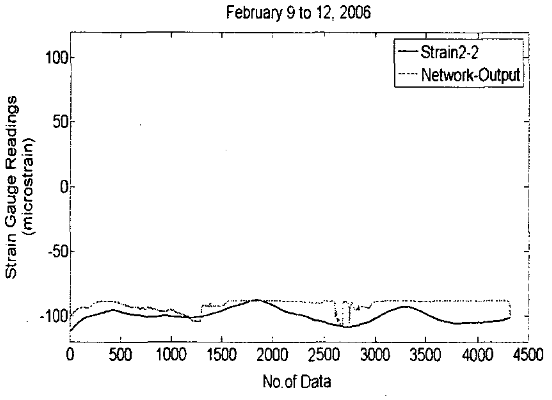

Figure-6.1c:Tracking changes in data pattern from February, 9-12 ($R^2 = -2.9$)

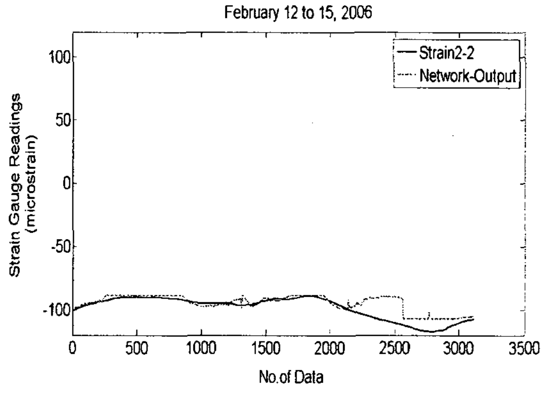

Figure-6.1d: Tracking changes in data pattern from February, 12-15 ($R^2 = 0.422$)

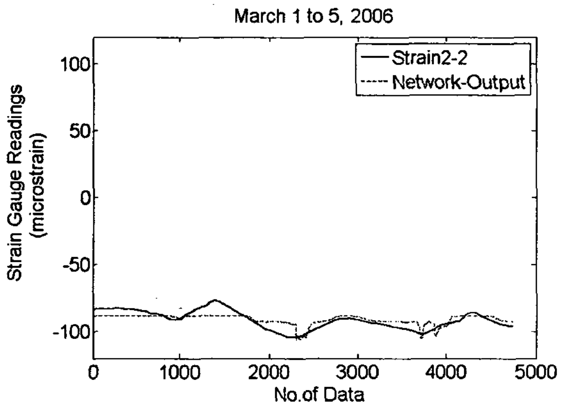

Figure-6.1e: Tracking changes in data pattern from March, 1-5 ($R^2 = 0.505$)

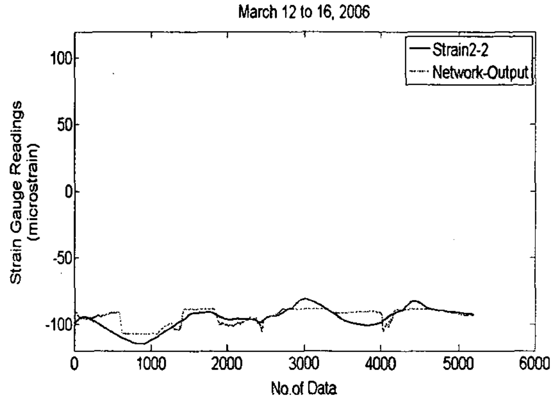

Figure-6.1f: Tracking changes in data pattern from March, 12-16 ($R^2 = 0.55$)

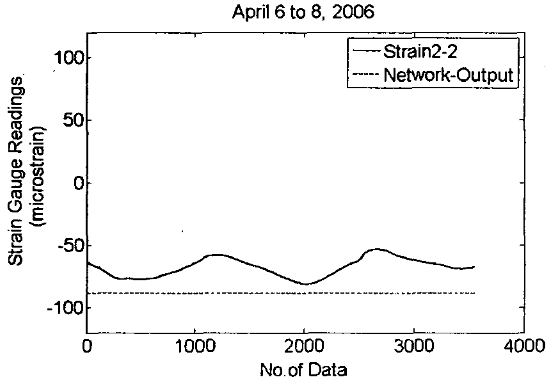

Figure-6.1g: Tracking changes in data pattern from April, 6-8 ($R^2 = -7.542$)

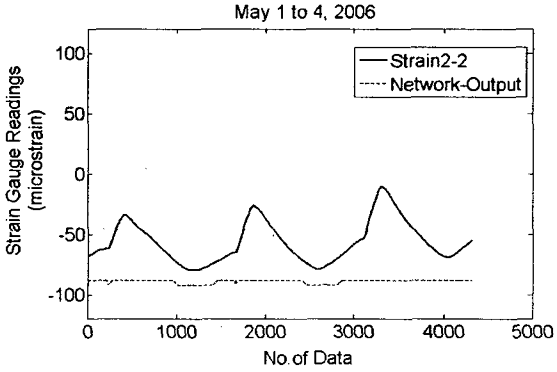

Figure-6.1h: Tracking changes in data pattern from May, 1-4 ($R^2 = -3.557$)

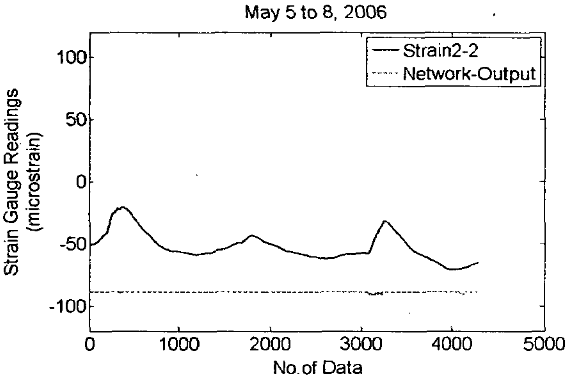

Figure-6.1i: Tracking changes in data pattern from May, 5-8 ($R^2 = -10.067$)

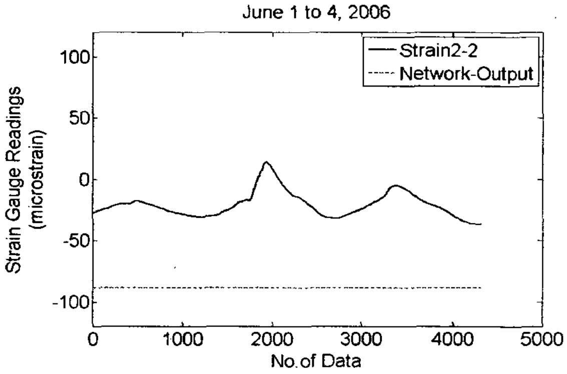

Figure-6.1j: Tracking changes in data pattern from June, 1-4 ($R^2 = -42.343$)

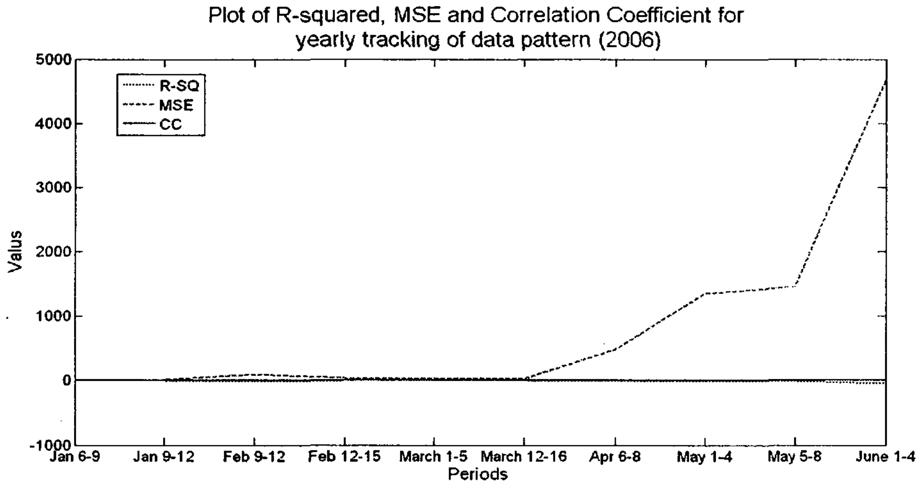

Figure-6.2: R-Sq., MSE and CC for tracking yearly changes in data pattern (2006)

Discussions

- In this analysis we have observed that the fitness between the network outputs and the actual data remains consistent until certain period of time (this can be observed from Figures-6.1a, 6.1b, 6.1c, 6.1d, 6.1e & 6.1f) and after this period the consistency in fitness decreases gradually with time (this can be observed from Figures-6.1g, 6.1h, 6.1i & 6.1j). Also this can be observed from plot of R^2 , MSE and CC in Figure-6.2

- There might be different reasons behind this inconsistency in data pattern. One of the possible reasons is that the relationship of data patterns among various sensors changes with seasonal changes. If the network is retrained progressively (i.e. dynamic) at a regular interval, it can track the patterns for a long time. Also the difference between the simulated and actual data can provide a measure of seasonal changes when the network is not retrained (i.e. static). This observations and ideas have been applied in our next analyses and the results are discussed in the following subsections

6.1.2 Seasonal changes in data pattern (Winter and Summer)

In this part of the analysis we have observed the seasonal changes in the data pattern during winter and summer of different years. For winter we considered per minute data for the months of January 2004, 2005 and 2006. In addition to this for summer we considered per minute data for the months of July 2003, 2004, 2005 and 2006. Then we converted per minute data into per hour data by averaging. For observation of changes in winter data firstly we used the per hour data of January, 2004 for training of the network. After this we tested this trained network with per hour data of January, 2005 and 2006 to track the changes in the pattern. Also for changes in summer data we trained the network with the per hour data of July, 2003. Then we tested this trained network with new inputs (per hour data) of July, 2004, 2005 and 2006.

Changes in winter (January/05, 06) data pattern:

The network was trained taking per hour data from January/04 as input and target. In this case 15 strain gauges and one temperature gauge reading were taken as input. On the other hand the data of strain1_1 was considered as target. The network was created by MATLAB software considering 4 hidden neurons in the hidden layer. The number of hidden neurons was selected simultaneously by changing the number of hidden neurons and checking the training performance with several trials. Finally, the trained network was tested with new inputs from January, 2005 and 2006. The output generated from the network is compared with the actual values which are shown in Figure-6.3a and 6.3b. The network training accuracy was verified by validating the trained network with the training input and plotting the output vs. actual as shown in Figure-6.3. For each plot the R-squared, mean-squared error (MSE) and correlation coefficients (CC) are calculated for comparison and assessment.

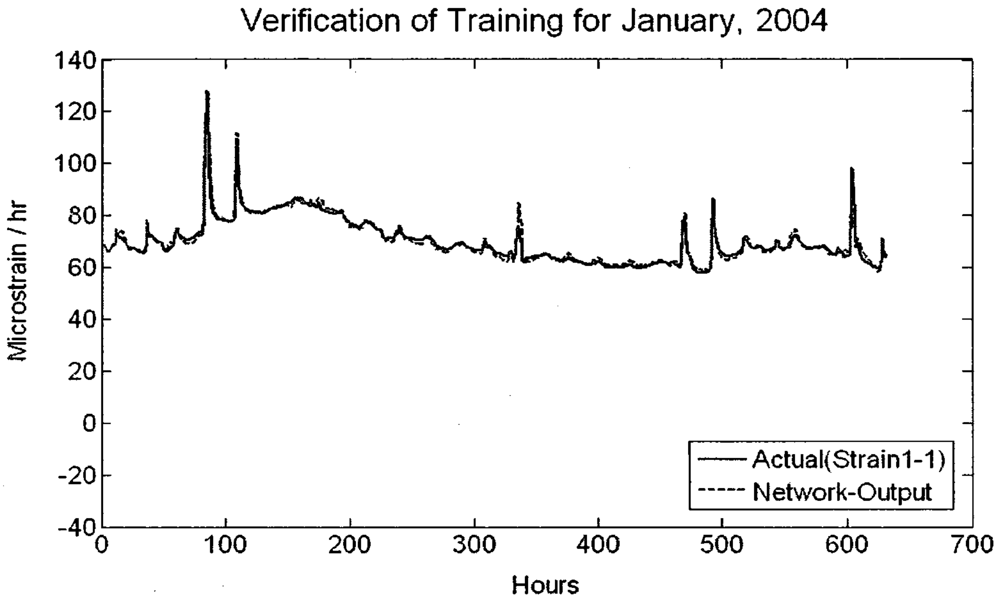

Figure-6.3: Verification of training accuracy by simulating the network with training input data of January, 2004. $R^2 = 0.965$, MSE= 2.527 & CC= 0.984

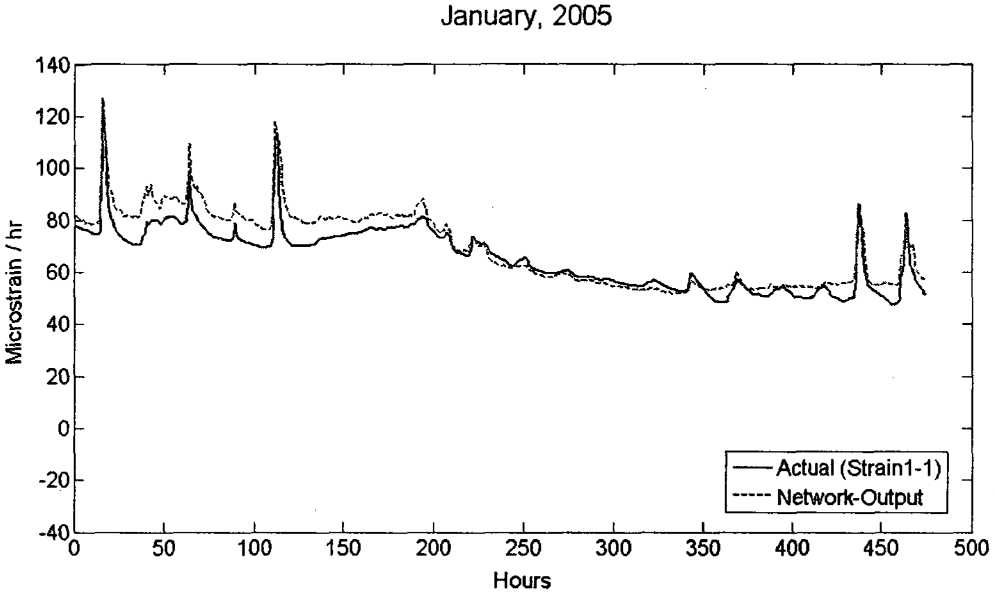

Figure-6.3a: Simulation of the input data of January, 2005. $R^2 = 0.754$, MSE= 34.662 & CC= 0.962

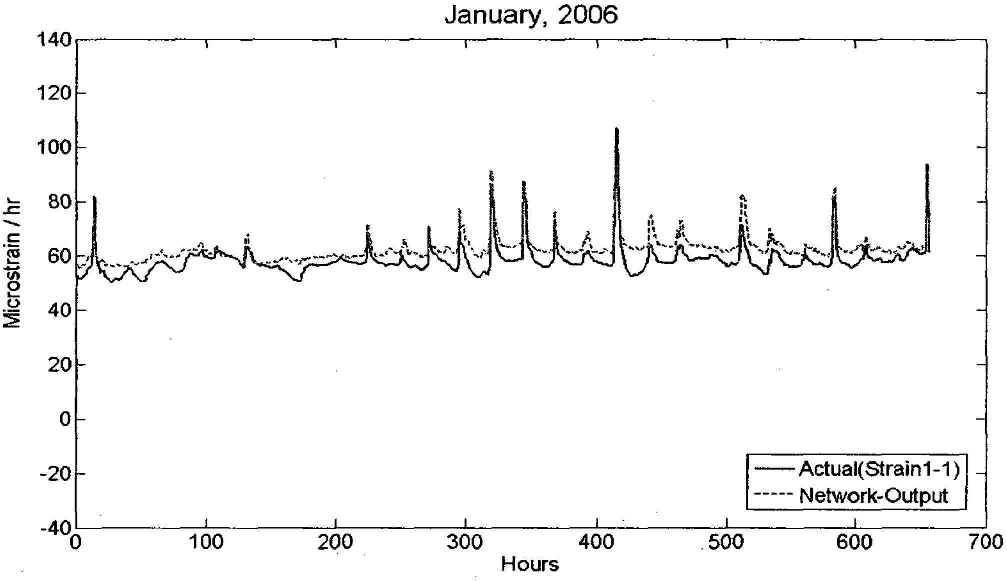

Figure-6.3b: Simulation of the input data of January, 2006. $R^2 = 0.085$, MSE= 24.893 & CC= 0.872

Changes in summer (July/04, 05, 06) data pattern:

The network was trained taking input and target from per hour data of July/04. In this case 15 strain gauges and one temperature gauge readings were taken as input data. On the other hand the strain1_1 reading was considered as target. The network was created by MATLAB software taking 8 hidden neurons in the hidden layer. In this case the number of neurons in the hidden layer was also selected optimizing the training performance by trial. The plots of the network output vs. the present target (actual data of Strain1_1) are shown in Figure-6.4a, 6.4b & 6.4c. The plot of the training verification is shown in Figure-6.4.

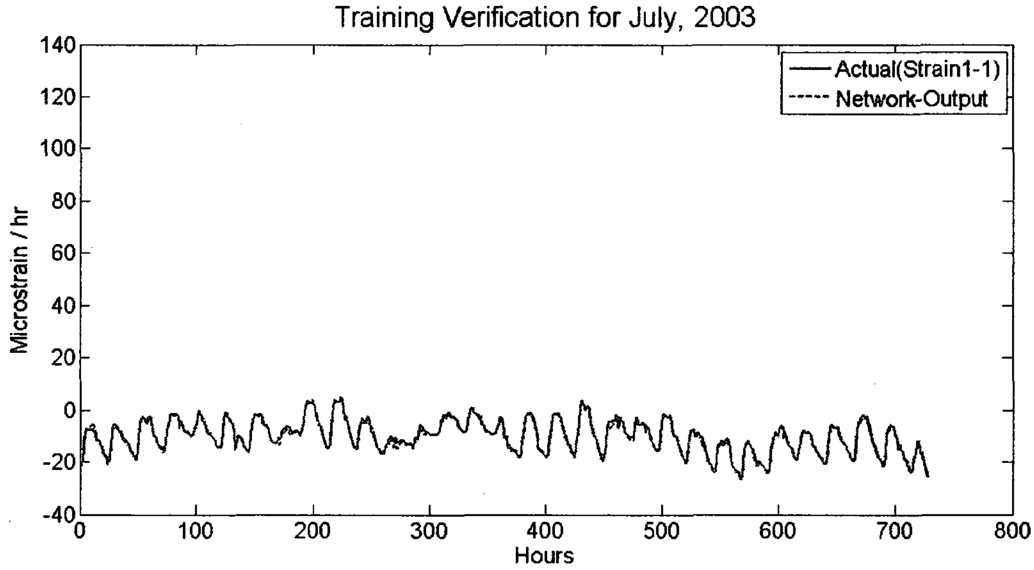

Figure-6.4: Verification of training accuracy by simulating the network with training input data of July, 2003. $R^2 = 0.982$, MSE= 0.716 & CC= 0.992

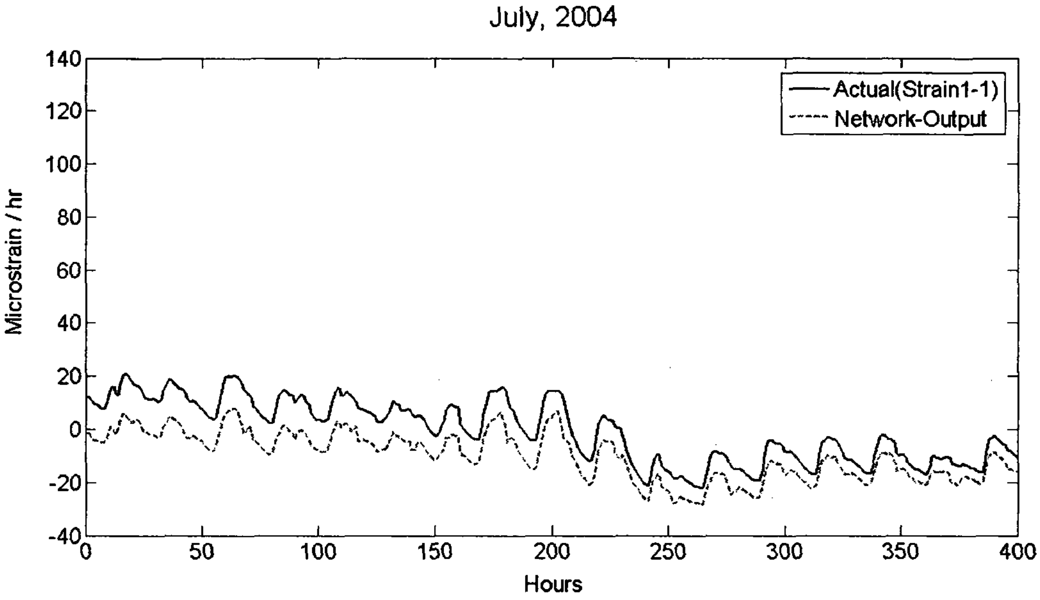

Figure-6.4a: Simulation of the network with input data of July, 2004. $R^2 = 0.278$, MSE = 98.462 & CC = 0.987

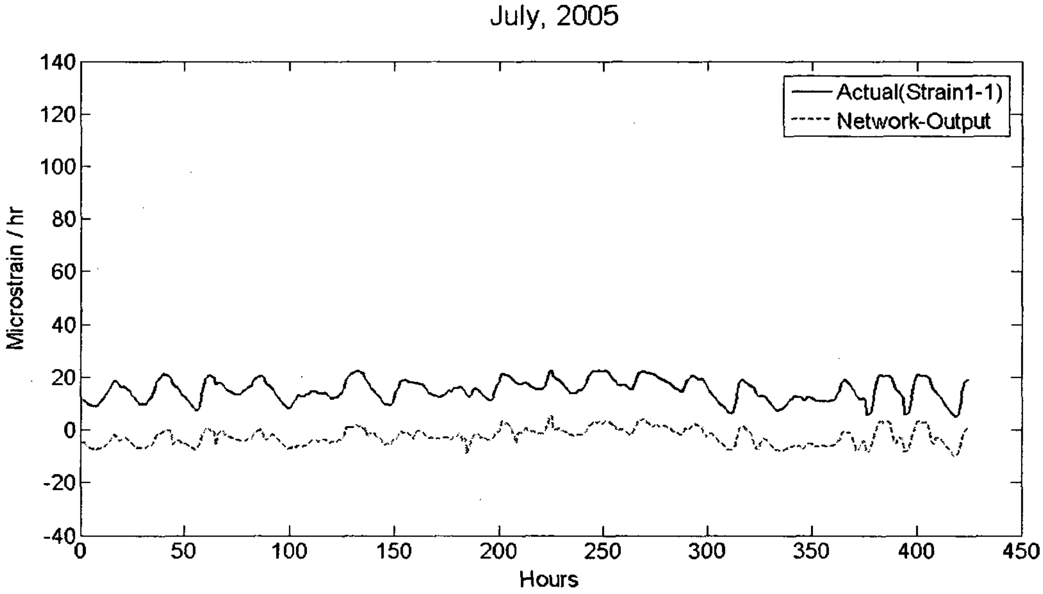

Figure-6.4b: Simulation of the network with input data of July, 2005. $R^2 = -17.816$, MSE = 317.499 & CC = 0.908

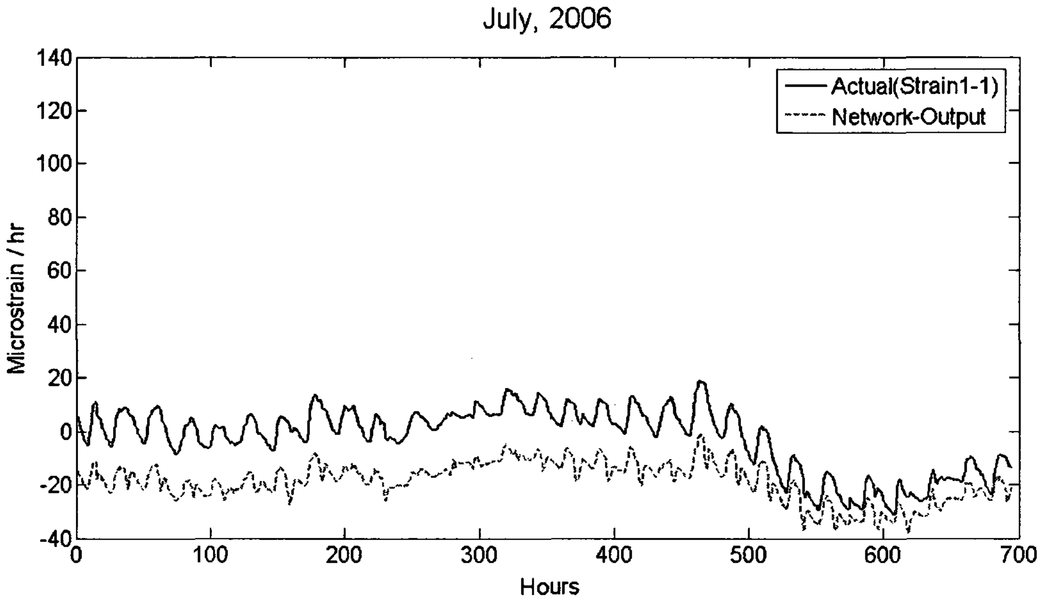

Figure-6.4c: Simulation of the network with input data of July, 2006. $R^2 = -1.122$, MSE= 297.157 & CC=0.919

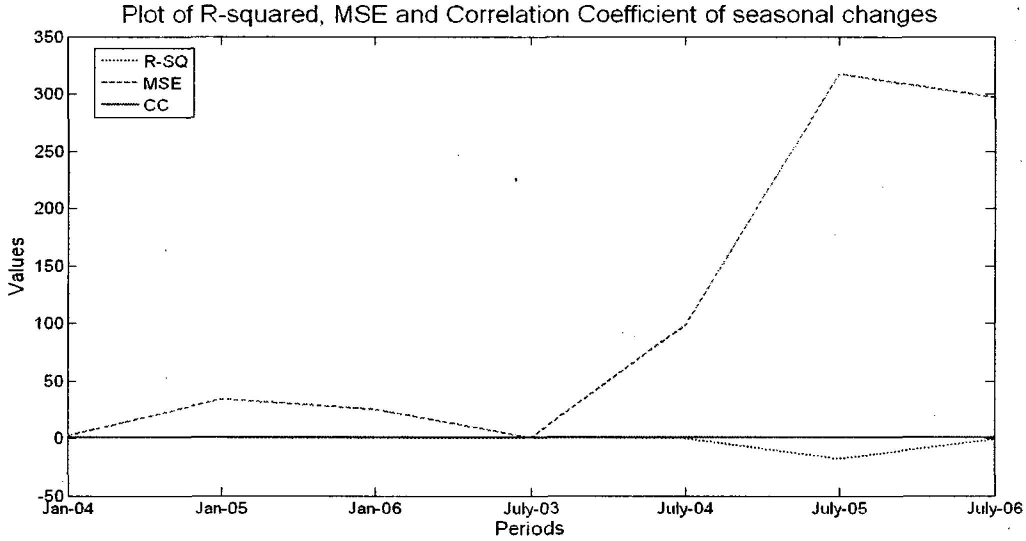

Figure-6.5: R-sq., MSE and CC for tracking of seasonal changes in data pattern

Discussion

- In this analysis, we have trained two separate networks; one for tracking the changes in data pattern in the winter and another for tracking the changes in data pattern in the summer. Although we have trained two separate networks for tracking seasonal (winter and summer) changes in data pattern, even it cannot track the changes in data pattern for a long time. It can be observed from the plot of MSE in Figure-6.5

- From these observations we decided that it would have been more effective to train a network with maximum number of inputs during training so that it will be able to learn more varieties of data pattern that will help it to track future changes in data pattern immediately after the training more accurately. We applied this methodology in the following analysis.

- R^2 close to 1 means good fitting between the output and actual data. Also smaller value of MSE means the difference between the actual and output is minimum. On the other hand value of CC close to -1 or +1 means the pattern is good. This can be observed from Figures-6.3 and 6.4. Whereas in Figures-6.3a, 6.3b, 6.4a, 6.4b and 6.4c the simulated output and actual target seem to follow similar pattern, although they appear to be shifted. That's why CC and R^2 is good, but MSE is bad (Figure-6.5). These apparent shifts are perhaps due to the seasonal changes or ageing from year to year.

6.1.3 Monthly changes in data pattern (March, 2006)

In the previous subsection 6.1.2, we have seen that for tracking the changes in winter and summer data, the trained network losses the accuracy in tracking the data pattern if it is represented with data after a long period of training. To minimize this error we considered training of a network taking maximum number of input records and testing it with minimum number of inputs immediately after the period for which data is used in training. For this analysis we trained network-1 with per minute data for maximum period covering December/2005 (40800 data), January (44880 data) and February/2006 (32640 data) after converting to per hour data (total data neglecting zeroes =1738 nos. or 1738 hrs) so that the networks could be trained with maximum number of information in its memory. Then we tested this trained network with minimum number of new inputs data immediately after the training period i.e. firstly, we tested the trained network-1 with the data (574 nos. or 574 hrs) for March/2006 and this plot is shown in Figure-6.6a; secondly, we tested the trained network-1 with first 15 days data (287 nos. or 287 hrs) of March/2006 and this plot is shown in Figure 6.6b.

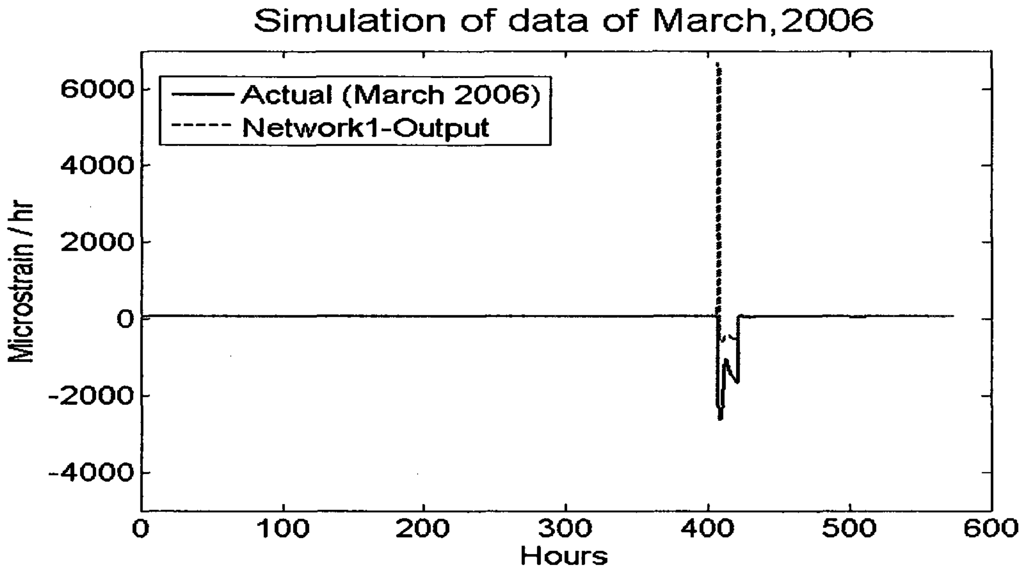

Figure-6.6a: Plot of correlation-ship for 30 days data (574 data) of March, 2006. R^2= -1.181, MSE=174261 & CC= -0.045

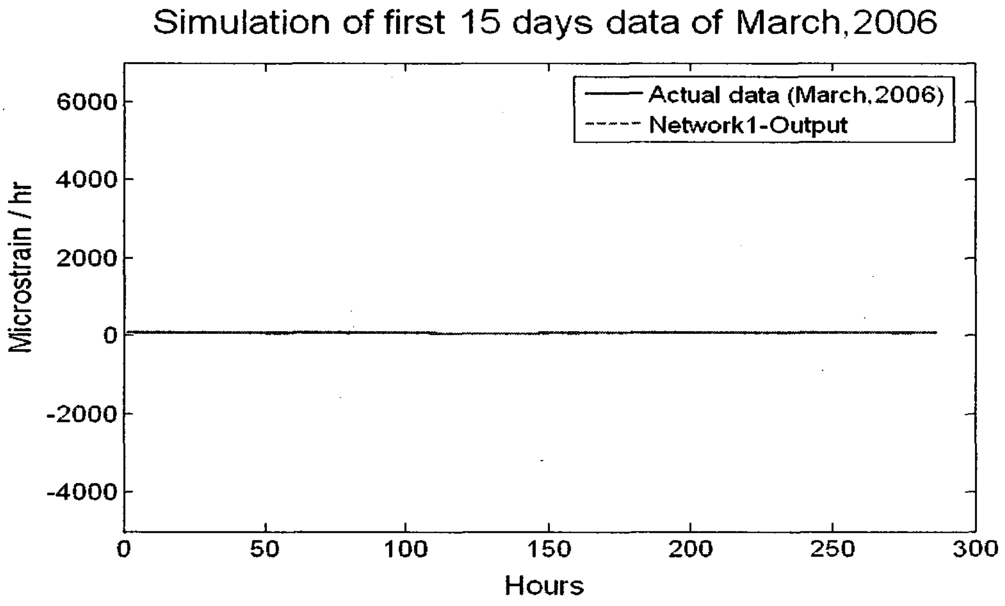

Figure-6.6b: Plot of correlation-ship for 1st fifteen days data (287 data) of March, 2006, R^2 =0.803, MSE=3.453 & CC=0.424

Discussion

- In this analysis we have seen that if a network is trained with a greater volume of input data in the training, the network can efficiently be trained and it can track the changes in data pattern more accurately if the trained network is represented with a small data range immediately after the training period.
- In the correlationship of 30 days data of March, 2006, we have seen from the plot (Figure-6.6a & 6.6b) that the data in 1st half of the month do not have any irregularities in the data pattern but the data in the 2nd half of the month has spike in it which might be due to the sensor malfunctioning or other structural changes at this particular location.
- The main goal of this analysis is to figure out which sensor data causes this spike in the data pattern. Initially, the identified sensor is treated as a defective sensor. Before detecting this defective sensor we have developed and verified an appropriate ANN algorithm that can successfully be applied for detecting the defective sensor. The step by step analyses are explained as follows:

6.2 Verification of data pattern in presence of defective sensor

In this section, how the data pattern changes in presence of defective sensor and their subsequent algorithms have been verified. This important feature of neural network has been analyzed and explained by a very simple method. In this analysis per minute data (37920 data points for a month) of the Portage Creek bridge has been considered for the period of January, 2004 that was produced from sixteen strain gauges and one temperature gauge located at column-2 of pier-2 of the bridge. Then we converted all these data of each sensor as an average output per hour (632 nos. or 632 hrs). This has been done to simplify the whole process so that their pattern of changes could easily be identified by using a neural network and which can be clearly visible from the plot of these results. The analysis procedure for identification of defective sensor from a group of sensors have been explained and performed in the following phases:

6.2.1 Network training:

Firstly, one sensor has been removed from sixteen selected sensors. Secondly, a feed forward network was built using MATLAB. This network was structured with fifteen neurons in the input layer and one neuron in the output layer. Also the number of hidden neurons was selected based on the performance and iteration of training. Then we trained up Network-1 taking 15 inputs (removing Strain1_2) and 1 target (Strain1_1) data.

We followed the same procedure and trained up Network-2, Network-3, Network-4, Network-5 and Network-6 by taking Strain1_1 always as target. The inputs for training of these networks were performed simultaneously by removing Strain2_1, Strain2_2, Strain3_1, Strain3_2 and Strain4_1.

6.2.2 Network simulation:

In this phase the trained networks have been tested making Strain4_1 defective by randomly selected data. This alteration in data has been made assuming that if Strain4_1 doesn't work properly due to some mechanical faults, then the readings of Strain4_1 will automatically be changed. After this, we tested all networks (Network-1, Network-2, Network-3, Network-4 & Network-5) with new inputs always including the rtandomly selected data of Strain4_1 and simultaneously removing Strain1_2, Strain2_1, Strain2_2, Strain3_1, and Strain3_2 following the procedure as described in subsection 4.4.1. Finally we tested Network-6 removing Strain4_1 from input data sets. All network outputs and their changing patterns in comparison to the actual data are shown in Figure-6.7 to Figure-6.12. In Figure-6.12, we observed that at this point we didn't notice any changes between the actual (Strain1_1) and the output from the Network-6. This is due to the elimination of altered input data of the defective Strain4_1 (data was randomly selected) from the sensors group.

From the plots of observations in Figure-6.7 to Figure-6.11, we could see that after training of Network-1, Network-2, Network-3, Network-4 and Network-5, when they have been tested each time with new inputs (that include altered data of Strain4_1), the outputs from all the networks have been shifted from the orginal target data (Strain1_1). But as soon as we eliminated the defective (i.e. altered) data of Strain4_1 and tested this input to Network-6, no discrepancy found in between the output and the actual target data (Strain1_1). This is clearly seen from the plot in Figure-6.12. It also indicates that the presence of defective sensor data causes some discrepancy in the data pattern as shown in Figure-6.7 to Figure-6.11 and the absence of it doesn't cause any changes in the pattern as shown in Figure-6.12.

In addition, for better understanding and evaluating this analysis we have calculated R^2 , MSE (Mean Squared Error) and CC (Correlation Coefficient) and presented them in the

Table-6.1. It should be noted that R^2 close to 1 means the fitness of the curve is good. Also R^2 less than 0 (negative) and more than 1 means the fitness of the curve is not good. In case of MSE if the value is higher it means the difference between the actual observation and the response from the model is more and if the value is smaller means the diffeerence is less. On the other hand the value of CC either close to -1 or 1 means the correlation is stronger.

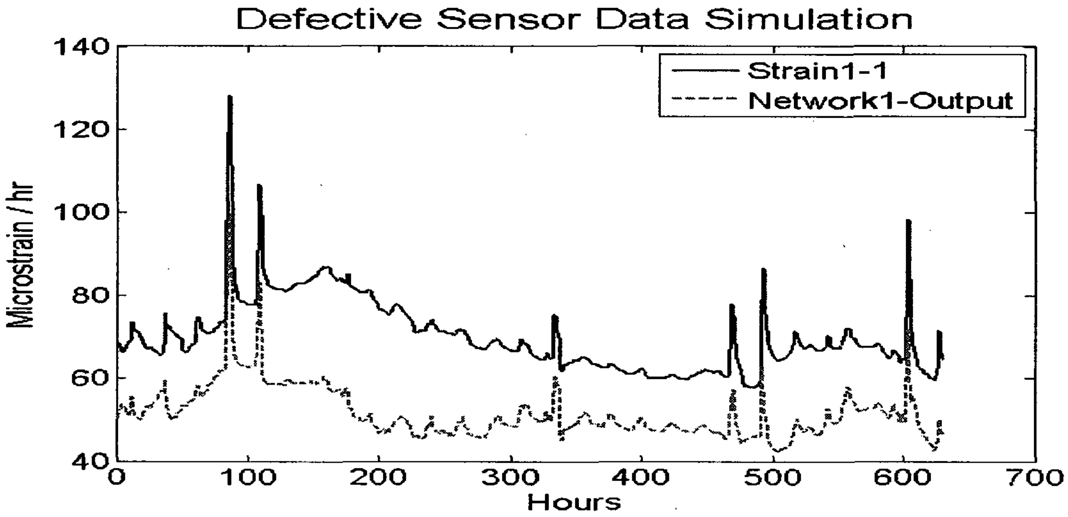

Figure-6.7: Simulation of Network-1 using the inputs by eliminating Strain1_2, Period-January, 2004

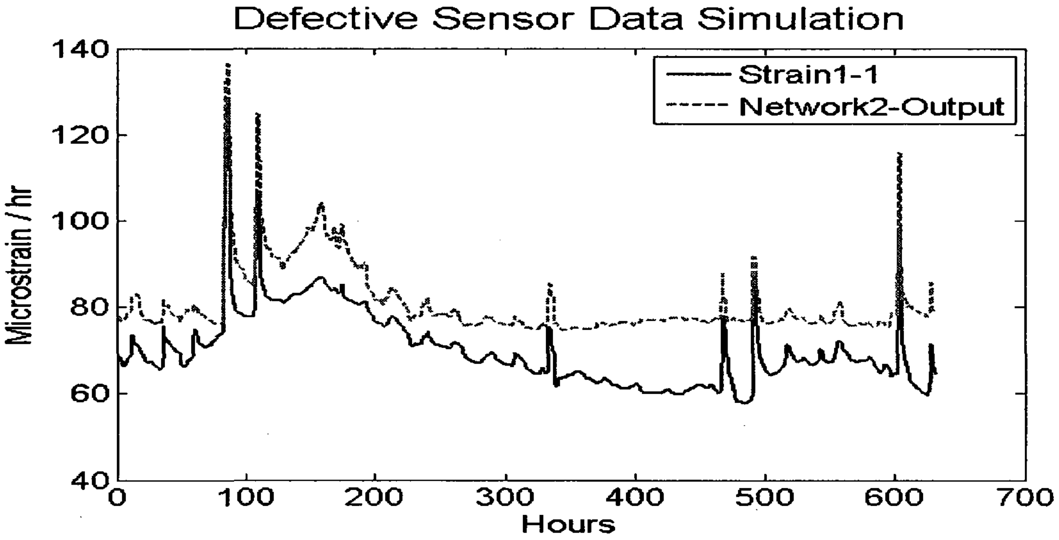

Figure-6.8: Simulation of Network-2 using the inputs by eliminating Strain2_1, Period-January, 2004

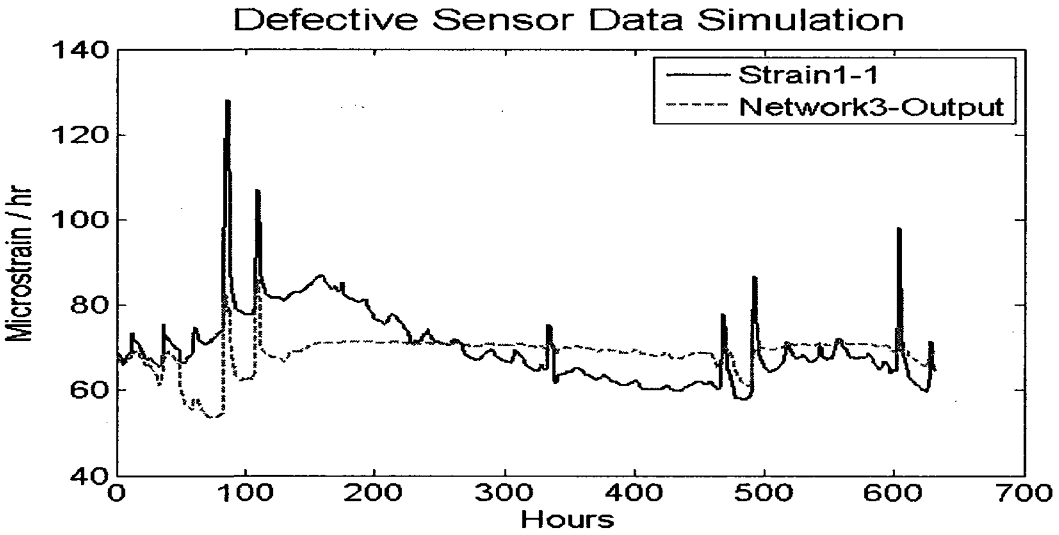

Figure-6.9: Simulation of Network-3 using the inputs by eliminating Strain2_2, Period-January, 2004

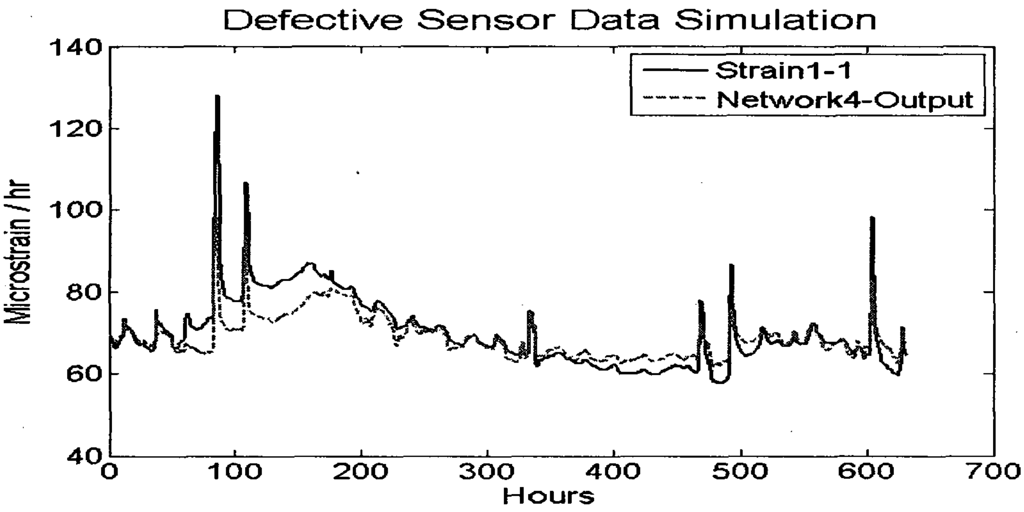

Figure-6.10: Simulation of Network-4 using the inputs by eliminating Strain3_1, Period-January, 2004

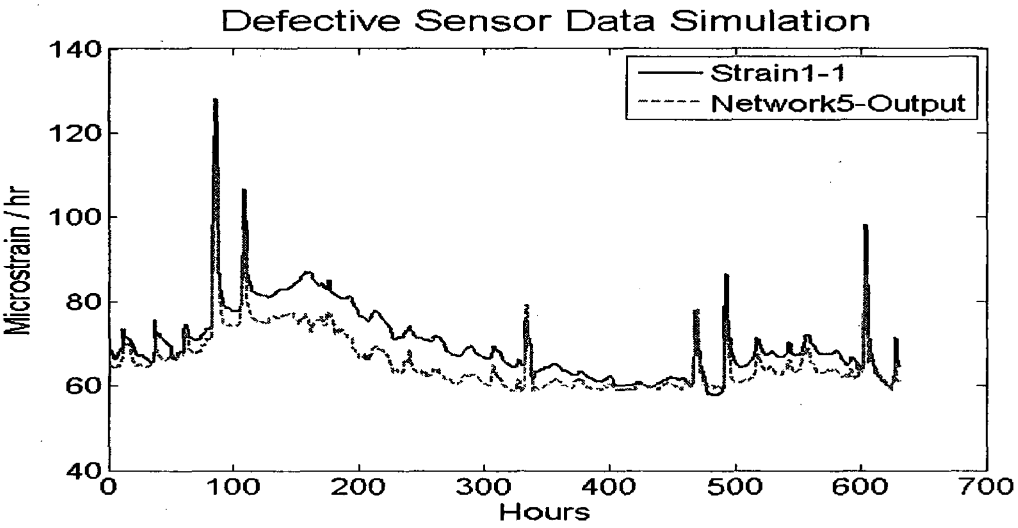

Figure-6.11: Simulation of Network-5 using the inputs by eliminating Strain3_2, Period-January, 2004

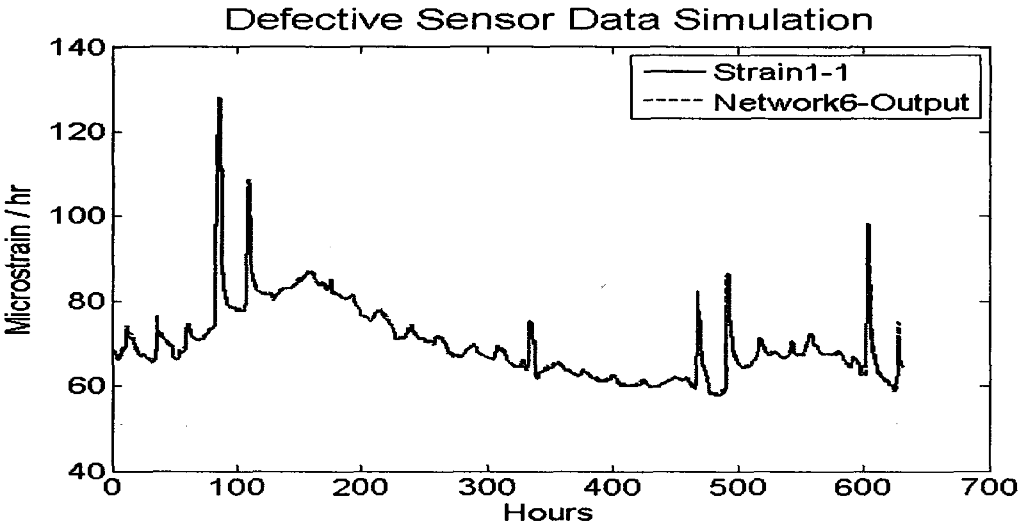

Figure-6.12: Simulation of Network-6 using the inputs by eliminating randomly selected data of Strain4_1, Period-January, 2004

Table-6.1: Table of correlation coefficient (CC), mean squared error (MSE) & R-squared

Name	Net-1	Net-2	Net-3	Net-4	Net-5	Net-6
CC	0.775	0.905	0.257	0.926	0.941	0.998
MSE	369.758	130.578	73.869	20.262	28.751	0.484
R^2	-4.044	-0.781	-0.007	0.723	0.607	***0.993***

Discussion

- From this analysis, we could certainly tell that the neural networks are able to recognize the defective sensor in a group of sensors based on the relationship of data patterns among a group of sensors. And this task can be performed by sequential elimination of sensors, traininig of individual network for each elimination and representing them with new inputs from similar sensors.

- From the above analysis, it has been observed that due to the presence of manually made defective sensor (SG4_1) in the inputs, a discrepancy in the correlation-ship always exists (it can be seen in the Figure-6.7 to 6.11).
- After removing the manually made defective sensor (SG4_1) from the inputs, there was no existence of discrepancy in correlation ship of the data pattern and this can be seen in the Figure-6.12.
- From this experiment we have developed the preliminary idea about how a neural network responds to the presence of defective sensor in a group of sensors. Also it gives us the idea that only when the defective sensor is removed the maximum R-squared value can be achieved (Table-6.1). These observations and algorithms have been applied for identification of defective sensors as described below:

6.3 Identify defective sensors from plots of R-Squared

6.3.1 Description

It has been observed from the previous analysis that for finding the response of ANN to the defective sensor data, one sensor has been removed simultaneously from each input to train model networks. Here, the similar procedure has been followed and plotted the R-squared values for each network outputs. This method has been entitled as sequential search method. The key steps in this method have been discussed in subsection 4.4.1. In the sequential searching of defective sensor, per minute monitoring data have been harvested from the SHM database recorded from 16 sensors and one temperature gauge installed in one column (Column-C2 of Pier-2) of the Portage Creek Bridge. Next these per minute data has been converted to average hourly data by averaging using excel sheet. Each input and target data has been pre-processed by de-noising using wavelet one dimensional multi-signal analysis for getting more accuracy in training. For pre-processing of each data, daubechies-4 (db4) wavelet at level-5 has been used. The steps that have been followed in sequential searching of defective sensor are outlined below:

- Preliminary identification of the presence of defective sensor in a group of sensors
- Development and training of model networks by sequentially taking out one sensor at a time
- Testing these model networks firstly with non defective sensors group and secondly making one sensor defective in the same group
- Finally plotting R-squared for data of training inputs, non defective sensor group and defective sensor group

6.3.2 Preliminary identification of the presence of defective sensor:

Generally, for detecting the defective sensor, first we need to be ascertained if there is any defective sensor present among the sensor group at all. For initial assessment of the presence of defective sensor we built and trained a model network taking all sensors (16 sensors) gauge including one temperature gauge data as input and one strain gauge data as target for the periods of December/05, January/06 and February/06. To verify the training of the model network, we represent the trained network again with the training input and calculated the R-squared value (Figure-6.13). This R-squared value will give us the idea about the accuracy of training and it works as a standard to compare other values of R-squared

Then we represented this model network with new input immediately after the training period i.e. first 15 days data of March/06. For this input if we don't see any significant changes in the data pattern or R-squared value in comparison to the standard value, it indicates that all sensors were functioning properly. But if we see significant changes in the data pattern, this will be considered as an indication of the presence of defective sensor or sensor malfunctioning. In reality we don't see any significant changes in the output signal when the model network was first tested with 15 days data of March/06 (Figure-6.14). For the verification of the response of an artificial neural network to the defective sensor data, we manually made one sensor (SG3_2) defective in the previous input data group and tested it again in the same model network. Now we can easily observe in Figure-6.14 that due to the presence of defective sensor data in the input data group, the output data pattern has been suddenly shifted in comparison to the actual data pattern. From this observation of plots (Figure-6.14) we can conclude that preliminary assessment of the presence of defective sensor can be done by using the ANN technique.

6.3.3 Development and training of model networks

In the previous step we have detected primarily the existence of defective sensor in a group of sensors (16 sensors and 1 temperature gauge). Now we need to identify the specific sensor that causes this change in the data pattern as shown in Figure-6.14. This could be done by sequentially taking out one sensor at a time from the new input data group and testing it each time to the model networks that have to be built and trained by taking out similar sensor at a time. For example, for testing of the 15 days input data of March/06 by taking out sensor SG1_2, we trained the model network by eliminating the same sensor (SG1_2) from training inputs of December/05, January/06 and February/06. Following this criteria we built and trained six networks by sequentially taking out six sensors (SG1_2, SG2_1, SG2_2, SG3_1, SG3_2 & SG4_1). For this training of model networks, we considered the training input and target data from December/05, January/06 and February/06.

6.3.4 Testing of the model networks

In this step we tested the model networks as discussed earlier with new inputs of first 15 days data of March/06 by taking out the similar sensor as in training of the model networks. In this analysis, for verification purpose we prepared three sets of new inputs i.e. training inputs (the inputs from Dec/05, Jan/06 & Feb/06), input without defective data of SG3_2 (March/06) and input with defective data of SG3_2 (March/06). Each set of new inputs were prepared by simultaneously taking out six sensors as we did during the preparation of six model networks.

6.3.5 Analysis of the output from the networks

Plots of R-squared for three sets of data (training, non defective sensor group and defective sensor group) give us the idea about the fitness of the curves. As we do not have any idea about the exact value of the data of defective sensor, we assumed three cases of the defective sensor group and plotted them to verify how ANN responds to these cases of defects.

Case-1: In this case the data of the sensor- SG3_2 was made manually defective keeping 20% data unchanged and 80% data set to zero. Then we tested the model networks with training, non defective sensors and defective sensors group. For each set of new inputs, we tested the six model networks with six inputs taking sequentially one sensor out. Also we plot R-squared values for these three sets of inputs as shown in the Figure-6.15. It could be observed from Figure-6.15 that when the defective sensor SG3_2 has been removed from the inputs with-defective data sets, the R-squared value coincides at one point with the R-squared for inputs without-defective data.

Case-2: Considering 20% data unchanged and 80% data changed to an arbitrary value of 100 micron. For this change the plots of R-squared value are shown in Figure-6.16.

Case-3: Considering 20% data unchanged and 80% data randomly changed by taking the value of percentage=500% =5, in the following formula:

$$DefectiveData = Originalvalue\left[1 + percentage(randomvalue - 0.5) \times 2\right]$$

For this changes the plot of R-squared value shown in Figure-6.17

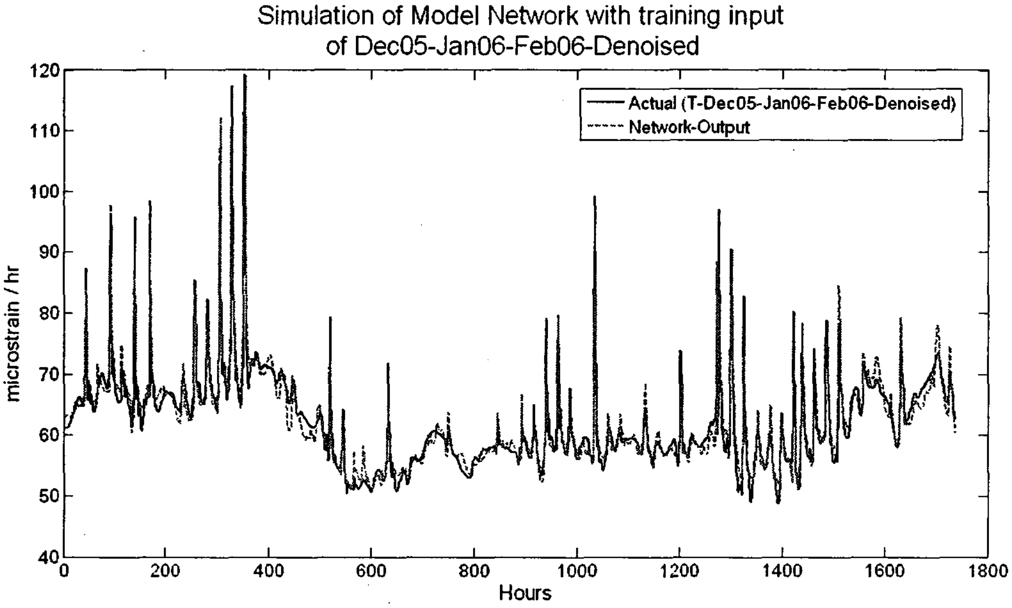

Figure-6.13: The plot of the output and the actual value when the model network was tested with training inputs. The R-squared= 0.954.

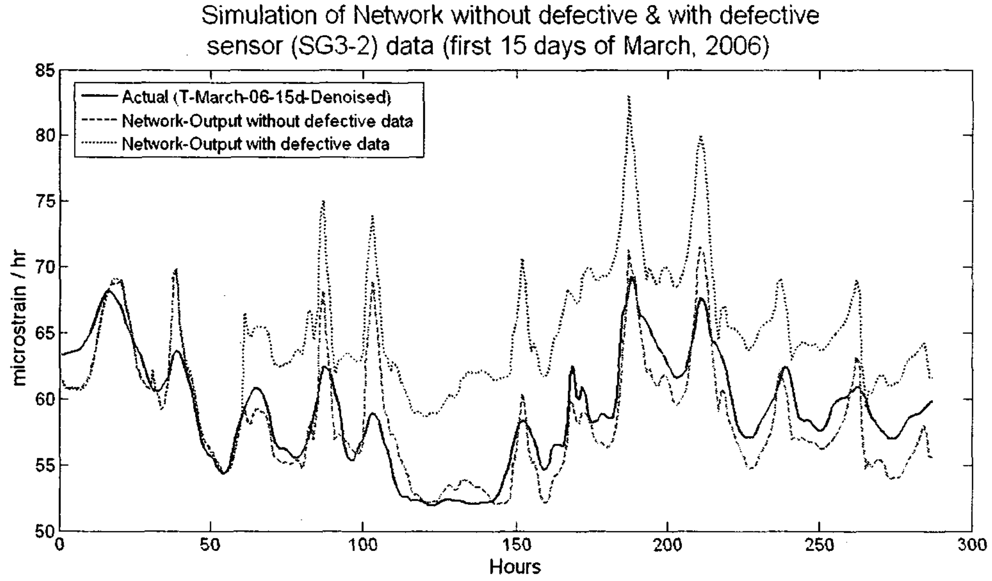

Figure-6.14: Plot of outputs of model network for new inputs with non defective and defective sensor (SG3_2) data (1st 15 days data of March/ 2006). R-squared (without defective data) = 0.707 and R-squared (with defective data) = -1.595

Table-6.2: Table of R-squared calculated from plots of network output and actual data for defective sensor Case-1

Sl. No.	Eliminated sensors	R-squared for training inputs	R-squared (In absence of defective sensor SG3_2)	R-squared (In presence of defective sensor SG3_2)
1	SG1_2	0.914	0.690	-2.878
2	SG2_1	0.911	0.522	-2.524
3	SG2_2	0.919	0.631	-1.706
4	SG3_1	0.929	0.691	-1.657
5	***SG3_2***	***0.922***	***0.739***	***0.739***
6	SG4_1	0.929	0.445	-1.632

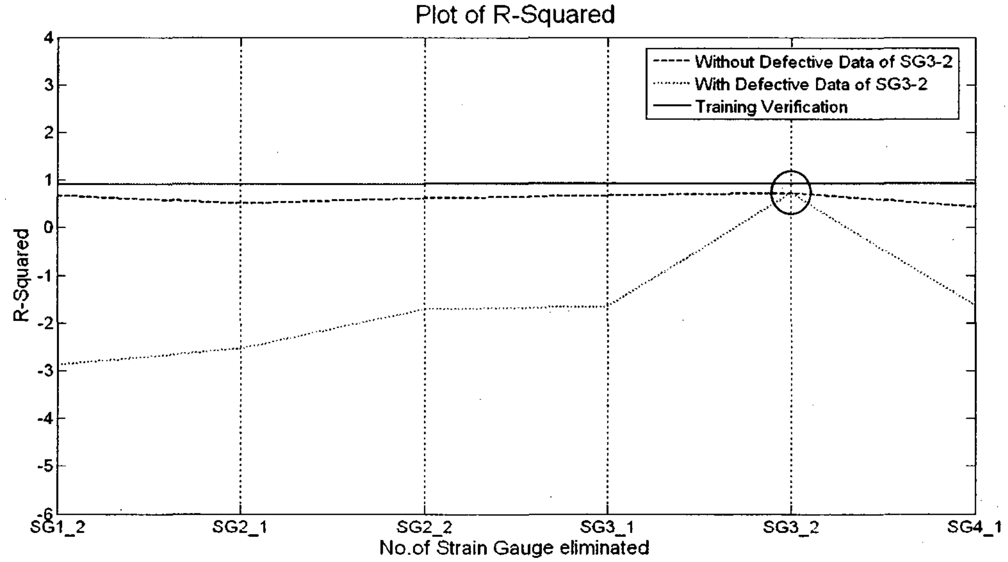

Figure-6.15: Plot of R-squared for inputs of non defective data of SG3_2, with defective data of SG3_2 and training inputs (Case-1)

Table-6.3: Table of R-squared calculated from plots of network output and actual data for defective sensor Case-2

Sl. No.	Eliminated sensors	R-squared for training inputs	R-squared (In absence of defective sensor SG3_2)	R-squared (In presence of defective sensor SG3_2)
1	SG1_2	0.914	0.690	-4.990
2	SG2_1	0.911	0.522	-4.937
3	SG2_2	0.919	0.631	-3.910
4	SG3_1	0.929	0.691	-3.534
5	***SG3_2***	***0.922***	***0.739***	***0.739***
6	SG4_1	0.929	0.445	-3.334

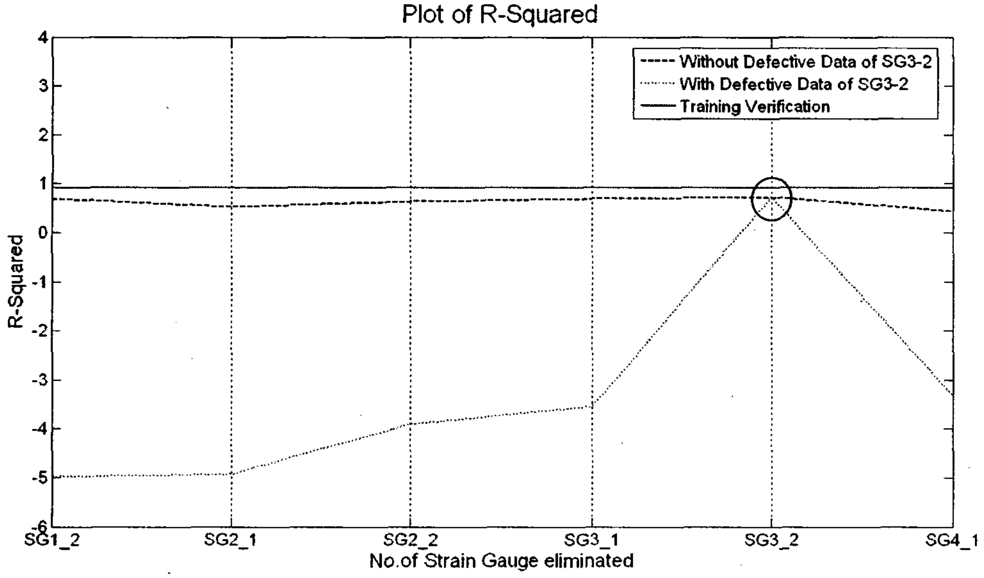

Figure-6.16: Plot of R-squared for inputs of non defective data of SG3_2, with defective data of SG3_2 and training inputs (Case-2)

Table-6.4: Table of R-squared calculated from plots of network output and actual data for defective sensor Case-3

Sl. No.	Eliminated sensors	R-squared for training inputs	R-squared (In absence of defective sensor SG3_2)	R-squared (In presence of defective sensor SG3_2)
1	SG1_2	0.914	0.690	-0.444
2	SG2_1	0.911	0.522	-0.889
3	SG2_2	0.919	0.631	-0.373
4	SG3_1	0.929	0.691	-0.011
5	***SG3_2***	***0.922***	***0.739***	***0.739***
6	SG4_1	0.929	0.445	-0.777

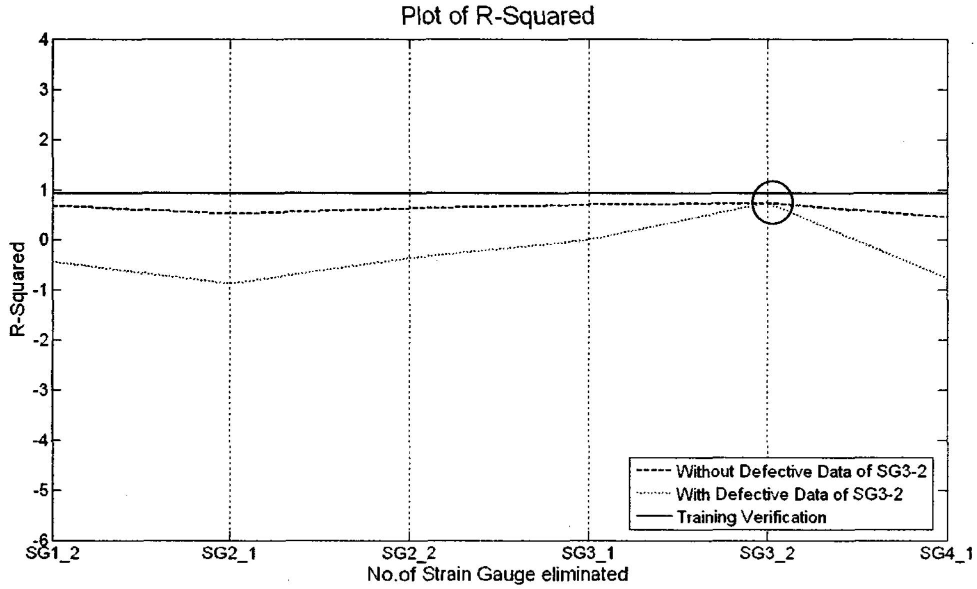

Figure-6.17: Plot of R-squared for inputs of non defective data of SG3_2, defective data of SG3_2 and training inputs (Case-3)

Discussion

- It has been observed from section 6.2 that the removal of defective sensor caused R-squared value to be higher in comparison to the cases when the defective sensor is present in the network

- Also in section 6.3, six sensors data have been chosen of which one sensor data (SG3_2) made defective considering three cases of defectiveness. Then we plotted R-squared for each cases and maximum R-squared value was found at the location of SG3_2
- The sequential search algorithm developed here has been tested with three different cases of defective data pattern and they could be detected accurately
- Finally we have applied these methodologies to detect the defective sensor from the SHM system of a Canadian Bridge. Application of these methodologies has been elaborately discussed in the following sections.

6.4 Sequential search and Binary Search

6.4.1 Description:

In the previous analysis, we made manually one sensor defective and demonstrated the sequential searching method to detect the defective sensor. But in reality we don't know which sensor is defective and in that case we need to analyze the data pattern of all sensors by eliminating each before presenting to the trained network. But this can make the whole process tedious if the number of sensors is high. Taking this issue into our consideration we considered binary search in association with sequential search method. These two methods of searching have been proven to be more promising for finding out the defective sensor from a series of sensors installed on a structure or structural component in SHM. The binary search has substantially reduced the time in comparison to the time required for sequential search only. For analyzing the effectiveness of these two methods we first harvested data from eight bidirectional strain gauges (16 strain data for horizontal and vertical direction) and one temperature gauge data from the SHM database of the Portage Creek Bridge. Then we divided these data into training and simulation data. For training purpose we took one minute interval data for the periods of December/05, January/06 and February/06 as training input and converted these strain data into average hourly strain (micro-strain/hr). On the other hand, we took the one minute interval data for the period of first 15 days of March/06 for simulation and also converted these strain data into average hourly strain (micro-strain/hr). Then we sorted out defective sensors by applying firstly, the sequential search method and secondly, the binary search method with the help of artificial neural network (ANN).

It should be noted here that in this section of the analysis we worked only with first 15 days data of March, 2006. As there was no evidence of abrupt changes in the data pattern at this portion (Figure-6.6b), we considered it as non defective portion of the data. To simulate the defective sensor data at this portion, we manually made the 10th sensor (SG6_1) defective by keeping 20% data unchanged and 80% data changed to zero. To validate the effectiveness of binary and sequential search method, this manually made defective sensor (SG6_1) has been detected by applying both methods as follows:

6.4.2 Sequential Search:

This method has been discussed in subsection 4.4.1. In sequential search procedure we sequentially took out one sensor each time and present the remaining to the respective trained network as input. The network has to be trained previously by sequentially taking out one sensor each time from the selected group of sensors. Then we plot the output and the present target in an excel sheet and calculate the R-squared value (Table-6.5). These

R-squared values were plotted against the number of sensor eliminated. It should be noted that for easier comparison we calculated R-squared for every set of inputs i.e. training data set and data set with defective sensor. Generally from plot of R-squared for training input and defective data set, the maximum value of R^2 for the defective data set can be found and the corresponding sensor at this point would be the defective sensor (Figure-6.18).

In this analysis 16 networks have been trained and plotted the R-squared for 16 new inputs to these trained networks. The R-squared are shown in Table-6.5 and the plot is shown in Figure-6.24. For both the cases we have observed that the R-squared value become maximum at the point when the defective sensor (SG6_1) is eliminated.

Table-6.5: Table of R-squared calculated from plots of network output and actual data for 1st fifteen days data of March, 2006 (16 sensors considered for sequential elimination)

Sl. No	Sensors Eliminated	R-squared for training inputs	R-squared for 1^{st} 15 days data (defective SG6_1) March, 2006
1	SG1_2	0.950	-3.037
2	SG2_1	0.930	-8.793
3	SG2_2	0.922	-2.088
4	SG3_1	0.934	-2.782
5	SG3_2	0.935	-2.030
6	SG4_1	0.946	0.150
7	SG4_2	0.936	-1.323
8	SG5_1	0.941	0.449
9	SG5_2	0.950	-5.270
10	***SG6_1***	***0.952***	***0.802***
11	SG6_2	0.956	-1.439
12	SG7_1	0.916	0.657
13	SG7_2	0.949	-0.180
14	SG8_1	0.925	-0.634
15	SG8_2	0.948	-1.573
16	SG9_1	0.949	-0.814

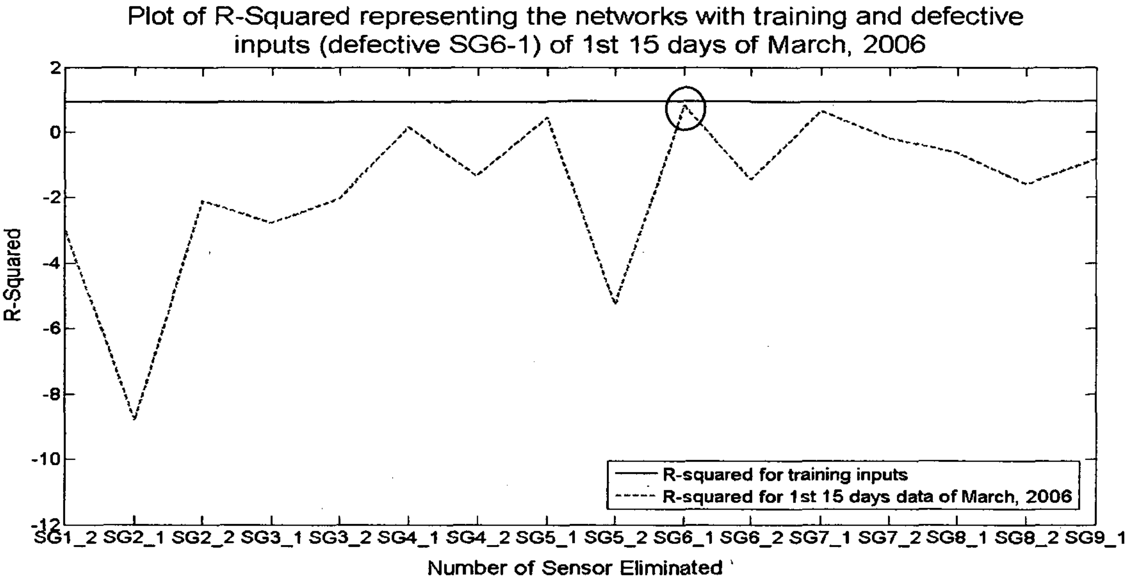

Figure-6.18: Plot of R-squared by sequentially taking out 16 sensors from first 15 days data of March, 2006. In this case the maximum $R^2 = 0.802$ is at the location of SG6_1.

6.4.3 Binary Search:

This method has been explained earlier in subsection 4.4.2. The binary search helps in the beginning by confirming the presence of defective sensor in a smaller group by reducing the size of the suspected group from a larger group. It also reduces the time in comparison to using the sequential search alone in a large group of sensors. In this search method we subdivided the sixteen strain data into two groups with eight strain data in each group namely G-8-1 and G-8-2 as shown in Figure-6.19. Then we trained two networks (Net-1 & Net-2) taking inputs from each data group and taking same target (SG1_1) for both networks. In training of both networks we considered maximum periods of data (December/05, January/06 & February/06) as input so that the network can store a maximum range of data pattern in its memory. Then Net-1 was tested with 15 days data of March/06 from group G-8-1. For this new input, the network output and the present target data plotted in Figure-6.20. After this Net-2 was tested with 15 days non-defective data and secondly with defective data of March/06 from sensor group G-8-2. For these two inputs, the two network outputs and the present target data are plotted in Figure-6.21.

From comparison of Figure-6.20 and Figure-6.21, it can be noticed that the difference in data pattern in Figure-6.21 is higher than that of Figure-6.20. Also by comparing the R-squared value it has been found that R-squared value abruptly changes with the changes in the values in data for defective sensor. This irregular data pattern and R-squared value of Figure-6.21 primarily indicates the presence of defective sensor is in the sensor group G-8-2.

Now following the same procedure we have subdivided the sensors of group G-8-2 further into two groups of four sensors namely G-4-1 and G-4-2. Then we have trained network-3 (Net-3) and network-4 (Net-4). For training of Net-3 we considered the data from twelve sensors (G-8-1 + G-4-1) as input. Also for training of Net-4 we considered the data from twelve sensors (G-8-1 + G-4-2) as input. Here we added first eight sensors group (G-8-1) with both the group to increase the number of inputs for both networks to develop better input-output correlation-ship. In addition to this we considered the previous target (SG1_1) for both the network. Then we tested the Net-3 firstly with 15 days non-defective data of March/06 from first group of twelve sensors. For this new input the network output and the present target data plotted in Figure-6.22. Secondly, we tested the same network with 15 days data of March/06 from first group of twelve sensors which contains the manually made defective sensor (SG6_1). For this test the network output and the present target data are plotted in the same Figure-6.22. In the next step we tested Net-4 with 15 days data of March/06 from second twelve sensors group. The network output and the present target data are plotted in Figure-6.23.

From comparison of Figure-6.22 and Figure-6.23 we observed that the difference in the data pattern of Figure-6.22 is higher than that of Figure-6.23. In addition to this, by comparing the R-squared value we noticed the abrupt changes occurred in R-squared when the Net-3 was presented with the defective sensor data. Also from these observations we could certainly say that the defective sensor is present in the 1st four sensors group (G-4-1). After being confirmed that the defective sensor was present in the 1st group of four sensors, we followed the sequential search procedure to identify the specific defective sensor from the 1st four sensors group. The calculated R-squared are shown in the Table-6.6 and their plots are shown in Figure-6.24

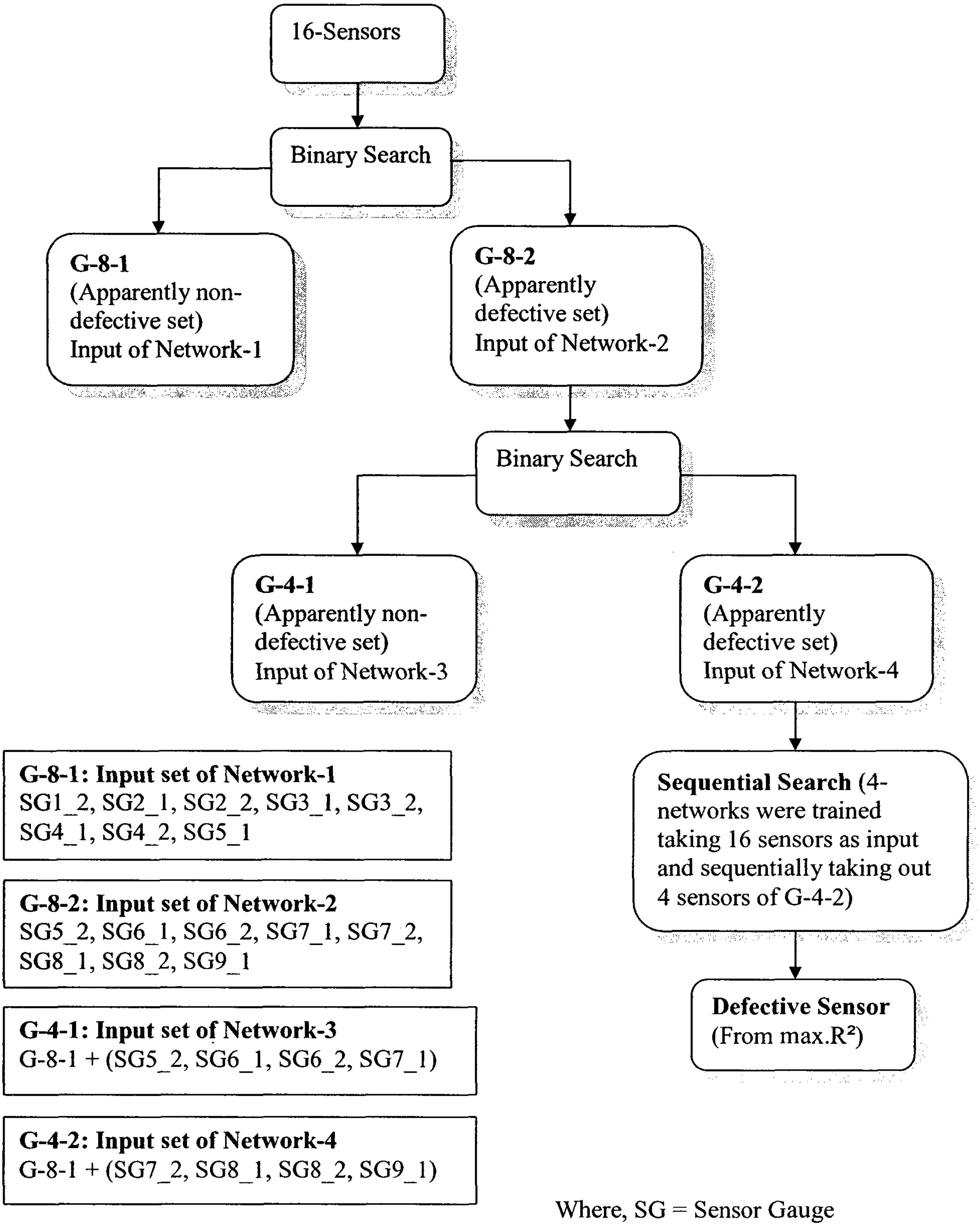
16-Sensors
Binary Search
G-8-1
(Apparently non-defective set)
Input of Network-1
G-8-2
(Apparently defective set)
Input of Network-2
Binary Search
G-4-1
(Apparently non-defective set)
Input of Network-3
G-4-2
(Apparently defective set)
Input of Network-4
Sequential Search (4-networks were trained taking 16 sensors as input and sequentially taking out 4 sensors of G-4-2)
Defective Sensor
(From max.R²)
G-8-1: Input set of Network-1
SG1_2, SG2_1, SG2_2, SG3_1, SG3_2, SG4_1, SG4_2, SG5_1
G-8-2: Input set of Network-2
SG5_2, SG6_1, SG6_2, SG7_1, SG7_2, SG8_1, SG8_2, SG9_1
G-4-1: Input set of Network-3
G-8-1 + (SG5_2, SG6_1, SG6_2, SG7_1)
G-4-2: Input set of Network-4
G-8-1 + (SG7_2, SG8_1, SG8_2, SG9_1)
Where, SG = Sensor Gauge

Figure-6.19: Schematic diagram of binary and sequential search method of defective sensor

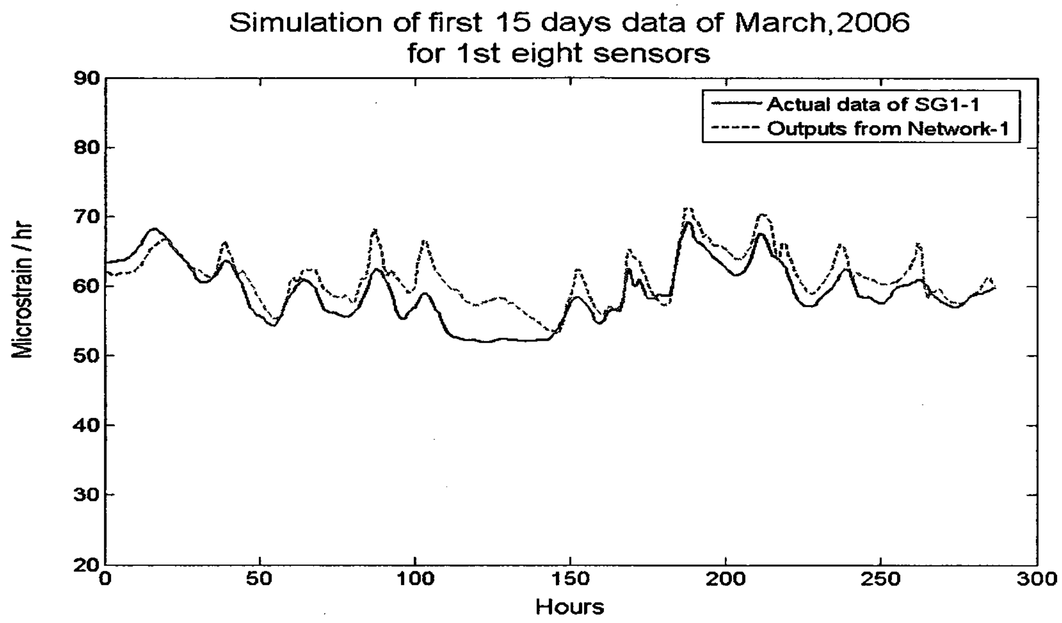

Figure-6.20: Simulation of Network-1 with 15 days data of March, 2006 from sensor group G-8-1. R-squared = 0.495

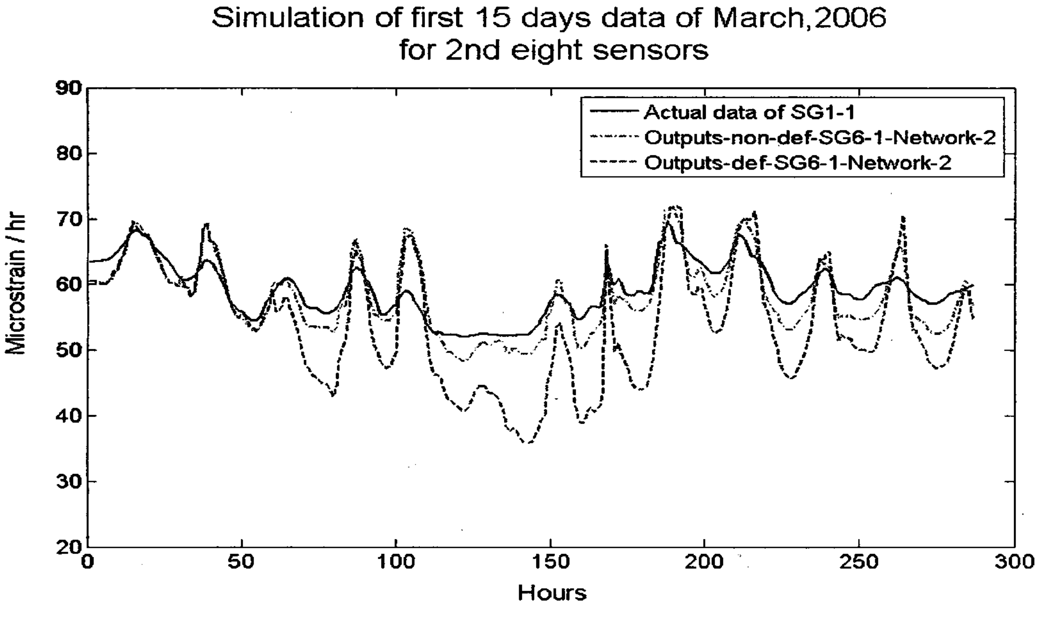

Figure-6.21: Simulation of Network-2 with 15 days non-defective and defective data of March, 2006 from group G-8-2. R-squared for non-defective sensor = 0.497 & R-squared for defective sensor data= -2.565

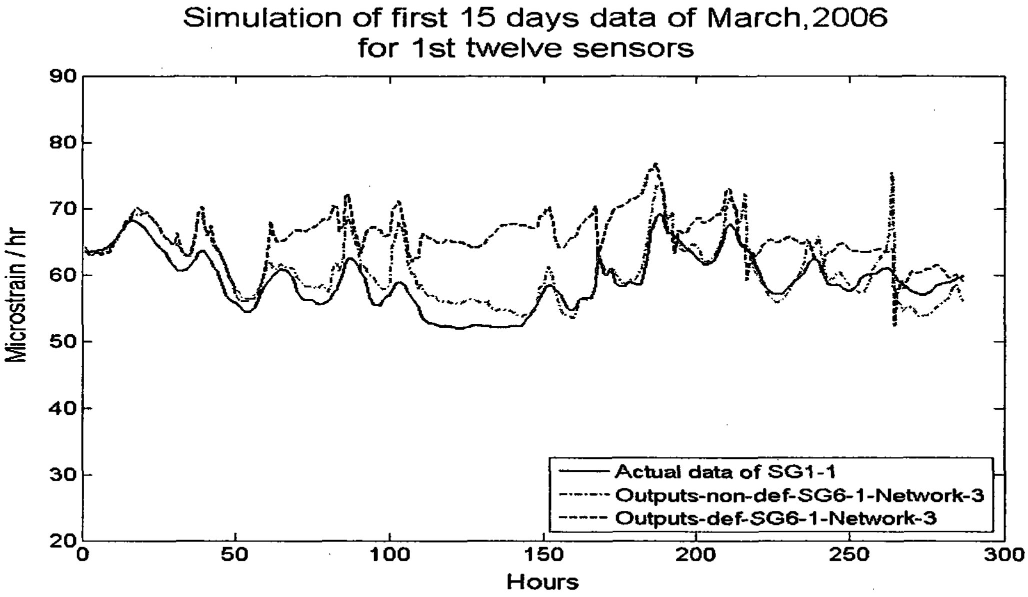

Figure-6.22: Simulation of Network-3 with 15 days non-defective and defective data of first twelve sensors (G-8-1 + G-4-1) of March, 2006. R-squared for non-defective sensor data = 0.546 & R-squared for defective sensor data = -2.567

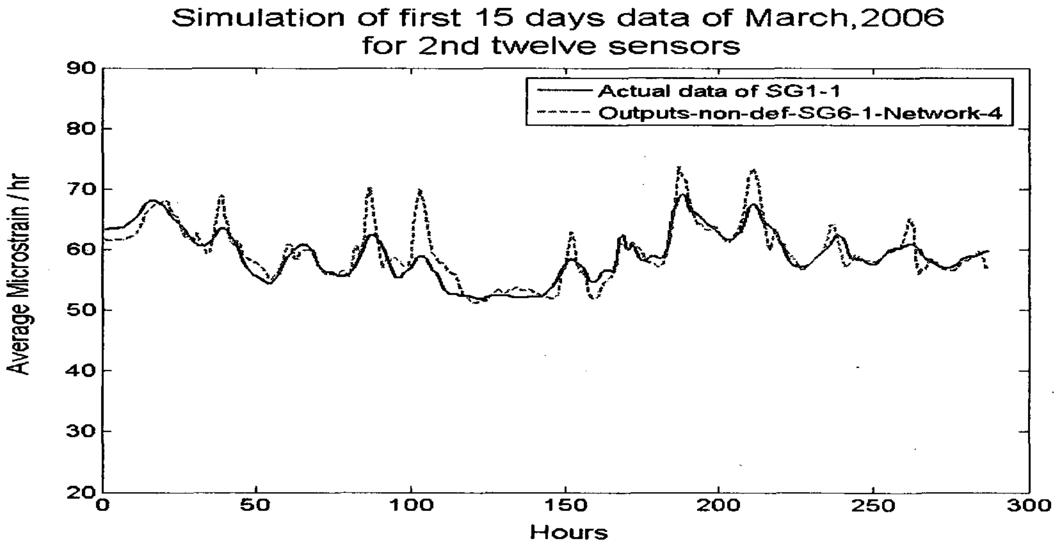

Figure-6.23: Simulation of Network-4 with 15 days data of second twelve sensors (G-8-1 + G-4-2) of March, 2006. R-squared for this plot = 0.724

Table-6.6: Table of R-squared calculated from plots of network output and actual data for 1st fifteen days data of March, 2006 (4 sensors considered for sequential elimination)

Sl. No.	Number of Sensors Eliminated	R-squared for training inputs	R-squared with defective sensor (SG6_1) data
1	SG5 2	0.732	-5.270
2	***SG6_1***	***0.844***	***0.802***
3	SG6_2	0.267	-1.439
4	SG7_1	0.827	0.657

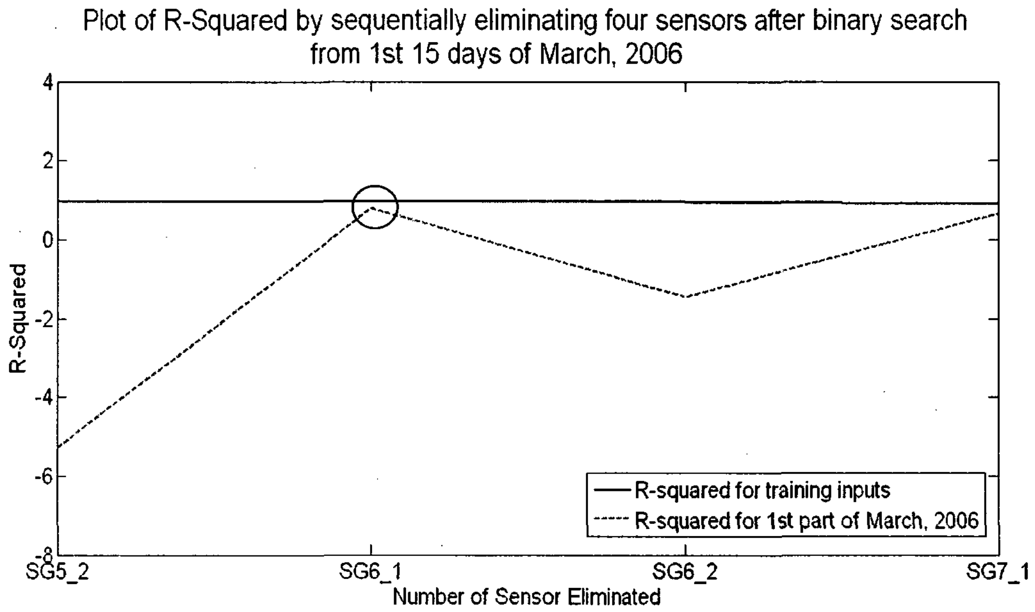

Figure-6.24: Plot of R-squared by sequentially taking out four sensors from G-4-1. The maximum $R^2 = 0.802$ is at the location of SG6_1

6.5 Identifying defective sensor from the second half of March, 2006

6.5.1 Introduction

In the previous analysis we observed that the binary search procedure has shown satisfactory response in finding out the defective sensor from a series of sensors in a structure. Now we are going to apply these methodologies for detecting the reason of spike in the data pattern of 2nd part of March 2006 (Figure-6.6a). From Figure-6.6b, we observed that there was no spike in the 1st fifteen days data and the spike was present in the 2nd fifteen days data. Now we are going to find which sensor data causes this spike in the 2nd fifteen days data of March, 2006. It should be noted that in this analysis the previously trained networks have been used and the same procedure has been applied as in section 6.4.

6.5.2 Binary Search

In this method the new inputs of the networks from the 2nd part of March/06 are divided into two groups. The first part consists of first group (G-8-1) of 8 sensors data and the second part consists of second group (G-8-2) of 8 sensors data. The data from these two sensors group are presented to the previously trained networks (Network-1 & Network-2) and the outputs are plotted in Figure-6.25 and Figure-6.26. From the changes in data pattern of these figures we could easily identify the shift in the plot (Figure-6.26) of second 8 sensors data. From this observation in Figure-6.26, we could preliminarily confirm the presence of defective sensor in the second 8 sensors group (G-8-2). Now sensor group G-8-2 has been further subdivided into two groups (G-4-1 and G-4-2) of 4 sensors in each group. Then these groups have been presented simultaneously to the previously trained networks (Network-3 and Network-4). These networks outputs are plotted in the Figure-6.27 and Figure-6.28. By comparing these two figures and their R-squared values group G-4-2 has been selected as apparently defective sensor group. Finally the defective sensor SG8_1 has been detected by applying the sequential search method in group G-4-2. The calculated R-squared are shown in Table-6.7 and their plots are shown in Figure-6.29. For verification of this result, the sequential search method has been applied again as follows:

6.5.3 Sequential Search

In the next step we sequentially took out one sensor at a time from 16 sensors of the 2nd part of March/06 and for each of this elimination the inputs were presented in the respective network that was trained by sequentially taking out similar sensor from 16 sensors data of December/05, January/06 and February/2006. For presentation of 16 inputs, the network produced 16 outputs and each output vs. actual target data has been plotted in a separate excel sheet. For each of these plots calculated R-squared values have been recorded in the Table-6.8. From this table the R-squared values were plotted in Figure 6.30. From this table and plot of the R-squared values we can observe that the maximum R-squared value found at the point where the sensor SG8_1 was eliminated. Also we could asses it from the observation of the best fit plots out of 16 plots (network output vs. actual target) from the separate excel sheet. In this case the best fit plot is sorted from separate excel sheet as shown in Figure-6.31. Following the similar procedure as in section 6.4, we could certainly say that the sensor data of SG8_1 is responsible for the spike or hump in the second 15 days data pattern. In addition to this, the data of this sensor might be erroneous or the sensor might be defective or this could be the warning of the structural changes at this location which requires further investigation.

From further observation of Table-6.8, it can be noticed that the elimination of sensor SG2_2 and SG3_1 have also shown better R-squared value. The plots of network output and actual target for eliminating these two sensors are shown in Figures-6.32 and 6.33. The combined effects of these sensors have been checked by eliminating and presenting them to the trained network with three probable groups or combinations and the calculated R-squared plotted as shown in Figure 6.34, 6.35 & 6.36. Now by comparison of these plots with the plot in Figure 6.31, it can be seen that still the elimination of sensor SG8_1 alone has shown better correlation-ship and maximum R-squared value. So, at the primary level the maximum impact has come from sensor SG8_1 which is mainly responsible to create the hump or spike in the 2nd 15 days data of March, 2006. Also, there might be more or less participation of other sensors due to structural changes which need further investigation.

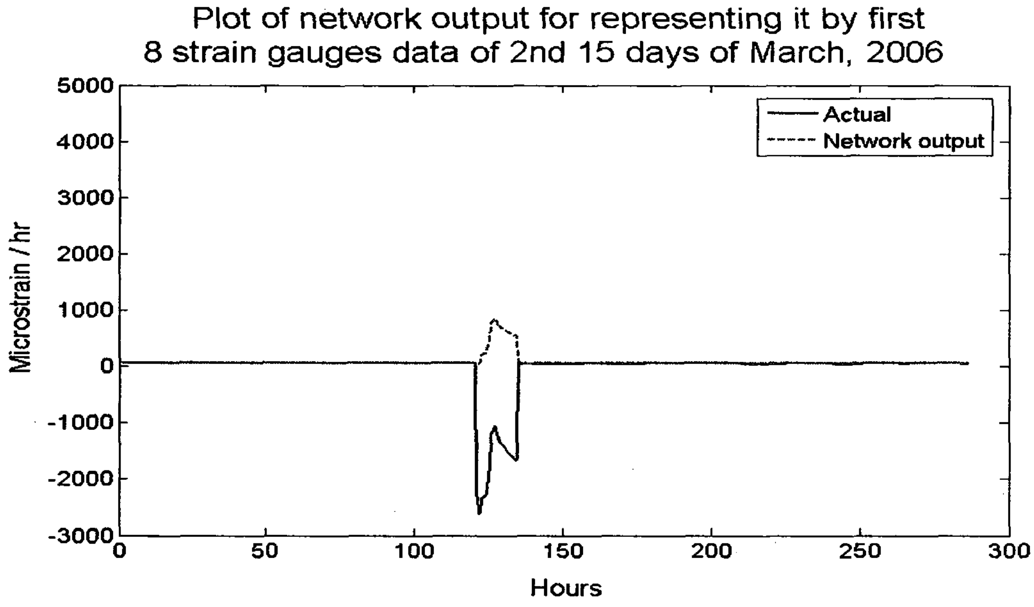

Figure-6.25: Plot of network output by representing it with the new inputs from 1st eight sensors (G-8-1) of second part of March, 2006. $R^2 = - 0.561$

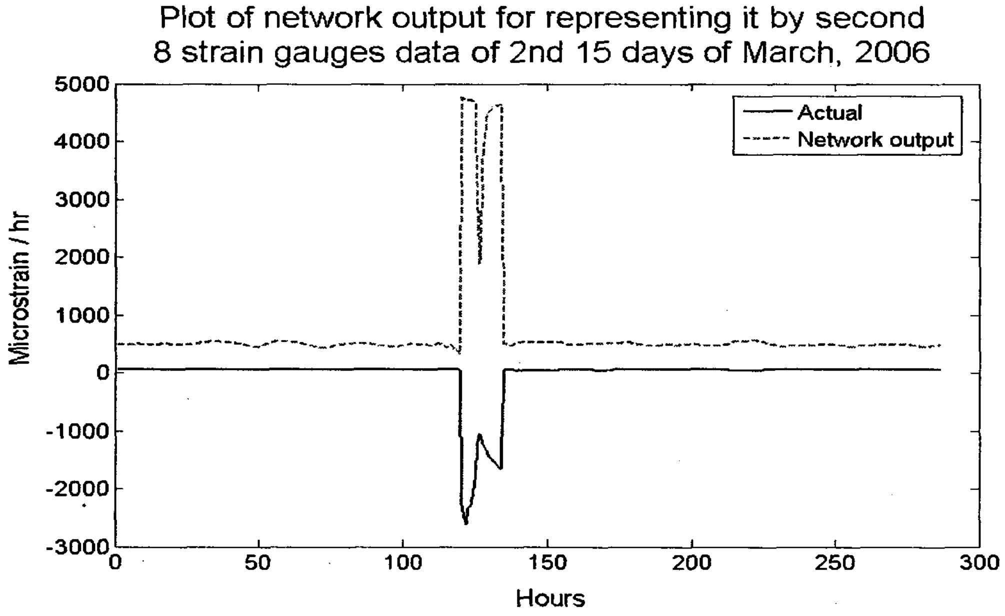

Figure-6.26: Plot of network output by representing it with the new inputs from 2nd eight sensors (G-8-2) of second part of March, 2006. $R^2 = -11.780$

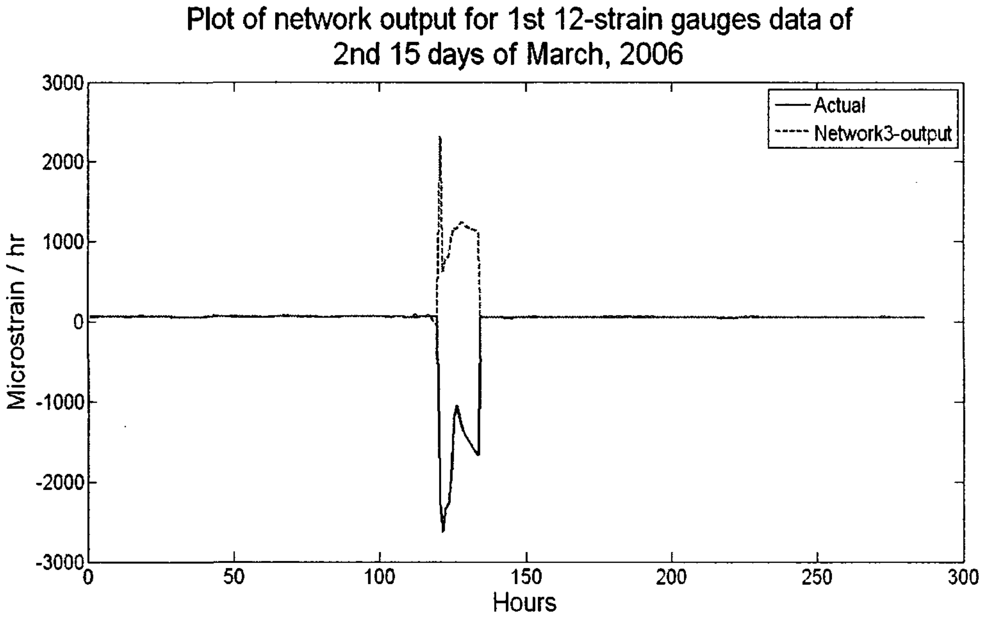

Figure-6.27: Plot of network output by presenting it with the new inputs from G-4-1 of four sensors of second part of March, 2006. R^2 = - 1.684

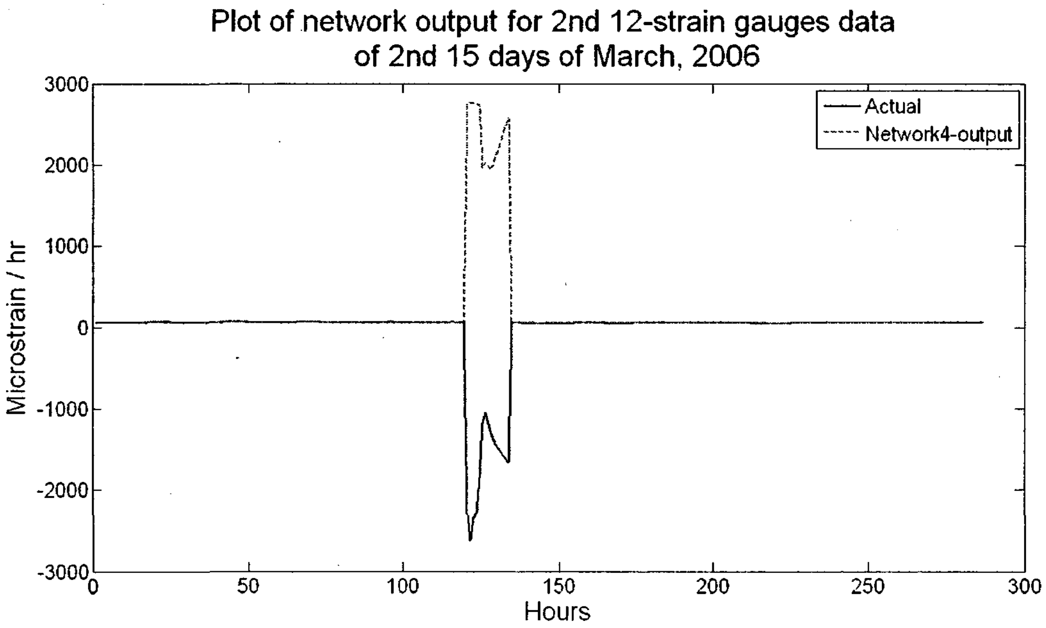

Figure-6.28: Plot of network output by representing it with the new inputs from G-4-2 of four sensors of second part of March, 2006. R^2 = - 4.456

Table-6.7: Table of R-squared calculated from plots of network output and actual data for 1st fifteen days data of March, 2006 (4 sensors considered for sequential elimination)

Sl. No.	Number of Sensors Eliminated	R-squared for training inputs	R-squared with defective sensor (SG6_1) data
1	SG7 2	0.949	-10.460
2	***SG8_1***	***0.925***	***0.855***
3	SG8_2	0.948	-43.949
4	SG9_1	0.949	-3.978

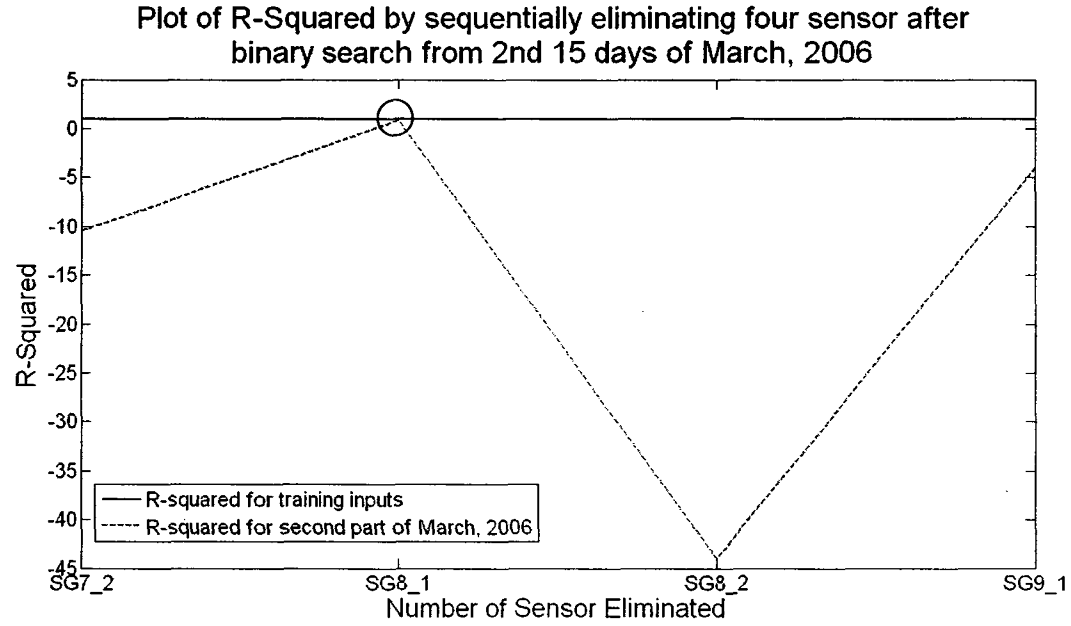

Figure-6.29: Plot of R-squared by sequentially taking out four sensors from G-4-1. The maximum R² = 0.802 is at the location of SG6_1

Table-6.8: Table of R-squared calculated from plots of network output and actual data for 2nd fifteen days data of March, 2006 (16 sensors considered for sequential elimination)

Sl. No.	Sensors Eliminated	R-squared for training inputs	R-squared for new inputs of 2nd 15 days data of March, 2006
1	SG1_2	0.950	-0.403
2	SG2_1	0.930	-1.291
3	SG2_2	0.922	0.683
4	SG3_1	0.934	0.140
5	SG3_2	0.935	-77.126
6	SG4_1	0.946	-0.960
7	SG4_2	0.936	-1.244
8	SG5_1	0.941	-2.008
9	SG5_2	0.950	-1.984
10	SG6_1	0.952	-43.344
11	SG6_2	0.956	-8.252
12	SG7_1	0.916	-2.718
13	SG7_2	0.949	-10.460
14	***SG8_1***	***0.925***	***0.855***
15	SG8_2	0.948	-43.949
16	SG9_1	0.949	-3.978

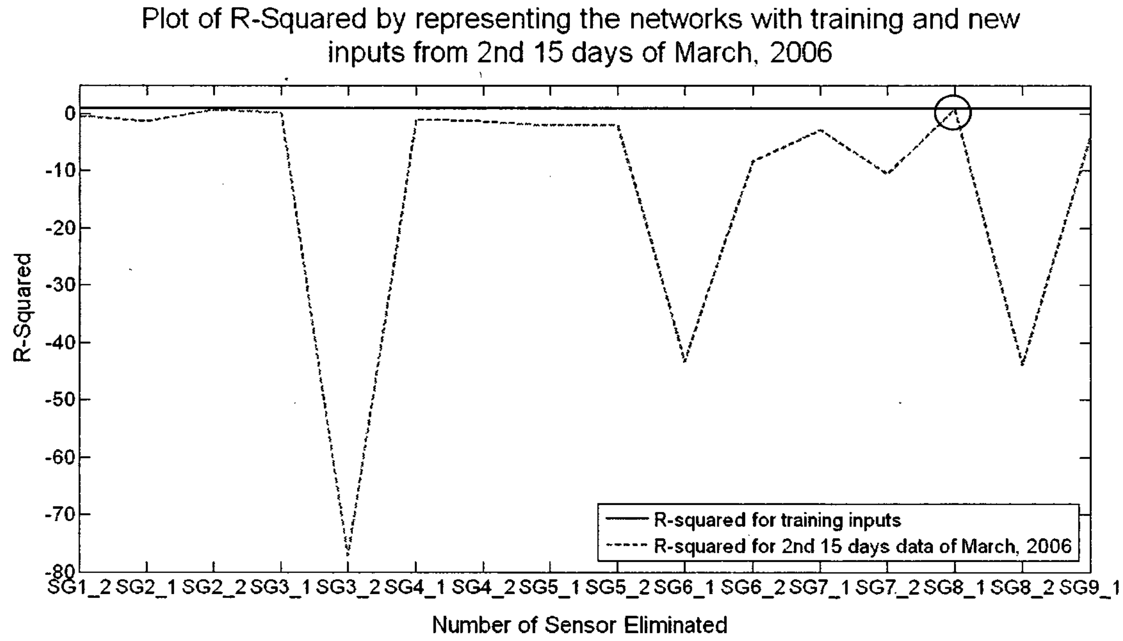

Figure-6.30: Plot of R-squared by representing the networks sequentially taking out 16 sensors from 2nd fifteen days data of March, 2006.

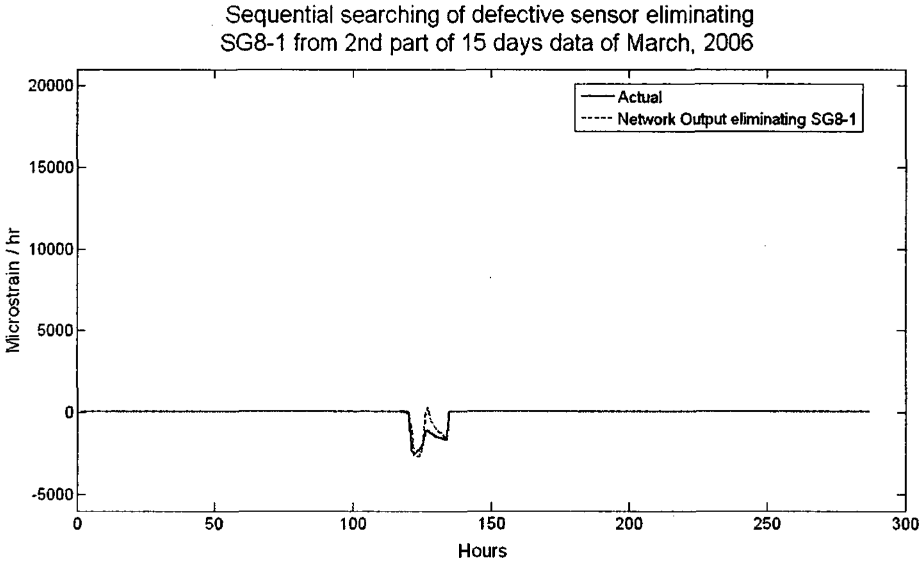

Figure-6.31: Apparently best fit plot of network output and actual target eliminating SG8_1 from input

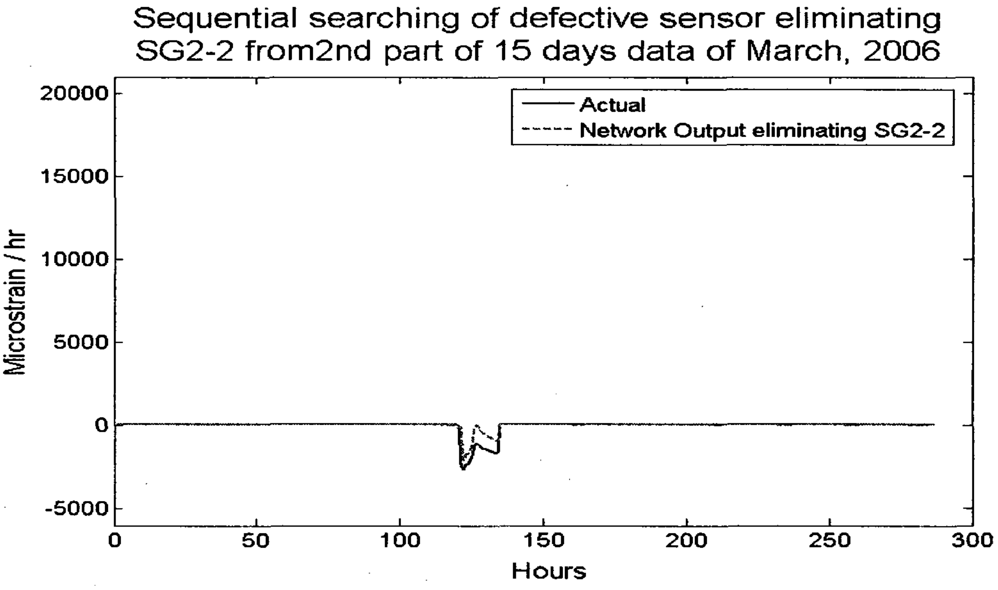

Figure-6.32: Apparently better fit of network output and actual target eliminating SG2_2 from input

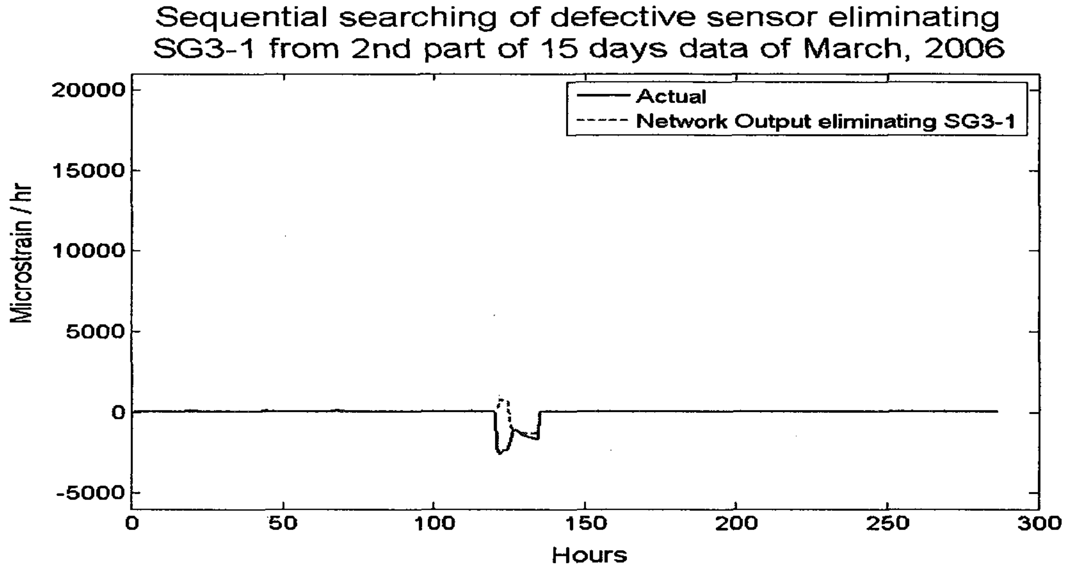

Figure-6.33: Apparently far better fit plot of network output and actual eliminating SG3_1 from input

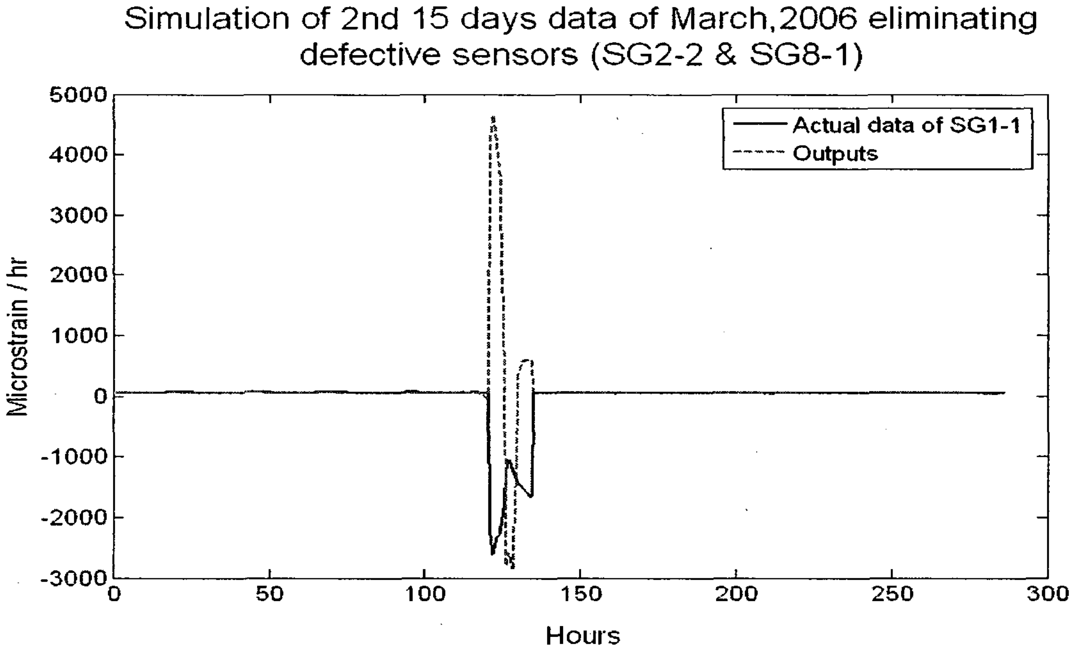

Figure-6.34: Plot of sensor output after elimination of SG2_2 & SG8_1

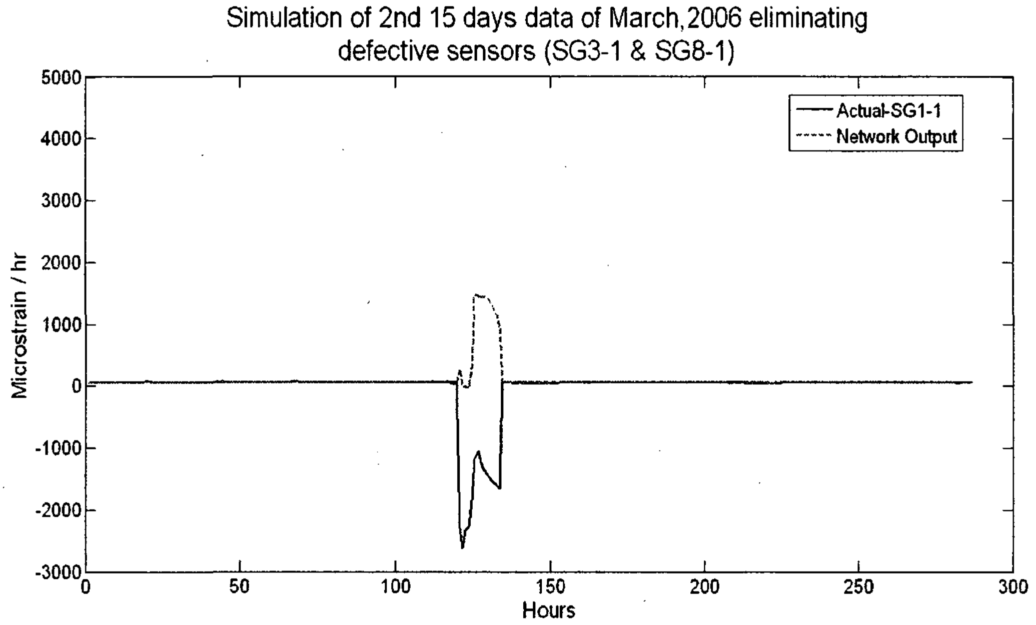

Figure-6.35: Plot of sensor output after elimination of SG3_1 & SG8_1

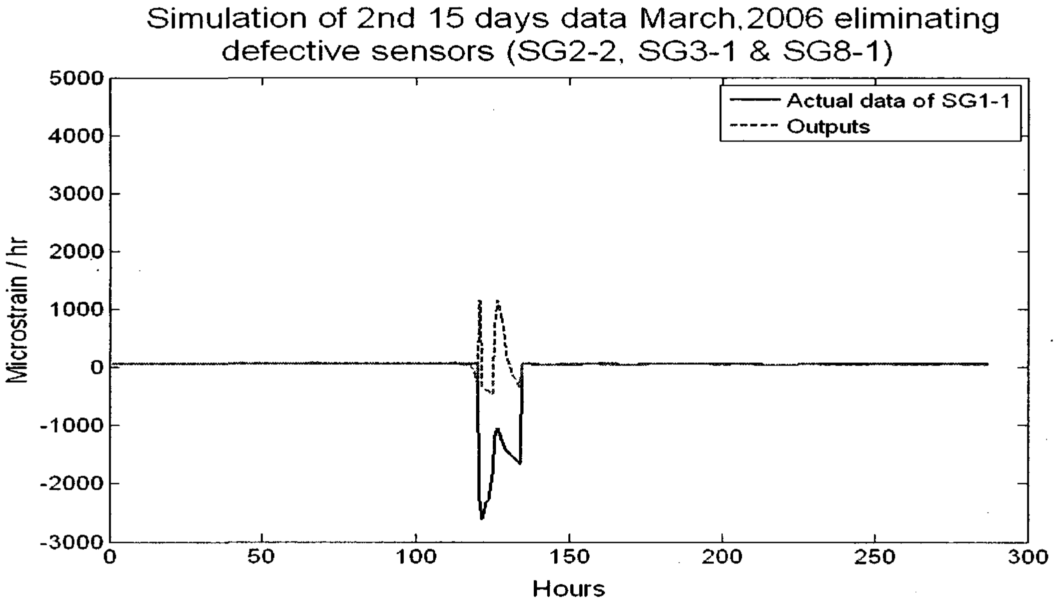

Figure-6.36: Plot of sensor output after elimination of SG2_2 & SG3_1, SG8_1

6.6 Data block sampling

The method of data block sampling has been shortly introduced in Chapter-4. A brief description on its application is explained here by selecting one strain gauge data of the Portage Creek Bridge:

The SHM data of the Portage Creek Bridge is collected through high speed internet line to an interactive web page at ISIS Canada's web site. The earliest and the last time of data available for downloading from the database are 2003-04-23 13:54:33 and 2006-08-26 17:58:57 respectively. However there are some times where data are not available from the source. These off-times have occurred at all the range of the monitoring. The duration of off times varies from a few hours to several months. 30 strains of 16 2d-strain gauges, 6 accelerometer readings of 2 3d-accelometers and 1 temperature data available for any starting time falling in the range limited by the time mentioned above. Sampling rates available are 1/32s, 1s, 10s and 1m. Number of data points available in a single download varies from 32 to 30,000. Data can be obtained both graphically and numerically.

In the beginning data was downloaded graphically over various sample rates and points at random starting times to see the overall nature and trend of the data. For example below (Figure-6.37) is the data plotted for S_4_1_C2 (strain gauge reading 4 along strain direction 1 of column 2) taken at 1 reading per second of 30000 points starting at time 22:31:07 on 2005-05-14. This data time duration is 8 hours 20 minutes. Careful inspection shows that there are some slopes with continuous oscillation about x –axis.

There are also some random vertical straight lines. It seems that those random lines were resulted from some sudden impacts. Two of them at right are long enough to be distinctive. Later analysis revealed that slopes are mostly the effect of temperature changes and those random vertical lines are the results of moving loads, most possibly heavy vehicles.

If we gradually zoom in by narrowing the data range and pick the starting time very close to the rightmost vertical line, we shall get clearer picture. By gradual inspection of the location of the line and changing starting point accordingly we finally arrive at a start time of 06:23:58 on 2005-05-15, we get the graph shown in Fig-6.38. The rightmost and longest vertical straight line in Fig-6.37 now looks like a spike. Sample rate of the signal block is 1 s and length is 128 points. The time duration of this block is 128 seconds. If we increase the rate to 32Hz, data plotted graph is shown in Fig 6.39. The time duration of this block is 8 seconds. It shows clearly that the change of strain over a short period of time is the result a gradually (not an impact as appeared in Fig 6.37) changing load. This is the characteristic of a load applied on a pier by a vehicle moving on a bridge.

Based on the observations similar to the one mentioned above, it is apparent that in order to study the structural behavior of the bridge we need to analyze it under two conditions, 1) steady state where only small oscillations is observed (Figure-6.40) 2) the agitated condition as shown in the Fig-6.39 in addition to small oscillations.

For verification of strain, the data or in other words signal blocks are collected for 2 types of analysis,1) Steady state strain 2) Live load strain. The brief description on how data can be collected for both cases is given below:

1) Steady state strain: Fig-6.40 shows typical steady state condition plotting. It represents the condition when there is no live load such as a vehicle on the bridge. Steady state strain at direction 1 (vertical) of strain gauge 4 (SG4_1-C2) of column 2 was taken for the study of this case. The sampling rate is 32 Hz and length is 256 points. Time duration of each or each block is taken 8 seconds as passing a vehicle is never found taking more than this time.

2) Live load strain: Fig-6.39 is typical live loaded condition time series of strain readings. By inspection as exemplified in this section and shown in Figures- 6.37, 6.38 & 6.39 data are collected only conditions where peak values are more than steady state conditions in S_4_1_C2.

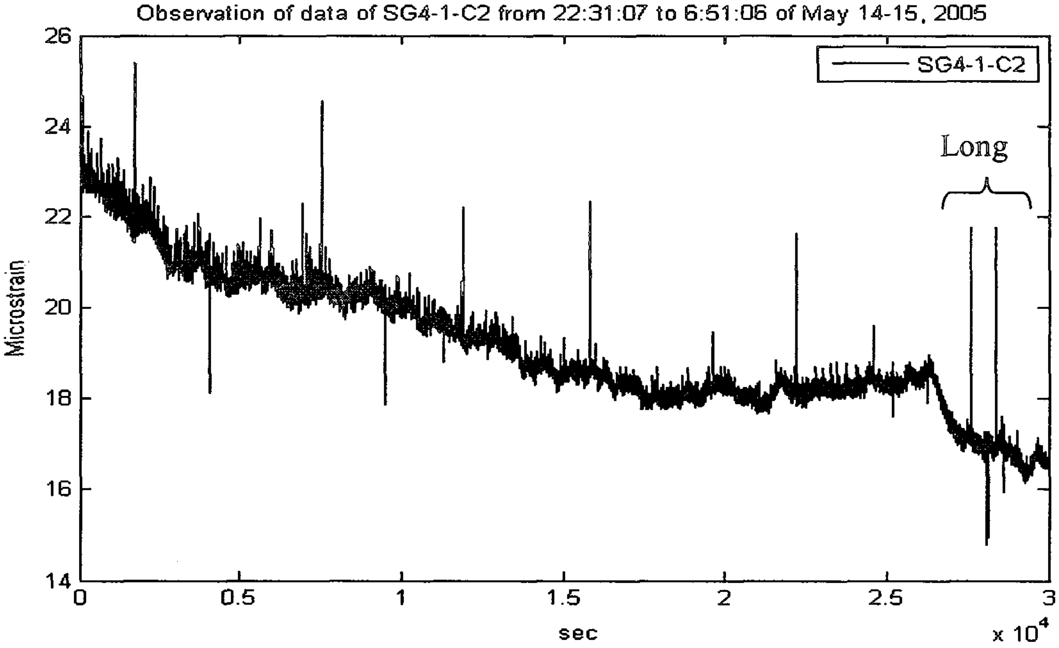

Figure-6.37: Plot of 30000 data (1 data per sec) from SG4_1_C2 from 22:31:07 to 6:51:06 of May 14-15, 2005

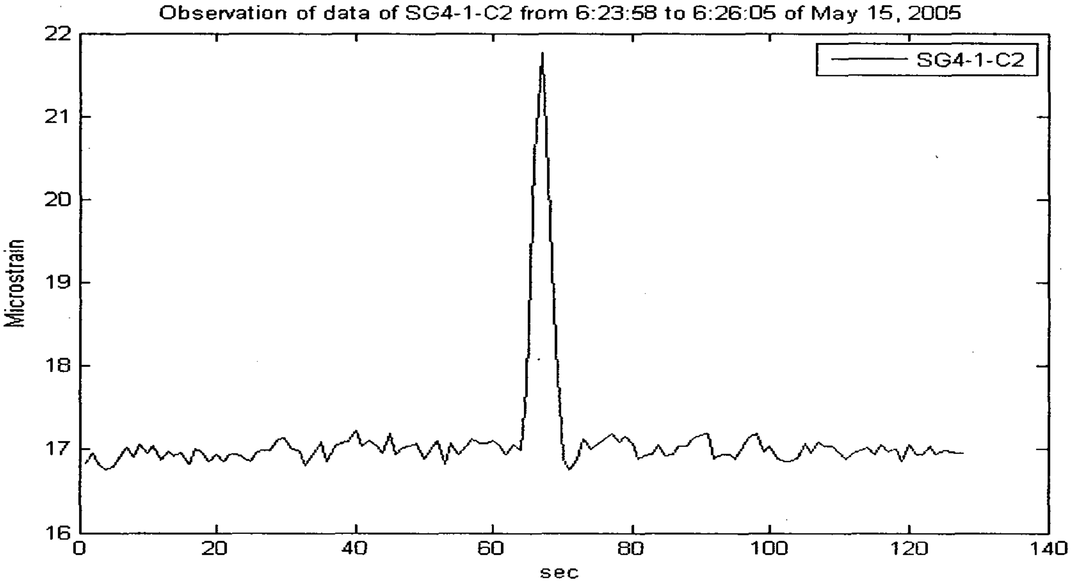

Figure-6.38: Plot of 128 data (1 data per sec) close to one highest peak at right of Figure from SG4_1_C2 from 6:23:58 to 6:26:05 of May 15, 2005

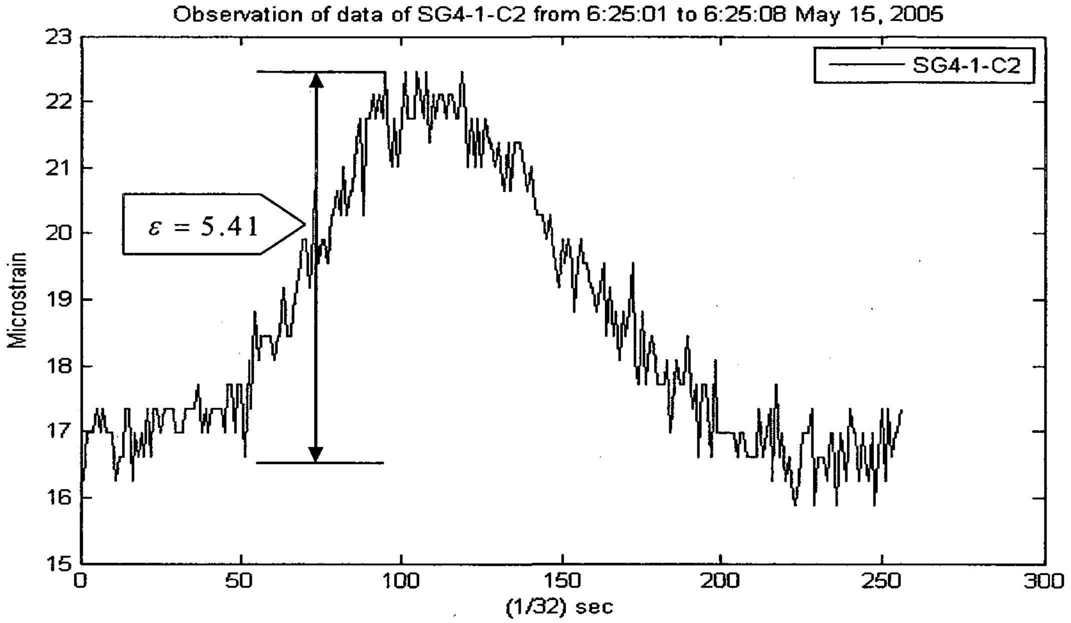

Figure-6.39: Plot of 256 data (32 data per sec = 8 sec) at the highest peak from 6:25:01 to 6:25:08 of May 15, 2005

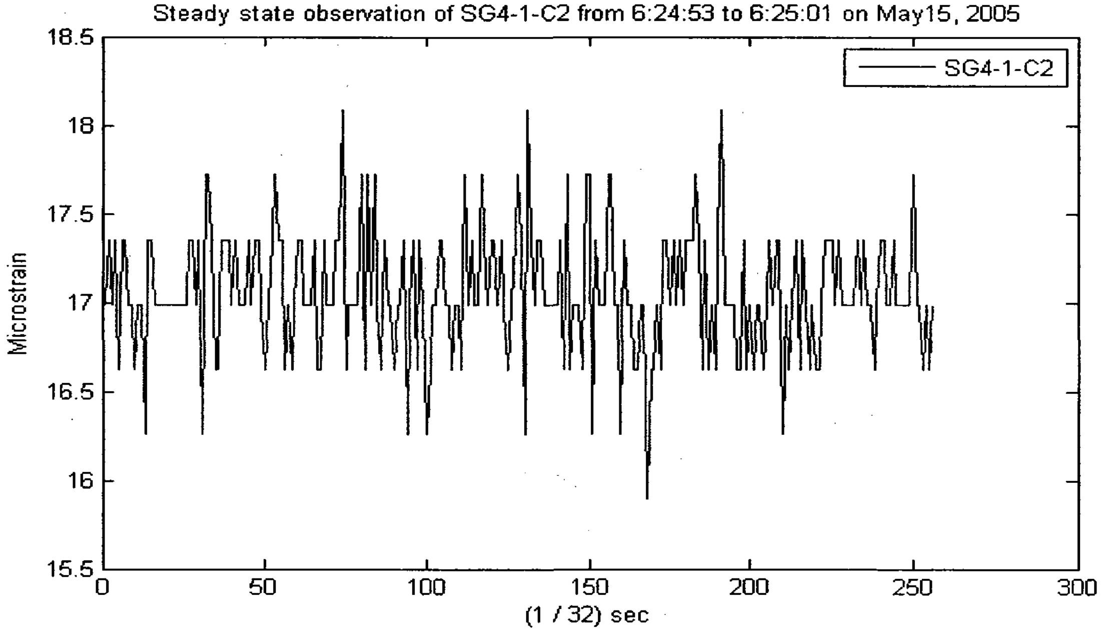

Figure-6.40: Plot of steady state block of 256 data points (32 data per sec = 8 sec) just before the highest peak from 6:24:53 to 6:25:01 of May 15, 2005

6.6.1 Strain calculation for static load

From the Journal by Huffman et al., (2006), it has been observed that a dynamic and static load test of the bridge was conducted in May, 2005 using 26 Ton truck that was positioned on the instrumented column of the Bridge yields similar strain patterns as discussed above. The vertical strain on a column has been calculated here considering the static load imposed by the 26 Ton truck:

We know strain, $\varepsilon = P/_{AE}$ -- (11)

Here diameter of column C2 =1676 mm, Area, A = 2206165 mm^2

Assume $f'_c = 30$ MPa, E = $4500\sqrt{f'_c}$ = 24647.52 MPa, P = 26 T =255060 N,

Hence, strain from the formula 10, $\varepsilon =$ 4.69 microstrain

From Figure-6.40, by averaging 256 data values we get = 17.06 microstrain.

From Figure-6.39, we took the maximum value of data peak and we get = 22.47 microstrain.

Now taking the difference of the average value of the reference block and the maximum value of peak from test block, we get = 22.47-17.06 = 5.41 microstrain which is almost close to the strain due to the static load = 4.69 microstrain. The differences can be attributed to the dynamic amplification effect. The dynamic amplification factor in this case works out to be 1.15, which is compatible with the Canadian Highway Bridge Design Codes (CAN/CSA-S6-06).

CHAPTER-7

Summary and Conclusions

7.1 Summary

Artificial Neural Network (ANN) as a pattern identifier is quite popular in the field of signal analysis. The working principle of ANN is derived from the function of human brain. An ANN is composed of an input layer, output layer and one or more hidden layer of computing nodes. They are trained using a known set of input and output (target) data and this trained network can be used to track changes of data pattern in future. Wavelet transform is another powerful signal analysis tool. Data decomposing, compressing and de-noising are some of the important tasks that can be performed by using Wavelet Transform.

In this thesis we have analyzed the SHM data of the Portage Creek Bridge by using artificial neural networks that can accept different sensor data as input and one sensor data as target. Such networks are utilized for identifying the relationship of the patterns of data from various sensors in the SHM system. We have de-noised all sensor data by using the wavelet transform before feeding them as input to the ANN. In this analysis the data have been harvested from the SHM database of the Portage Creek Bridge (ISIS SHM Database). We have considered eight bidirectional sensors and one temperature gauge data of column-2 of pier-2 of the Bridge. These eight sensors can measure sixteen strain values both in horizontal and vertical directions. Firstly we collected data for eight months period in 2006 for tracking the changes in data pattern for every 3-5 days period of each month. In this case we have observed that a trained network can not properly track the changes of data pattern if the network is not retrained progressively. Secondly, we have tried to track the changes in data pattern of winter (January, 2004, 2005 & 2006) and summer (July, 2003, 2004, 2005 & 2006) by two separately trained networks, one for the winter and other for the summer. From this analysis we observed that the performance of the trained networks do not increase in comparison to the previous results due to seasonal effect. Then we assumed that if a network can be trained with maximum number of inputs, it can store maximum data pattern in its memory and it can efficiently track the changes of data pattern if it is presented with new data immediately after the training period. With this assumption we trained one network taking data of three months (December, January & February, 2006) as input and one strain gauge (SG1_1) as target. Also we present this network with the data of March, 2006 to track the changes. In this case we got satisfactory response of the network. From this plot of network output and the present target (data of SG1_1) we can see that there is only a big hump in the data pattern. At the preliminary stage we assumed that the reason of this hump in the data pattern was due to the defective sensor and did further investigation by developing an ANN algorithm. For developing an appropriate ANN algorithm, we first established the effectiveness of the binary and sequential search method by manually changing the data for one sensor (SG6_1) among sixteen sensors of the first 15 days data of March, 2006. Also the response of the network in presence of defective sensor has been revealed that R-squared value is maximum when the defective sensor has been removed. In reality we do not know a priori which sensor is defective and as such we considered the maximum value of the R-squared as main indicator in detecting the defective sensor. Then we applied this algorithm to the data from the second half of March, 2006 to find out the defective sensor that caused the hump in the data pattern.

In finding a defective sensor, preliminarily we applied only the sequential search method. But in this method we needed to train many networks and the process takes long time. To avoid this drawback we have applied the binary search method to identify a smaller group that contains defective sensors and subsequently used the sequential search method to identify defective sensor from this smaller group.

Above all from the present analysis and verification it has been observed that the techniques developed in this thesis can be extended in real time monitoring of structures. Using the trained network a system can be developed which can successfully be applied in the continuous monitoring of valuable structures. The model of this system has been shown with block diagram as in Figure-7.1. In this system trained network will be validated with new sensor outputs. If the validation is OK, the network will progressively be updated again with new sensor data of SHM. But if the validation is not OK, then further investigations need to be done to identify the existence of Sensor malfunctioning, Structural changes and Seasonal changes. In this case, retraining of the network might be done if required.

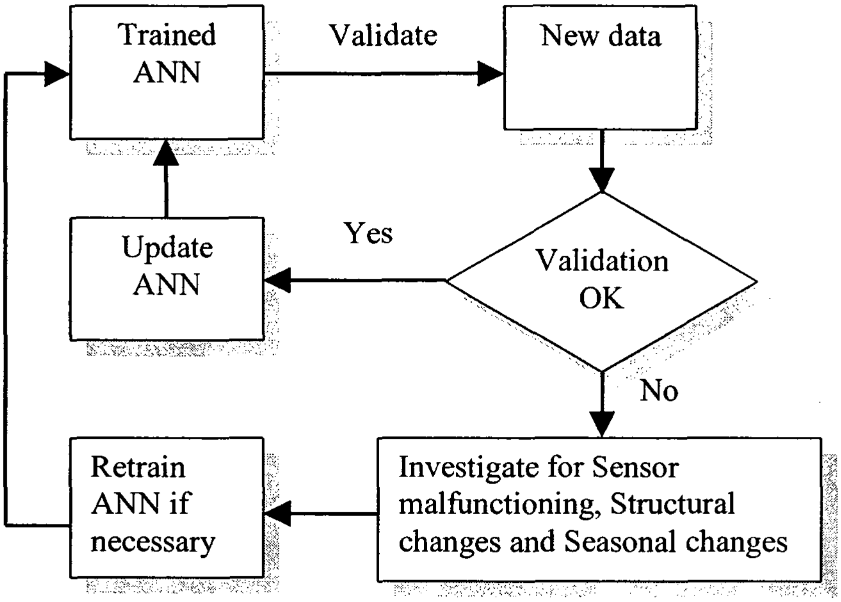

Figure-7.1: Diagram for potential real time monitoring using the developed techniques

7.2 Conclusions

In this thesis we have developed a signal based monitoring system by de-noising the sensor data using wavelet transform and presented these de-noised data into ANN to assess the pattern of the outputs, which can provide some insight on the integrity of the data used for SHM. Then we have applied this method to analyze the monitoring data of the Portage Creek Bridge, BC, Canada. From these analyses and the corresponding results, the following conclusions can be made:

- In a SHM system, it is very important to ensure that all sensors are working properly without any defect in it. It is not possible to find out the defective sensor by visual inspection of the sensors or direct observation of the signals. Therefore, the application of ANN pattern recognition algorithm in association with binary and sequential search process to find out the defective sensor is an innovative step in SHM

- ANN as pattern identifier can track the changes in data pattern as a continuous monitoring tool. But to ensure more accuracy in this procedure, the network should be retrained after a certain period as a continuous process with maximum number of training inputs so that it can store maximum pattern in its memory. Then this trained network should be presented with new inputs immediately after the training
- Data de-noising by wavelet transform and data normalization has been found more effective before training and presentation of ANN
- By capturing and plotting the output signal (signal blocks) from any sensor of the testing column, we can verify the strain due to the traffic load.

7.3 Limitations and future works

- In this thesis we have discussed only the method of detection of single defective sensor from a series of sensors in a SHM system. It is dealt within the context of binary search method applied to the data set for the second half of March, 2006. There might be more than one defective sensor in the series and that need to be detected. This work can be done by following the similar procedure too. Firstly, by eliminating one sensor, secondly by eliminating two sensors at a time and as such increasing the number of elimination from a group of sensors and plotting the R-squired values. This procedure needs more networks to be constructed, trained and simulated. However a thorough study and elaborate analysis on this topic can be considered as a future work.

- Only one data source (i.e., the data for the Portage Creek Bridge) has been used here to develop and validate the methods developed here. Further work need to be done considering different data sources.

- As because we don't know the actual value of output from the defective sensor, we have randomly chosen three (6.3.5) cases of defective sensor data. Further work can be done to simulate the defective sensor data and the analysis of it.

- In our analysis we have converted minute data into average hourly data. With this conversion of data, there is a possibility of loosing important information in it. Future analysis considering the data with respect to the actual time interval needed to assess the correct data format.

REFERENCES

Abudayyeh, O., Bataineh, M.A. and Qader, I.A., 2004, "An imaging data model for concrete bridge inspection", Advances in Engineering Software 35 (2004) 473–480. © 2004 Elsevier Ltd. All rights reserved.

Alampalli, S. and Ettouney, M., 2006, "Structural Health Monitoring as a Bridge Management Tool", copyright ASCE, Structures 2006.

Anderson, J., Pellionisz, A. and Rosenfeld, E.,1990, "Neurocomputing 2: Directions for Research". Cambridge Mass.: MIT Press, 1990.

Article on collapse, 2007, "I-35W Mississippi River Bridge collapse": <http://en.wikipedia.org/wiki/I-35W_Mississippi_River_bridge>.

Adeli, H. and Jiang, J., 2009, " Intelligent Infrastructure", CBC Press, Taylor and Francis Group

Bagchi, A., Humar, J., Xu, H. and Noman, A. S., 2009, "Model based damage identification in a continuous Bridge using vibration data", ASCE Journal of Performance of Constructed Facilities, Accepted, June.

Bagchi, A., 2005, "Updating the mathematical model of a structure using vibration data" Journal of Vibration and Control, 11(12): 1469–1486.

Ben-Akiva, M. and Gopinath, D., 1995, "Modeling Infrastructure performance and User Costs", Journal of Infrastructure Systems, ASCE, Vol-1, No-1, pp.33-43.

Balageas, D., Fritzen, C.P. and Guemes, A., 2006, "Structural Health Monitoring". ©ISTE Ltd. 2006, USA & U.K.

Bagnariol, D., 2003, “Investigation of failure: Sgt. Aubrey Cosens V.C. Memorial Bridge”. Ministry of Transportation Ontario.

Card, L. and McNeill, D.K., 2004, “Novel event identification for SHM systems using unsupervised neural computation”. Proceedings of SPIE Vol. 5393, SPIE, Bellingham, WA.

Czepiel, E., 1995, “Bridge Management Systems; Literature Review and Search” Web Page: http://www.iti.northwestern.edu/publications/technical_reports/tr11.html.

Commission report, 2007, “De La Concorde Overpass failure”, ISBN 978-2-550-50961-5, Gouvernment du Quebec. Web : <www.joconl.com/article/id24924>.

CAN/CSA-S6-88, 1988, “Canadian Highway Bridge Design Code”. Canadian Standards Association, Mississauga, Ontario, Canada.

CAN/CSA-S6-00, 2005, “Canadian Highway Bridge Design Code, Release 2005”. Canadian Standards Association, Mississauga, Ontario, Canada.

CAN/CSA-S6-06, 2006, “Canadian Highway Bridge Design Code, Release 2006”. Canadian Standards Association, Mississauga, Ontario, Canada.

Cox, C. Philip, 1987, “A handbook of introductory statistical methods”. ©1987 by John Willey & Sons, Inc. p. 109,237

Devroye, L., Gyorfi, L. and Lugosi, G., 1996, “A Probabilistic Theory of Pattern Recognition”. Berlin: Springer-Verlag, 1996.

Datta,A., Mavroidis, C., Krishnasamy, J. and Hosek, M., 2007, “Neural network based fault diagnostics of industrial Robots using wavelet multi-resolution analysis” 1-4244-0989-6/07©2007 IEEE.

Demuth, H., Beale, M. and Hagan, M., 2008, “Neural Network toolbox User’s Guide”, The Mathworks Inc., 2008.

Daubechies, I. (1992), “Ten lectures on wavelets”, SIAM, Cambridge University Press.

FHWA, 1995, “Seismic Retrofitting Manual for Highway Bridges”. Federal Highway Administration, U.S. Department of Transportation.

Fang, X., Luo, H. and Tang, J., 2005, “Structural damage detection using neural network with learning rate improvement”, Computers and Structures 83 (2005) 2150–2161.

Graps, A.L., 1995, "An Introduction to Wavelets", published in the IEEE Computational Sciences and Engineering, Volume 2, pp.50-61.

Humar, J., Bagchi, A. and Xu, H., 2006, “Performance of Vibration-based Techniques for the Identification of Structural Damage”, Structural Health Monitoring-An International Journal, SAGE Publications, Vol 5(3): 0215–27.

Hudson, S. W., Carmichral III, R. F., Moser, L. O. and Hudson, W. R., 1987 “Bridge Management System”, TRB-NCHRP, Report 300.

Hagan, M.T., Demuth, H.B. and Beale, M.H., 2002, “Neural Network Design”, Publisher: Martin Hagan. Web: <http://hagan.okstate.edu/nnd.html>.

Hopfield, J.J., 1982, “Neural networks and physical systems with emergent collective computational abilities,” Proceedings of the National Academy of Sciences , Vol. 79, pp. 2554-2558.

Huffman, S., Bagchi, A., Mufti, A., Neale, K., Sargent, D. and Rivera, E., 2006, “GFRP Seismic Strengthening and Structural Health Monitoring of Portage Creek Bridge

Concrete Columns", The Arabian Journal for Science and Engineering, Volume 31, Number 1C. pp. 25-42.

ISIS Educational Module 5, 2004, "An Introduction to Structural Health Monitoring", Principal Contributor: *L.A. Bisby*, Ph.D., P.Eng. Department of Civil Engineering, Queen's University, Contributor: M.B. Briglio, ISIS Canada.

ISIS, 2009, Official website of ISIS Canada Research Network, http://www.isiscanada.com, University of Manitoba, Winnipeg, MB, Canada.

Johnston, D. W., Chen, C., and Abed-Al-Rahim, I., 1994, "Developing User Costs for Bridge management Systems", Transportation research Circular, TRB, Vol. 423, pp. 139-149.

Jain, A.K., Duin, R.P.W. and Mao, J., 2000, "Statistical Pattern Recognition: A Review" 0162-8828/2000 IEEE.

Jain, A.K., Mao, J. and Mohiuddin, K.M., 1996, "Artificial Neural Networks: A Tutorial", Computer, pp. 31-44, Mar. 1996.

Johnson, P.M., Couture, A. and Nicolet, R., 2007, "Report of the Commission of inquiry into the collapse of a portion of the de la Concorde overpass". p.191.

Kim, J-T. and Stubbs, N., 2003, "Crack detection in beam-type structures using frequency data". Journal of Sound and Vibration; 259(1):145–60.

Khiem, N. T. and Lien, T. V., 2002, "The dynamic stiffness matrix method in forced vibration analysis of multiple-cracked beam". Journal of Sound and Vibration; 254(3):541–55.

Karibasappa, K. and Patnaik, S., 2004, "Face recognition by ANN using Wavelet transform Coefficients". IE (I) Journal-CP, Vol 85, May 2004.

Kohonen, T., 1995, "Self-Organizing Maps". Springer Series in Information Sciences, vol. 30, Berlin.

Morcous, G., November, 2000 "Case-Based Reasoning for Modeling Bridge Deterioration." A PhD thesis in the department of Building, Civil & Environmental Engineering of Concordia University, Montreal, Canada.

Moore, M., Phares, B., Graybeal, B., Rolander, D., and Washer, G., 2001, "Reliability of Visual Inspection for Highway Bridges, Volume I: Final Report." Federal Highway Administration Report FHWA-RD-01-020.

McCulloch, W. and Pitts, W., 1943 "A logical calculus of the ideas immanent in nervous activity," Bulletin of Mathematical Biophysics. , Vol. 5, pp. 115-133.

Misiti, M., Misiti, Y., Oppenheim, G. and Poggi, J-M., 2008, "Wavelet Toolbox™ 4 User's Guide" The MathWorks, Inc., 2008.

Medda, A., Chicken, E. and DeBrunner, V., 2007, "Sigma-Sampling wavelet denoising for structural health monitoring", 1-4244-1198-X/07, IEEE, 2007.

Mufti, A., Bakht, B. and Limaye, V., 2000, " Instrumentation of Portage Creek Bridge". Technical Project Report T3.3.1, 2000.

Marwala, T., 2000, "Damage identification using committee of Neural Networks", Journal of Engineering Mechanics, ASCE, 126 (1), 43-50.

NeuroDimension, Inc., 2009: "NeuroSolutions: Introduction to Neural Networks" Web Page: <http://www.nd.com/neurosolutions/products/ns/whatisNN.html>.

Noman, A. S., 2008, "Application of Vibration-based methods and Statistical Pattern Recognition techniques to Structural Health Monitoring", M. A. Sc. Thesis, Department of BCEE, Concordia University.

Ovanesova, A.V. and Suarez , L.E., 1999, "Wavelet Application to Structural Dynamics", Conference Program, CRC'99, Mayaguez, Puerto Rico.

OIC, 2000, "Data Dolphin Data Collection Software"– User Manual, Optimum Instruments Inc.,Version 1.5, 2000.

Pittner, S. and Kamarthi, S.V., 1999, "Feature Extraction from Wavelet Coefficients for Pattern Recognition Tasks", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 1, pp. 83-88.

Parag C. DAS, 1999, "Management of Highway Structures", published by Thomas Telford Publishing.

Ripley, B., 1993, "Statistical Aspects of Neural Networks, Networks on Chaos: Statistical and Probabilistic Aspects". U. Bornndorff-Nielsen, J. Jensen, and W. Kendal, eds., Chapman and Hall, 1993.

Rumelhart, D.E. and McClelland, J.L., 1986, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", Vol. 1, Cambridge, MA: MIT Press.

Shi, Z.Y., Law, S.S. and Zhang L.M., 2000, "Damage localization by directly using incomplete mode shapes". Journal of Engineering Mechanics; 126(6):656–60.

Timothy C. Urdan, 2005, "Statistics in Plain English", Lawrence Erlbaum Associates, Publishers, London. p.106. < http://en.wikipedia.org/wiki/Mean_squared_error>.

Vapnik, V.N., 1998, "Statistical Learning Theory", New York, John Wiley & Sons, 1998.

Watanabe, S., 1985, " Pattern Recognition: Human and Mechanical". New York: Wiley

Wenyuan, C., Lei, Z. and Guotang, B., 2007, "Application of Neural Network and Wavelet Analysis in Monitoring Multiple Structural Damage" The Eighth

International Conference on Electronic Measurement and Instruments, ICEMI'2007.

Widrow, B. and Hoff, M.E., 1960, "Adaptive switching circuits,"1960 IRE WESCON Convention Record, New York: IRE Part 4, pp. 96-104.

Zang, C., Grafe, H. and Imergun, M., 2001, "Frequency-domain criteria for correlating and updating dynamic finite element models". Mech. Syst. Signal Process; 15(1):139–55.

Zou, Y., Tong, L. and Steven, G.P., 2000 "Vibration based model dependent damage (delamination) identification and health monitoring for composite structures". A review. J. Sound Vib; 230:357–78.