

**A COMPARATIVE STUDY OF VIBRATIONAL  
RESPONSE BASED IMPACT FORCE LOCALIZATION  
AND QUANTIFICATION USING DIFFERENT TYPES OF  
NEURAL NETWORKS**

**WANG YANRU**

**THESIS SUBMITTED IN FULFILMENT OF  
THE REQUIREMENTS FOR DEGREE OF MASTER OF  
MECHANICAL ENGINEERING**

**FACULTY OF ENGINEERING  
UNIVERSITY OF MALAYA  
KUALA LUMPUR**

**2018**

**UNIVERSITY OF MALAYA  
ORIGINAL LITERARY WORK DECLARATION**

Name of Candidate: Wang Yanru

Matric No: KQK160013

Name of Degree: Master of Mechanical Engineering

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

A Comparative Study of Vibrational Response Based Impact Force Localization and Quantification Using Different Types of Neural Networks

Field of Study: Vibration

I do solemnly and sincerely declare that:

- (1) I am the sole author/writer of this Work;
- (2) This Work is original;
- (3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
- (4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
- (5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
- (6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature

Date:

Subscribed and solemnly declared before,

Witness's Signature

Date:

Name:

Designation:

## Abstract

Impact force identification plays an extremely important role to monitor structure health, where impacts can damage the structure, such as moving vehicle and industrial machines. Direct identification is not efficient and difficult due to environmental constraints and complex machine movements. Therefore, a lot of indirect methods were proposed in past studies. ANNs is one of the methods, which has drawn attention by researchers in recent years due to its unique advantages compared with other indirect identification techniques. It can analyze complex relationship of nonlinear input-output by learning from datasets without any mathematical model. Past studies mainly used conventional network Multilayer Perceptron (MLP) for impact identification and it was already successfully applied in some fields. But there are some disadvantages for MLP, such as local minima and slow learning speed. Radial Basis Neural Network (RBFN) was proven more error-tolerant and better than MLP using input feature peak arrival time (PAT). In the previous study, more accurate input feature (i.e. minimum arrival time (MAT)) was proposed to compare the accuracy of RBFN and MLP. In this study, the effort to find better neural network algorithms in the application of impact force identification continues. Another two Adaptive Neuro-Fuzzy Inference System (ANFIS) and Generalized Regression Neural Network (GRNN) are proposed to study their effectiveness in solving impact identification problem. This is because GRNN can avoid local minima like MLP. In addition, it has similar architecture with RBFN. MLP & RBFN use Gaussian function as activation function, but GRNN has one extra special linear layer, where outputs are considered in this layer, thus GRNN's performance in impact force identification is expected to be good. Moreover, GRNN evaluates each output independently from the

other outputs. It may be more accurate than MLP when there are multiple outputs. In addition, ANFIS uses hybrid learning algorithm. It is mixed with least mean square and gradient descent method, which cause many advantages, such as much better learning ability and less computational time. Therefore, this study will compare the accuracy and the effectiveness of ANFIS and GRNN with the conventional RBFN and MLP algorithms through experimental verification. The results showed that the proposed neural networks GRNN and ANFIS were effective to identify impact force. The most appropriate neural network for impact force localization was GRNN, which improved the accuracy by 66.11% than previous algorithm RBFN. In addition, the most proper neural network for impact force quantification was ANFIS, which improved the accuracy by 42.35% than the common used neural network MLP.

## Abstrak

Pengenalpastian daya impak memainkan peranan yang amat penting untuk memantau kesihatan struktur, di mana kesannya boleh merosakkan struktur seperti memindahkan kenderaan dan mesin perindustrian. Pengenalan langsung adalah tidak cekap dan sukar disebabkan oleh kekangan alam sekitar dan pergerakan mesin yang kompleks. Oleh itu, banyak kaedah tidak langsung telah dicadangkan dalam kajian masa lalu. ANNs adalah salah satu kaedah yang telah mendapat perhatian dalam kalangan para penyelidik sejak beberapa tahun kebelakangan ini kerana kelebihan uniknya berbanding dengan teknik pengenalan tidak langsung yang lain. Ia boleh menganalisis hubungan kompleks output-input bukan linear dengan membelajari dari dataset tanpa sebarang model matematik. Kajian lalu menggunakan rangkaian konvensional Multilayer Perceptron (MLP) untuk mengenal pasti kesan dan ia telah berjaya digunakan dalam beberapa bidang. Tetapi terdapat beberapa kelemahan untuk MLP, seperti minima tempatan dan kelajuan pembelajaran yang perlahan. Radial Basis Neural Network (RBFN) terbukti lebih bertoleransi dengan kesilapan dan lebih baik berbanding dengan MLP yang menggunakan ciri input Peak Arrival Time (PAT). Dalam kajian terdahulu, ciri input yang lebih tepat (iaitu Minimum Arrival Time (MAT)) telah dicadangkan untuk membandingkan ketepatan RBFN dan MLP. Dalam kajian ini, usaha untuk mencari algoritma rangkaian neural yang lebih baik dalam penerapan pengenalan daya kesan berterusan. Satu lagi dua Adaptive Neuro-Fuzzy Inference System (ANFIS) dan Generalized Regression Neural Network (GRNN) telah dicadangkan untuk mengkaji keberkesananannya dalam menyelesaikan masalah pengenalan impak. Ini kerana GRNN boleh mengelakkan minima tempatan seperti MLP. Di samping itu, ia mempunyai seni bina yang serupa dengan

RBFN. MLP & RBFN menggunakan fungsi Gaussian sebagai fungsi pengaktifan, tetapi GRNN mempunyai satu lapisan linear khas yang istimewa, di mana output dipertimbangkan dalam lapisan ini, oleh itu prestasi GRNN dalam pengenalan daya impak dijangka menjadi baik. Selain itu, GRNN menilai setiap keluaran secara bebas daripada output yang lain. Ia mungkin lebih tepat daripada MLP apabila terdapat banyak output. Di samping itu, ANFIS menggunakan algoritma pembelajaran hibrid. Ini bercampur dengan kuadrat minima dan kaedah keturunan kecerunan, yang menyebabkan banyak kelebihan, seperti keupayaan pembelajaran yang lebih baik dan kurang masa pengiraan. Oleh itu, kajian ini akan membandingkan ketepatan dan keberkesanan ANFIS dan GRNN dengan algoritma RBFN dan MLP konvensional melalui pengesahan eksperimen. Keputusan menunjukkan bahawa rangkaian neural yang dicadangkan GRNN dan ANFIS berkesan untuk mengenal pasti daya impak. Rangkaian neural yang paling sesuai untuk penyetempatan daya impak ialah GRNN, yang meningkatkan ketepatan sebanyak 66.11% daripada algoritma RBFN sebelumnya. Di samping itu, rangkaian neural yang paling sesuai untuk kuantifikasi daya impak adalah ANFIS, yang meningkatkan ketepatan sebanyak 42.35% daripada MLP rangkaian saraf yang biasa digunakan.

## **Acknowledgements**

First of all, I would like to express my deepest appreciation to my supervisor, Dr. Khoo Shin Yee. At the beginning, I did not know how to do research and also knew nothing about my research field. Although I am interested in neural networks, it was very hard for me to start the research. Dr. Khoo taught me research methodology and some basic knowledge about vibration and neural networks. Once I met problems, he was always available to give me some advices and new ideas. I enjoyed the process of doing research due to his patient guidance step by step. Secondly, I would like to appreciate all the lectures who have taught and guided me throughout my postgraduate studies in University of Malaya. Finally, I would also like to express my deepest appreciation to my parents and friends. Without their support, encouragement and assistance, I cannot successfully complete my research project.

## Table of Contents

Abstract.....	ii
Abstrak.....	iv
Acknowledgements.....	vi
Table of Contents.....	vii
List of Figures.....	xi
List of Tables.....	xiii
List of Abbreviations.....	xiv
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1 Background.....	1
1.2 Problem statement and significance.....	2
1.3 Objectives of the research.....	4
1.4 Research gap and research flow of this study.....	4
<b>CHAPTER 2: LITERATURE REVIEW.....</b>	<b>7</b>
2.1 Force identification method.....	7
2.2 Overview of artificial neural networks.....	9
2.3 Fundamentals of artificial neural networks.....	10
2.4 Neural network model.....	12
2.4.1 Multilayer Perceptron.....	12
2.4.2 Radial Basis Function Network.....	14
2.4.3 Generalized Regression Neural Network.....	15

2.4.4 Adaptive Neuro-Fuzzy Inference System .....	17
<b>CHAPTER 3: RESEARCH METHODOLOGY.....</b>	<b>22</b>
3.1 Equipment and experimental set up .....	22
3.1.1 Test rig .....	22
3.1.2 Accelerometer .....	23
3.1.3 Impact hammer .....	24
3.1.4 Data acquisition system (DAQ) .....	25
3.2 Impact identification methodology .....	26
3.2.1 Time-domain feature extraction.....	26
3.2.2 Arrangements of sensors and trial impacts .....	27
3.2.3 Neural network schemes .....	28
3.2.4 Optimize network parameters .....	29
<b>CHAPTER 4: RESULTS AND DISCUSSION .....</b>	<b>36</b>
4.1 Results.....	36
4.1.1 Results for impact localization.....	36
4.1.2 Results for impact quantification .....	37
4.2 Discussion.....	38
4.2.1 Selection of best neural network for impact localization.....	38
4.2.2 Selection of best neural network for impact quantification .....	44
4.2.3 Comparison of impact localization and quantification results with the previous studies .....	47
<b>CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS .....</b>	<b>51</b>

5.1 Conclusions.....	51
5.2 Recommendations.....	52
References.....	53
Appendix A.....	58
Appendix B .....	65

University of Malaya

## List of Figures

Figure 1.1: Model of an input-output process.....	2
Figure 1.2: Work flow of the present study.....	6
Figure 2.1: Neuron cell.....	11
Figure 2.2: Model of artificial neuron.....	12
Figure 2.3: MLP architecture.....	12
Figure 2.4: RBFN architecture.....	14
Figure 2.5: GRNN architecture.....	16
Figure 2.6: ANFIS architecture for two-input Sugeno fuzzy model.....	18
Figure 3.1: Experimental Set-up for impact force identification.....	22
Figure 3.2: S100C ICP accelerometer.....	23
Figure 3.3: Mounting methods and their effects on accelerometer's sensitivity.....	24
Figure 3.4: Impact hammer.....	25
Figure 3.5: Data acquisition module.....	26
Figure 3.6: Arrangement of the sensors and impact locations.....	27
Figure 3.7: Structure of 10-fold cross validation.....	28
Figure 3.8: Impact (a) localization and (b) quantification schemes using various neural networks.....	29
Figure 3.9: Optimize hidden neuron number (MLP) for impact force localization.....	30
Figure 3.9: Optimize hidden neuron number (MLP) for impact force quantification.....	31
Figure 3.10: Optimize radius (ANFIS) for impact force localization.....	33
Figure 3.11: Optimize radius (ANFIS) for impact force quantification.....	34
Figure 4.1: (a) Subsample 9, which has best identification performance among all the	

subsamples for ANFIS; (b) Subsample 3, which has worst identification performance among all the subsamples for ANFIS.....	38
Figure 4.2: (a) Subsample 2, which has best identification performance among all the subsamples for GRNN; (b) Subsample 7, which has worst identification performance among all the subsamples for GRNN.....	40
Figure 4.3: (a) Subsample 1, which has best identification performance among all the subsamples for RBFN; (b) Subsample 3, which has worst identification performance among all the subsamples for RBFN.....	41
Figure 4.4: (a) Subsample 10, which has best identification performance among all the subsamples for MLP; (b) Subsample 3, which has worst identification performance among all the subsamples for MLP.....	41
Figure 4.5: (a) Radial error percentage within 0.1 cm for ANFIS (subsample 9); (b) Radial error percentage within 0.1 cm for GRNN (subsample 2); (c) Radial error percentage within 0.1 cm for RBFN (subsample 1); (d) Radial error percentage within 0.1 cm for MLP (subsample 10).....	43
Figure 4.6: (a) Subsample 8, which has best identification performance among all the subsamples for ANFIS; (b) Subsample 10, which has worst identification performance among all the subsamples for ANFIS.....	44
Figure 4.7: (a) Subsample 8, which has best identification performance among all the subsamples for GRNN; (b) Subsample 4, which has worst identification performance among all the subsamples for GRNN.....	45
Figure 4.8: (a) Subsample 2, which has best identification performance among all the subsamples for RBFN; (b) Subsample 4, which has worst identification performance among all the subsamples for RBFN.....	45

Figure 4.9: (a) Subsample 8, which has best identification performance among all the subsamples for MLP; (b) Subsample 9, which has worst identification performance among all the subsamples for MLP.....46

Figure 4.10: Comparison of success rate percentage within 5% for ANFIS and RBFN (subsample 2).....47

University of Malaya

## List of Tables

Table 1.1: Differences in approach between conventional computing and ANNs.....	2
Table 1.2: Previous impact force identification results.....	5
Table 3.1: Optimized parameters of different network for impact force localization.....	34
Table 3.2: Optimized parameters of different network for impact force quantification....	35
Table 4.1: Impact localization testing errors of subsequent subsamples for RBFN, GRNN, ANFIS and MLP.....	36
Table 4.2: Success rate percentage of impact localization for different subsamples.....	37
Table 4.3: Impact quantification testing errors of subsequent subsamples for RBFN, GRNN, ANFIS and MLP.....	37
Table 4.4: Success rate percentage of impact quantification for different subsamples...	37
Table 4.5: Comparison of testing impact data #27 with training data at location 2.....	39
Table 4.6: Iterations and training error to achieve error goal for different networks.....	39
Table 4.7: Localization results comparison of current study with previous studies.....	49
Table 4.8: Localization results comparison of current study with previous studies.....	50

### **List of Abbreviations**

ANNs	:	Artificial Neural Network
ANFIS	:	Adaptive Neuro Fuzzy Interface System
BPN	:	Backpropagation Network
CGBP	:	Conjugate Gradient Backpropagation
DAQ	:	Data Acquisition
DOF	:	Degree of Freedom
FRF	:	Frequency Response Function
GRNN	:	Generalized Regression Neural Network
MAT	:	Minima Arrival Time
MATLAB	:	Matrix Laboratory
MBP	:	Marquardt Backpropagation Algorithm
MLP	:	Multilayer Perceptron
MSE	:	Mean-squared-error
PAT	:	Peak Arrival Time
RBFN	:	Radial Basis Function Network
RMS	:	Root-mean-square
SVD	:	Singular Value Decomposition
SVM	:	Support Vector Machine
TC	:	Threshold-Crossing

## CHAPTER 1: INTRODUCTION

### 1.1 Background

Impact force is a high force or shock over a short time period when two or more bodies collide. It is common to encounter in vehicles, aircrafts and industrial machines. It happens with brittle deformation other than resilient deformation, which generates more severe effect than other forces. In some situations, it can cause extremely destructive effect to the bodies. For example, impact force happens when vehicles collide. Therefore, localization and quantification of impact force are very important so that some protective measures can be taken in advance, taking necessary maintenance work timely or monitoring the dynamic condition of any object. In addition, design modification of the structure subsequently can be made by analyzing the impact force of damaged structure. Analytical method is the common used method among researchers. It performs quite well in most situations. But direct identification of impact force is not feasible and efficient due to limited surrounding conditions and economic reasons (Md Sazzad et al., 2017). For example, it is hard to identify the impact force in a moving vehicle. Even it is possible to make direct identification in the structure, a lot of sensors should be put in proper positions. It is quite difficult to find a proper mathematical function or theoretical relation between response and force in such a complicated nonlinear vibration system. In addition, the predicted accuracy is low due to the presence of external interference signals. Therefore, indirect methods were developed to identify impact force by researchers (Md Sazzad et al., 2017). Many researchers did a series of studies and they found artificial neural networks (ANNs) had outstanding performance to identify and quantify impact force compared with analytical methods (Sam et al., 2013; Liu et al., 2011; Mohfouz,

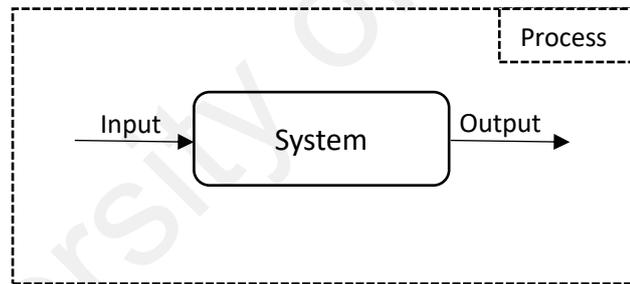
2011; Ozkaya et al., 2002). Table 1.1 shows the differences in approach between conventional computing and ANNs.

**Table 1.1:** Differences in approach between conventional computing and ANNs

Characteristics	Conventional computing (including expert systems)	Artificial neural networks
Learning method	By rules	By example (Socratically)
Functions	Logically	Perceptual pattern
Processing style	Sequential	Parallel

## 1.2 Problem statement and significance

Identification of impact force is an inverse problem (Ghajari et al., 2013). Mathematical model and mechanical structure are built to determine the external force. The input or system is estimated from its response, either of them is known (Fig. 1.1).



**Fig. 1.1:** Model of an input-output process

The response of structure can be obtained by mounted sensors. Equation of vibrating system can be expressed as follows (Ma Sazzad et al., 2016):

$$[M]\{\ddot{X}(t) + [C]\{\dot{X}(t)\} + [K]\{X(t)\} = \{F(t)\} \quad (1.1)$$

Eq. (1.1) shows input is external force, and outputs are displacement, velocity and acceleration, while mass, damping, stiffness, boundary condition, etc. represent the system. In practice, the force identification process is very complicated due to the ill-

posedness, thus it is difficult to find an accurate solution (Qiao, Zhang et al., 2014). In this study, artificial neural network is suggested to identify the impact force instead of the conventional inversion force identification method. Application of neural networks attracted increasing attention in industries and modern research labs due to its capabilities that consider effect of noises or uncertainties in ambient environment (Chandrashekhara et al., 1997; Md Sazzad et al., 2016). Multilayer perceptron network was first used to identify impact force due to its straightforward mechanism developing both simple and complex input-output relationship. It has a high accuracy in force identification problem. But it presents some disadvantages, such as low convergence speed and achieving local optimization rather than global optimization and so on. In order to avoid these problems and improve predicted accuracy, several studies used RBFN and GRNN to achieve global minima instead of local minima (Chen et al., 2015; Worden et al., 2000; LeClerc, et al., 2017). Md Sazzad et al. compared the performance of MLP and RBFN using PAT as input feature, and found RBFN was better than MLP (Md Sazzad et al., 2017). He also compared the performance of PAT and MAT by delivering the input signals to MLP, and found that MAT feature had higher accuracy than PAT for impact force localization (Md Sazzad et al., 2017). Based on the previous studies, MAT feature was chosen as input to study the performance of MLP and RBFN for localization. In terms of quantification, peak-to-peak combined MAT were applied as input feature, where the accuracy would be examined. In addition, other two preferred neural networks are studied to solve force identification problems, which are generalized regression neural network (GRNN), and adaptive neuro-fuzzy inference system (ANFIS).

### **1.3 Objectives of the research**

Based on the limitation of previous studies, the objectives of this study can be summarized as follow:

- To study the effectiveness of GRNN and ANFIS in terms of accuracy and precision in force identification problem.
- To propose the most appropriate neural network for impact force localization and quantification problem.

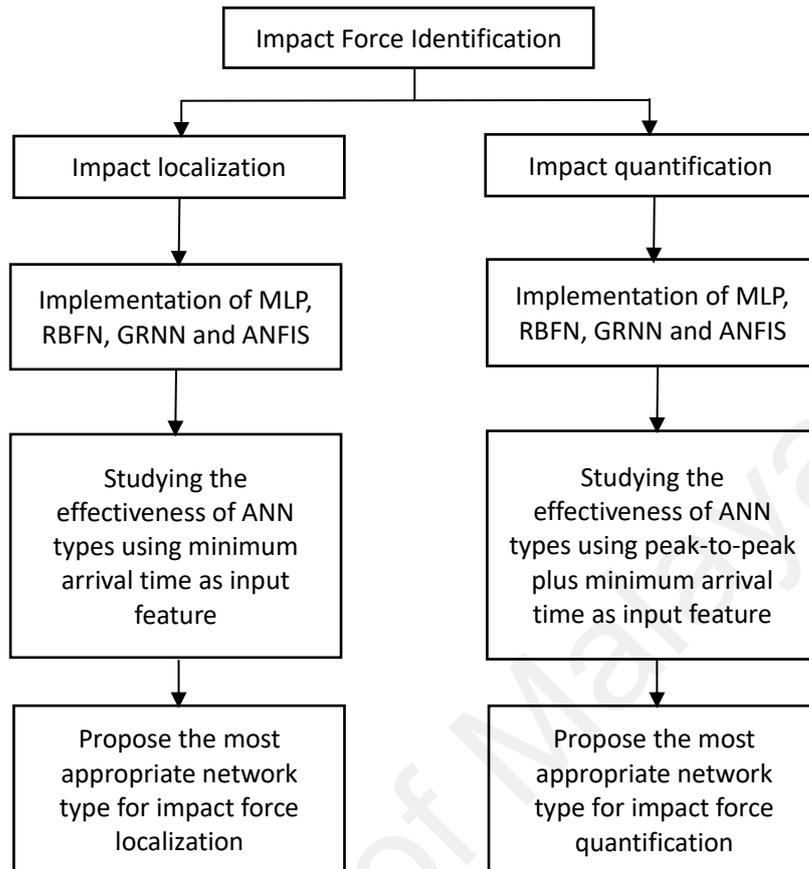
### **1.4 Research gap and research flow of this study**

Previous researchers (Md Sazzad et al.) have completed a series of studies in impact force identification and their work showed the impact force identification results using neural network technique as it needs very little training data with high force identification accuracy. Table 1.2 shows the results of their studies. Paper 1 (Md Sazzad et al., 2017) demonstrated that RBFN was better than MLP by using feature PAT or peak-to-peak for impact identification. Paper 2 (Md Sazzad et al., 2017) indicated that MAT performance was better than PAT for impact localization, where same neural network MLP was used to compare. Moreover, Md Sazzad (2016) presented that RBFN combined MAT feature was better than MLP combined PAT feature for impact force localization in his thesis. So far the performances of “RBFN+MAT” and “MLP+MAT” have not been studied. Therefore, MAT feature was chosen for this study to compare the performance of RBFN and MLP for impact force localization. In terms of quantification, “RBFN+Peak-to-peak+MAT” performed better than “MLP+Peak-to-peak+PAT”. Based on this finding, “MLP+Peak-to-peak+MAT” was proposed to compare the prediction accuracy with “RBFN+Peak-to-peak+MAT”. In addition, another two networks ANFIS and GRNN were proposed to study their effectiveness for impact identification problem against the

conventional algorithm used in the field (i.e. RBFN and MLP). Furthermore, it was clear to show that “RBFN+Peak-to-peak” could acquire smaller testing error than “RBFN+Peak-to-peak+MAT”. But it could not demonstrate that Peak-to-peak was better than Peak-to-peak combined MAT due to different subsample used in different study. In order to make sure the robustness of the proposed algorithm, force identification by using 10 subsamples would be implemented in this study. Besides, previous studies did not optimize the parameter well and chose the parameter by trial and error, this study gives a systematic way to optimize the parameter for neural network. The research flow of this study is shown in flow chart below (Fig. 1.2).

**Table 1.2:** Previous impact force identification results by Md Sazzad et al.

Various studies	Different networks combined with different features for localization testing	Localization		Different networks combined with different features for quantification testing	Quantification	
		Error range (radial cm)	Success rate (%)		Error range (%)	Success rate (%)
Paper 1 (1 subsample)	MLP+PAT	1.8825	84% (within 2 radial cm error)	MLP+Peak-to-peak	7.4925	80.5% (within 10% error)
	<b>RBFN+PAT</b>	1.2625	92% (within 2 radial cm error)	<b>RBFN+Peak-to-peak</b>	6.21	99.5% (within 10% error)
Paper 2 (10 subsamples)	MLP+PAT	1.64 (1.17-2.01)	87.58% (77.2%-93.67%)(within 2 radial cm error)	N/A	N/A	N/A
	<b>MLP+MAT</b>	0.76 (0.53-1.02)	96.72% (94.13%-99.2%)(within 2 radial cm error)	N/A	N/A	N/A
Thesis (1 subsample)	MLP+PAT	3.02	N/A	MLP+(Peak-to-peak+PAT)	13.91	N/A
	<b>RBFN+MAT</b>	0.7497	N/A	<b>RBFN+(Peak-to-peak+MAT)</b>	7.49	N/A
Current study (10 subsamples)	MLP+MAT	To be examined in this study		MLP+(Peak-to-peak+MAT)	To be examined in this study	
	RBFN+MAT					
	GRNN+MAT					
	ANFIS+MAT					



**Fig. 1.2:** Work flow of the present study

## CHAPTER 2: LITERATURE REVIEW

### 2.1 Force identification method

Indirect method is used for impact force identification. It will encounter ill-posed problem due to the unstable operation environment. The accuracy was already improved with the development of modern technologies. Indirect method can be divided to two major classes, i.e. direct inverse method and soft computing method. Direct inverse method is build based on time domain and frequency domain response. Frequency domain signal is obtained by Fourier transformation to transfer time domain signal over a certain period. But most of the time, the transfer functions fail to transfer the signal near the region of resonance frequency. Hence small variation of response will result in large changes of identified force. Besides, the lack of information may result in an ill-posed problem. For example, a finite number of measured responses are used to identify impact force, while the actual responses is a continuum. A variety of regulation methods are used to overcome the ill-posed problems, such as Tikhonov regularization and Singular Value Decomposition (SVD). It was applied by a lot of researchers, who proved regulation methods were very useful to overcome ill-posed problem (Kim & Lee, 2008; Liu & Shepard, 2005; Thite & Thompson, 2003). But there still are problems exist, such as choosing appropriate regularization parameters and high computing cost and so on. Due to many disadvantages using frequency domain response, time domain based direct inverse method is more popular. It represents the real time information about impact force, but the results can be obtained based on theoretical modeling of structures without considering the surrounding uncertainties or noise. Thereby, ANNs have played an important role in recent years. It can avoid the mentioned drawbacks above by treating

the system as a black box (i.e. without the needs of mathematical model and hence it does not suffer ill-posed problem). Hagan and Menhaj (1994) compared the accuracy for function approximation problem among Marquardt backpropagation algorithm (MBP), Backpropagation with variable learning rate (VLBP) and Conjugate gradient backpropagation (CGBP) by training feedforward neural networks. They found that Marquardt algorithm took less computational time and less iterations to converge compared with the other two algorithms for sine wave, square wave, 2-D sinc function and 4-D function problems. Thus Marquardt algorithm will be applied in this study. Jones and Sirkis (1997) developed neural-based method to determining experimental impact locations and magnitudes on a fully-clamped isotropic plate. They found that the RMS error became less after using data manipulation techniques and the accuracy can be improved by increasing the training set. Chandrashekhara et al. (1997) determined the contact force on laminated composite plates using finite element analysis and neural networks (Chandrashekhara et al., 1997). It was proven that neural network could be a promising alternative approach compared with traditional techniques. Ghajari et al. (2013) developed neural networks to identify impact force in composite stiffened panels compared with traditional method, and found the accuracy was improved using neural networks. Moreover, he categorized the large mass impact force and small mass impact force to trained separated networks for each of them, and found the prediction accuracy was improved. Yang, Yan & He (2016) identified the location and magnitude of moving load by using multilayer perceptron network. Constant moving load was proven achieving good recognition accuracy, while time-varying force was not better as constant moving load. But it still can identify load value. Moreover, time-domain features was very popular among researchers in recent years, such as PAT, MAT, peak-to-peak and TC and so on,

where most commonly used feature was PAT (Ghajari et al., 2013; Sharif et al., 2012; Ghajari et al., 2012). TC was also robust to identify impact force. Several studies used TC as input feature for impact localization (Ghajari et al., 2012; Sharif et al., 2012; Haywood et al., 2004). Md Sazzad et al. (2017) compared different input features and found that MAT could achieve better localization accuracy of impact force. Besides, peak-to-peak was impact magnitude dependent and was successfully used in impact force quantification (Worden & Staszewski, 2000).

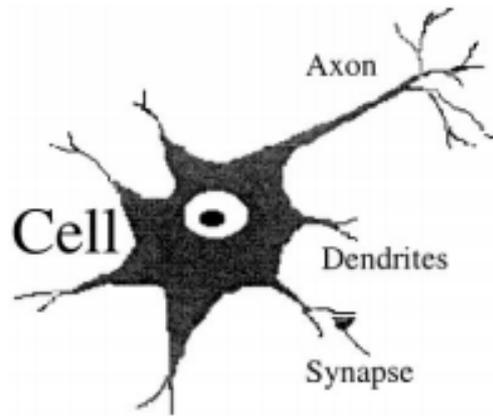
## **2.2 Overview of artificial neural networks**

Artificial neural network originates from the 1940s, when researchers tried to find algorithm to mimic human brain. Walter Pitts invented computational model based on mathematics and algorithms in 1940, which is the original model of artificial neural network (Walter Pitts, 1940). It paved the way to use this computational model to practice situations later (Warren et al., 1943). Rosenblatt (1958) invented an effective algorithm for pattern recognition in 1958, which he named “perceptron”. But he described exclusive-or circuit could not be processed by basic perceptron. In addition, Minsky and Papert proposed that computer did not have enough processing power to process large neural networks (Minsky et al., 1969). The development of neural network became slow until computers achieved far greater processing power. Besides, further modifications on perceptron were done by several researchers. Werbos (1975) invented back propagation algorithm, and it effectively solved exclusive-or questions. In addition, it had successful application in practical situation, such as autonomous car and character recognition. But there still had many open questions, such as over-fitting, network structure and bad local minimum points and so on. Poor results could be generated by training deep neural

networks that include more than one hidden layer. SVM (support vector machine) was developed by Vapnik et al. in 1995, which was a shallow architecture. It became very popular and overtook the development of artificial neural networks. Hinton et al. (2006) introduced successive layers with restricted Boltzmann machine to learn high-level representation. As long as sufficient layers were developed, the neural network could process deep learning efficiently. Ng and Dean developed a neural network to recognize higher-level concepts, such as cat, car and people (Ng & Dean, 2012). In recent years, artificial neural networks are vastly used and it becomes popular in solving numerous vibration related problems. It was proven to be a good method to monitor and solve structural health-related problems, mechanical faults diagnosis and identification problems of mechanical parameters (Carden & Fanning, 2004; Chang & Zhou, 2002; Doebling et al., 1996; González & Zapico, 2008; Hakim & Abdul Razak, 2014b; Mangal et al., 1996; Shiguemori et al., 2004; Worden & Dulieu-Barton, 2004; Xiao et al., 2007).

### **2.3 Fundamentals of artificial neural networks**

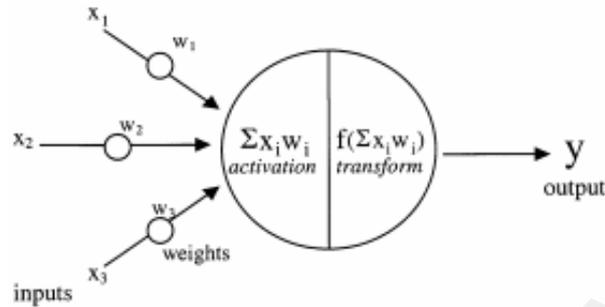
Artificial neural network is a computational algorithm which resembles biological neural network of animal nervous system to automatically detect the magnitude of impact force, as well as the location from a series of giving responses. It mimics the brain. For average brain, there are 100 billion neurons and each neuron has 100-1000 connections with others (Kustrin & Beresford, 1999). Neurons consist of a cell body which controls the cell activity (Fig. 2.1). Dendrites carry information to neuron cell. Axon takes away the signal from the cell body and pass it to synapse, which is the junction between one neuron and the next one (Kustrin & Beresford, 1999). The whole brain is a fully connected network and it can be used to learn by itself.



**Fig. 2.1:** Neuron cell (Kustrin & Beresford, 1999)

A single artificial neuron includes four core elements: inputs, net function, transfer function and one output (Fig. 2.2). Inputs multiply synaptic weight to pass through transfer function, such as sigmoid function, hyperbolic tangent sigmoid and Gaussian and so on. Most functions can be approximated by a single hidden layer (Bill, 2012). Synaptic weight is random value at initial training process. The objective of ANN is to minimize training error by adjusting these weight values by learning process. There are two connection types of neural networks, feedback and feed forward architecture. Feedback network has connections from output to input neurons. The output can feedback the result to hidden layer. The synaptic weight is repeatedly adjusted until the error goal is achieved. Training error is the difference between predicted value and target and it should be designed more than zero to avoid over fitting (Md Sazzad et al., 2016). Feed forward network do not have connections from output to input neurons. It cannot memorize the previous state of neural network. Most common used training algorithms are Levenberg-Marquardt, Quasi-Newton and Conjugate Gradient and so on (Hagan et al., 1994; Zakaria et al., 2010; Chen et al., 1991). There are two method for training process, supervised and unsupervised. Supervised learning is common used method for prediction and

classification (Kustrin & Beresford, 1999). It has exact output for training process. Unsupervised learning does not have desired output goal for training process. It is used to find relationship among complicated sets of data (Kustrin & Beresford, 1999).

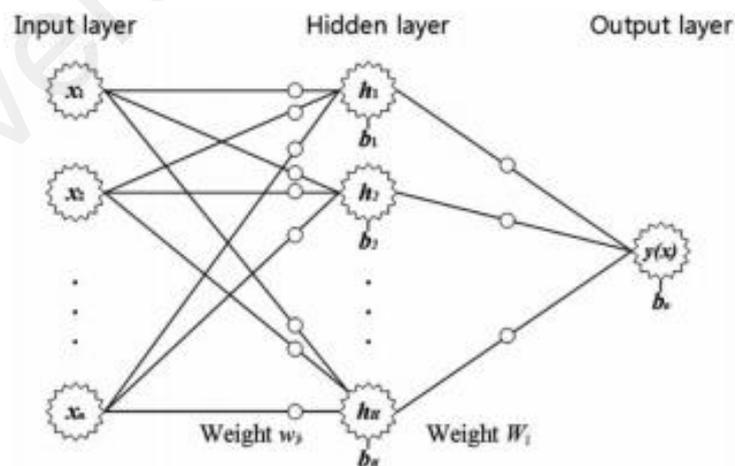


**Fig. 2.2:** Model of artificial neuron (Kustrin & Beresford, 1999)

## 2.4 Neural network model

### 2.4.1 Multilayer Perceptron

MLP is the most frequently used neural network with supervised learning technique. It can be used for classification and function approximation. It includes three layers, which are input layer, multi or single hidden layer and output layer. Bias is added to each neuron of hidden layer. Fig. 2.3 shows a simple architecture of MLP with one hidden layer.



**Fig. 2.3:** MLP architecture (Kim & Parnichkun, 2016)

Previous study shows that MLP with one hidden layer is capable to perform universal approximation (Hecht-Nielsen, 1987). Thus one hidden layer was chosen for this study. Weighted input is passed through hidden layer, then arrive at output layer (Eq. 2.1).

$$u = b + \sum_{j=1}^N w_j x_j \quad (2.1)$$

Where  $u$  is the weighted input;  $b$  is bias weight;  $w$  is weight value for each input;  $N$  is number of input and  $x$  is input. The default activation function is hyperbolic tangent sigmoid (Eq. 2.2).

$$y = \frac{e^u - e^{-u}}{e^u + e^{-u}} \quad (2.2)$$

Where  $u$  is the weighted input;  $y$  is output from hidden layer.

Backpropagation is a very popular training algorithm. The default training function is Levenberg-Marquardt. The training program is try to find the global minima by gradient descent, but sometimes the training stops when the local minima are found. BP algorithm has a performance index, which is the least mean square error (Lee, 2003; Civalek, 2004; Noorzaei et al., 2007). It uses mean square error (MSE) to calculate error between desired output and predicted value. The weight and bias value are adjusted to minimize mean square error.

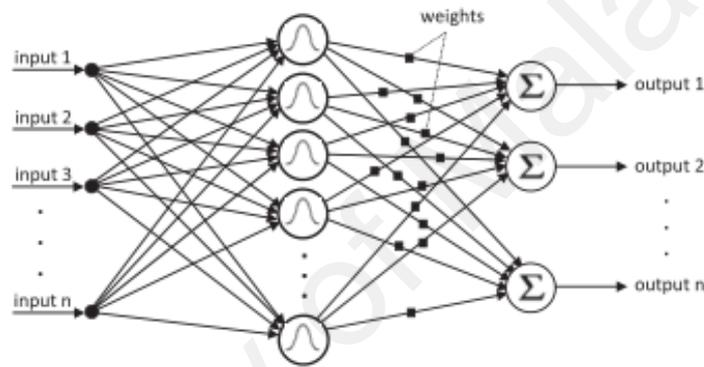
The important thing for MLP is to optimize maximized hidden neuron. Different numbers of hidden neuron are selected to training to find the best one that has least mean square error. RBFN training code can be found in terms of Neural Network Toolbox (Demuth, Beale, & Hagan, 2008). It is given by,

$$\text{Net} = \text{newff}(P, T, S)$$

Where  $P$  is the input vector;  $T$  is the output vector;  $S$  is the maximum number of neurons.

## 2.4.2 Radial Basis Function Network

RBFN is a feed-forward network with supervised learning technique. It comprises of input, one hidden layer and output. The architecture of RBFN is showed in Fig. 2.4. Gaussian function (Eq. 2.3) is generally used activation function for RBFN. It is non-linear transfer function, which transforms the input signal into another form, then gets the output by passing linear summation. Euclidian distance was used to determine the distance between input vectors. In addition, spread constant was a parameter that should be optimized before training.



**Fig. 2.4:** RBFN architecture (Md Sazzad et al., 2017)

$$\varphi_i(x) = e^{-\frac{\|x-c_i\|^2}{\sigma^2}} \quad (2.3)$$

Where  $x$  is the input vector,  $c_i$  is the center vector of  $i$ th hidden neuron and  $\sigma$  is a predefined spread value of the function ( $\sigma > 0$ ). Eq. 2.4 gives the output of RBFN that equal to the weighted sum of the hidden neurons' responses.

$$y_j = \sum_{i=1}^n w_{ij} \varphi_i(\|x - c_i\|) + w_{oj}, j = 1, 2, \dots, n \quad (2.4)$$

Where  $n$  is the number of nodes in hidden layer,  $x$  is input of the network,  $c_i$  is the center vector of  $i$ th hidden neuron,  $w_{ij}$  is weight of  $i$ th neuron of the hidden layer,  $\varphi$  is Gaussian function and  $w_{oj}$  is bias of the  $j$ th neuron of output layer.

The important parameters for RBFN are predefined spread constant and optimized

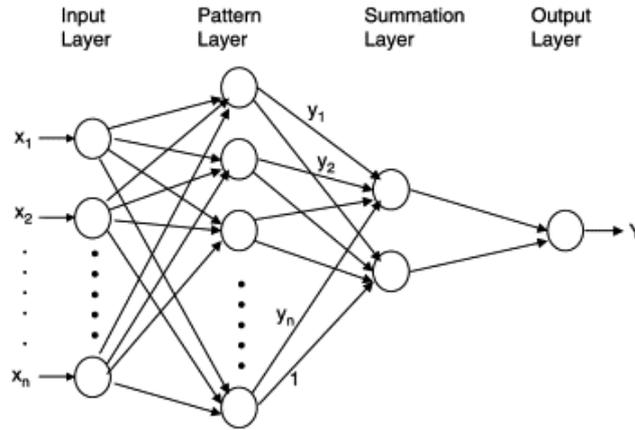
maximized neuron. The spread constant of Gaussian function should be lied between the minimum distance of two adjacent input and the maximum distance of any pair of input vectors. The optimization of spread constant should according to the mean-square-error (MSE). RBFN training code can be found in terms of Neural Network Toolbox (Demuth, Beale, & Hagan, 2008). It is given by,

$$\text{Net} = \text{newrb}(\text{P}, \text{T}, \text{GOAL}, \text{SPREAD}, \text{MN}, \text{DF})$$

Where P is the input vector, T is the output vector, GOAL is error goal, SPREAD is predefined parameter, MN is the maximum number of neurons and DF is neuron number to add between displays.

### **2.4.3 Generalized Regression Neural Network**

GRNN is a variation of RBFN (Shaikh et al., 2010), which can be used for function approximation. It was proposed by D. F. Specht, which does not require an iterative training procedure and approximates any arbitrary function between input and output vectors (D. F. Specht, 1991). GRNN includes four layers, which are input layer, pattern layer, summation layer and output layer (Fig. 2.5). By definition, y is dependent variable and x is independent variable, where x estimates a most probable value for y by regression (Hikmet & Murat, 2006). GRNN has very fast learning speed and convergence to the optimal regression surface when the numbers of sample data are very large, but it still can get good forecasting result when the datasets are very small (Hong et al., 2013; Mahmood et al., 2010).



**Fig. 2.5:** GRNN architecture (Hikmet & Murat, 2016)

Inputs are fed from input layer to pattern layer or radial basis layer. The activation function is Euclidean function in this layer, which is same with hidden layer of RBFN. It decides that how much weight the training sample will contribute. The next layer is summation layer, which includes two parts: numerator part and denominator part. Output part includes one neuron by dividing the numerator part of the summation layer by the denominator part. GRNN works by measuring how far given samples pattern is from patterns in the training set and it is no training patterns, such as optimum number of neurons in hidden layer, learning rate and learning algorithms and so on like MLP (Reza Rooki, 2016). The output can be obtained according to Eq. 2.5 and it is weighted exponentially from Gaussian distance. Gaussian distance measures how well the each training sample can represent the prediction position.

$$Y(x) = \frac{\sum Y_i e^{-\frac{d_i^2}{2\sigma^2}}}{\sum e^{-\frac{d_i^2}{2\sigma^2}}} \quad (2.5)$$

Where  $d_i^2 = (x - x_i)^T (x - x_i)$ ,  $x$  is the input sample,  $x_i$  is the training sample,  $Y_i$  is the output sample of input sample,  $d_i$  is the Euclidean distance from  $x$  and  $x_i$ ,  $\sigma$  is the

predefined spread value.

The only one unknown parameter is spread constant  $\sigma$ , which can be obtained by training process to get an optimum value, where the MSE (Mean Squared Error) is the minimum. GRNN training code can be found in terms of Neural Network Toolbox (Demuth, Beale, & Hagan, 2008). It is given by,

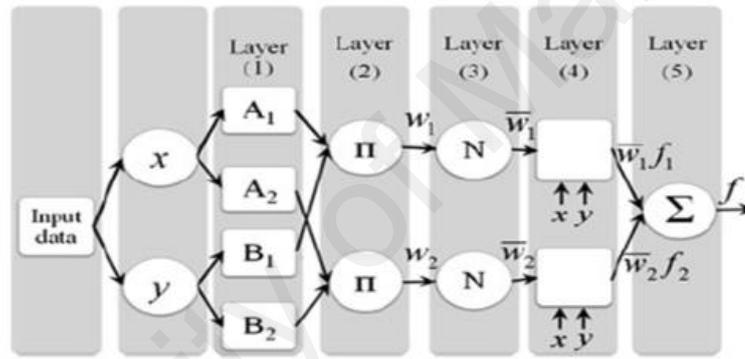
$$\text{net} = \text{newgrnn}(\text{P}, \text{T}, \text{SPREAD});$$

Where P is input vector, T is output vector and SPREAD is predefined constant for the optimized GRNN structure.

#### **2.4.4 Adaptive Neuro-Fuzzy Inference System**

Adaptive Neuro-Fuzzy Inference System (ANFIS) is feedforward multilayer neural network with adapted nodes, which incorporates both neural networks and fuzzy logic principles. Neural networks utilize historical dataset for prediction of future values. However, it is difficult to interpret the knowledge acquired by it, as meaning associated with each neuron and each weight it quite complex to comprehend (Neha et al., 2016). For fuzzy logic, control signal is generated from the rule base, which is drawn on historical data random in nature. The output is also random which may prevent optimal results. Fuzzy logic cannot learn from the data, but fuzzy-based models are easily understood and it utilizes linguistic terms (words or sentences) and the structure of if-then rules. It can change the qualitative aspects of human knowledge to precise quantitative analysis. ANFIS can avoid these disadvantages and make the selection of the rule base more adaptive to the situation. The architecture of ANFIS was first introduced by Jang in 1993 (Jang, 1993). The parameters of adaptive nodes are adjusted by specifying the error

terms. The output is predicted according to the parameters of adaptive nodes. It combines neural networks and fuzzy logic principles, thus it uses a hybrid learning algorithm, namely gradient descent approach and least mean square. Gradient descent approach is used to adjust non-linear parameters ( $a_i, b_i, c_i$ ) in backward pass. Least square method is applied to adjust linear parameters ( $p_i, q_i, r_i$ ) (Chang et al., 2006), which reduces the computational effect and time in forward pass. The Sugeno fuzzy model is utilized in fuzzification and defuzzification of the system (Lin & Huang, 2012) for ANFIS model. Basic structure of ANFIS is introduced in terms of two-input Sugeno fuzzy model (Fig. 2.6).



**Fig. 2.6:** ANFIS architecture for two-input Sugeno fuzzy model (Hakim & Razak, 2013)

For a first-order Sugeno fuzzy model, it has two fuzzy if-then rules in the common rule set (Takagi & Sugeno, 1985) as given in (Eq. 2.7) and (Eq. 2.8).

$$\text{Rule 1: If } x \text{ is } A_1 \text{ and } x \text{ is } B_1, \text{ then } f_1 = p_1x + q_1y + r_1 \quad (2.7)$$

$$\text{Rule 2: If } x \text{ is } A_2 \text{ and } x \text{ is } B_2, \text{ then } f_2 = p_2x + q_2y + r_2 \quad (2.8)$$

Where  $p_i, q_i$  and  $r_i$  ( $i = 1$  or  $2$ ) are linear parameters in the then part of the first-order Sugeno model.

The architecture of ANFIS consists of five layers (Hakim & Razak, 2013). A short

explanation is given as follows.

**Layer 1:** Input nodes. It is also called fuzzification layer. Input signal is fed to the node  $i$ , which is associated with a linguistic label  $A_i$  or  $B_{i-2}$ . Each node generates membership grades. These nodes belong to the appropriate fuzzy sets by membership functions, as shown below;

$$O_{1,i} = \mu_{A_i}(x) \text{ for } i = 1, 2, \text{ or} \quad (2.9)$$

$$O_{1,i} = \mu_{B_{i-2}}(y) \text{ for } i = 3, 4 \quad (2.10)$$

Where  $x, y$  are the inputs to node  $I$ ;  $A_i, B_{i-2}$  are the linguistic labels;  $O_{1,i}$  is the membership grade of a fuzzy set  $(A_1, A_2, B_1, B_2)$ .

The bell-shaped membership function is chosen in this study. The function is given by Eq. (2.11) and Eq. (2.12).

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \text{ for } i = 1, 2, \text{ or} \quad (2.11)$$

$$\mu_{B_{i-2}}(y) = \frac{1}{1 + \left| \frac{y - c_i}{a_i} \right|^{2b_i}} \text{ for } i = 3, 4 \quad (2.12)$$

Where  $a_i, b_i$  and  $c_i$  are parameters of membership functions. These parameters can change the shapes of the membership functions.

**Layer 2:** Rule nodes. This layer known as membership layer and each node is fixed node labeled Prod. The output of each node represents the firing strength of a rule. The output values can be obtained by multiplying signal from layer 1 and then deliver to the next layer, as shown in Eq. (2.13). Each node represents the firing strength for each rule

in this layer.

$$O_{2i} = \omega_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y), \quad i = 1, 2 \quad (2.13)$$

**Layer 3:** Average nodes. Each node is fixed node labeled Norm in this layer. It calculates the ratio of *ith* rules firing strength to sum of all rule' firing strengths (Eq. (2.14)).

$$O_{3i} = \bar{\omega}_i = \frac{\omega_i}{\omega_1 + \omega_2}, \quad i = 1, 2 \quad (2.14)$$

Where  $O_{3,i}$  is outputs, called normalized firing strengths,  $\bar{\omega}$  is taken as normalized firing strength.

**Layer 4:** Consequent nodes. It is also called defuzzification layer. Each node is adaptive node in this layer. It calculates the contribution of each *ith* rule's toward the total output, as shown in Eq. (2.15). Individual nodes of this layer are connected to the respective normalization node in layer 3 and input signal.

$$O_{4,i} = \bar{\omega}_i f_i = \bar{\omega}_i (p_i x + q_i y + r_i), \quad i = 1, 2 \quad (2.15)$$

Where  $p_i, q_i$  and  $r_i$  are parameter set of the Sugeno fuzzy model in consequent layer.

**Layer 5:** Output nodes. Each node is fixed node and labeled as sum. It calculates the overall output by summing all the incoming signal from pervious layer (Eq. (2.16)).

$$O_{5,i} = \sum_{i=1}^4 \bar{\omega}_i f_i = \frac{\sum_{i=1}^4 \omega_i f_i}{\sum_{i=1}^4 \omega_i} \quad (2.16)$$

The genfis 2 function was used to generate a model from data using clustering. It is a fast, one-pass method that does not perform any iterative optimization. The only one

parameter that need optimizes is radius, which must be a real numeric value in the range [0 1]. When fuzzy systems are designed by using fuzzy clustering, each cluster corresponds to a fuzzy rule. Therefore, the number of clusters decides the number of rules. A smaller cluster radius will usually generate many small clusters in the data, which yields in many rules and the prediction results become more accurate (H. Md. et al., 2012).

$$\text{fismat} = \text{genfis2}(\text{in}, \text{out}, \text{radius})$$
$$\text{fis} = \text{anfis}([\text{in out}], \text{fismat}, [\text{MN GOAL}])$$

Where in is input, out is output, radius is predefined parameter, MN is the maximum hidden neuron, and GOAL is the error goal.

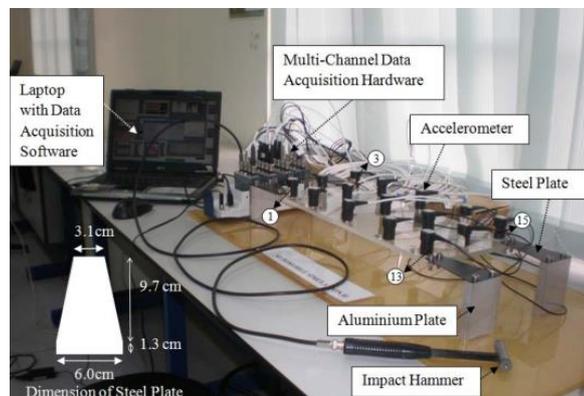
University of Malaya

## CHAPTER 3: RESEARCH METHODOLOGY

### 3.1 Equipment and experimental set up

#### 3.1.1 Test rig

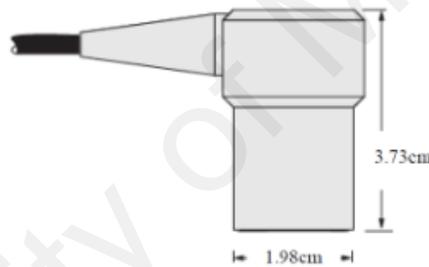
The test rig is constructed with rectangular Perspex plate, where the dimensions are 48cm for length, 20cm for width and 0.9cm for thickness. It was built by previous researcher (Khoo et al., 2014), as showed in Fig. 3.1. The weight of this plate is 1.10kg and it is ground supported at its four corners. These supports are made of aluminum and steel plate. The dimensions of aluminum are 6.4cm for length, 1.3cm for width and 8.9cm for thickness. When the vehicle body strikes structure, the force generated is very complicated with infinite Degree of Freedom (DOF). But it can be simplified into few DOF in a plain structure (Weaver Jr et al., 1990). Vertical plane is considered in this study, which includes translational motion and rotational motion about the mass centroid of the structure. The simulated structure can generate similar dynamic vibration behavior with actual vehicle. It is found that the previous studies used 4-9 sensors for most of the cases (Haywood et al., 2004; LeClerc et al., 2007; Worden & Staszewski, 2000). 6 accelerometers were chosen for this study and they were mounted in a symmetric order in the center of the structure.



**Fig. 3.1:** Experimental Set-up for impact force identification (Khoo et al., 2014)

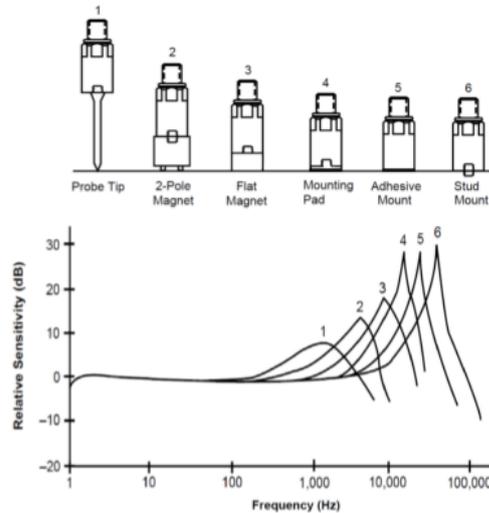
### 3.1.2 Accelerometer

Accelerometer is one of the most commonly used sensors for vibration analysis. S100C Wilcoxon Research Integrated Circuit Piezoelectric (ICP) accelerometer is used as the response sensor in this study. It has a built-in charge amplifier. When sensor body vibrates, stressed piezoelectric quartz releases proportional charge to acceleration. The internal circuit converts charge to low impedance voltage and then provides output through sensor's casing. It is capable to measure the temperature from  $-50^{\circ}\text{C}$  to  $80^{\circ}\text{C}$ . It can detect a wide range frequency of signal from 0.5-10000.0 Hz. The sensitivity of this kind of accelerometer is 100mV/g and it can measure the acceleration up to  $784.8\text{ms}^{-2}$ . The dimension of accelerometer is 3.73cm in height and 1.98cm in diameter (Fig. 3.2).



**Fig. 3.2:** S100C ICP accelerometer (Khoo et al., 2014)

There are several mounting methods for accelerometer, such as probe tip, 2-pole magnet, flat magnet, mounting pad and so on. Appropriate mounting method should be chosen for different types of studies. It can affect the sensitivity and performance of sensor, especially in high frequency regions. Cyanoacrylate adhesive mounting method is chosen in this study to avoid any phase lag based on the flat surface. It can generate good performance and also is very convenient to install compare with other method.



**Fig. 3.3:** Mounting methods and their effects on accelerometer's sensitivity (Khoo et al., 2014)

### 3.1.3 Impact hammer

An impact hammer is used to generate impulse in the structure. When it strikes the test structure, a nearly constant impulse can be obtained over a very short time. It is capable to excite all the natural frequencies of the test structure. The amplitude and frequency content of impact force can be affected by impact velocity, hammer size, contact's material and hammer length.

In this study, a PCB ICP® impact hammer model 086C03 is used as the impact tool with weight 0.16g. The sensitivity is 2.09mV/N and it can generate impact force peak with a wide range  $\pm 2200$ N. The tip is covered with vinyl material where frequency range can up to 2.5 kHz. Quartz force sensor is mounted at the head of impact hammer. The impact force information can be sensed by built-in sensing elements of hammer. Mechanical signal can be converted into electric signal and finally transferred into computer to analyze.



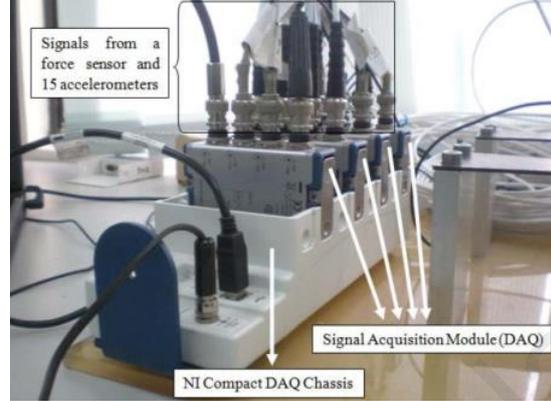
**Fig. 3.4:** Impact hammer (Khoo et al., 2014)

### **3.1.4 Data acquisition system (DAQ)**

Data acquisition (DAQ) hardware is also known as analogue digital convertor (ADC), which converts analogue signal comes from accelerometer to digital signal. Then the digitized signal can be sent to computer by DAQ software. It works as an interface that connects the outside with computer. National Instrument-Universal Serial Bus (NI-USB) dynamic signal acquisition module, model NI-USB 9233 was used as DAQ hardware in this study. It comprises 4 input channels which can receive simultaneous signal from accelerometer. The data filtering can adjust the frequency part of analyzing signal by building cut-off frequency. The range of sampling rate of NI-USB 9233 is 2-50kS/s. The dynamic range of this kind of DAQ module is over 100dB and the voltage range is  $\pm 5V$ .

Four NI-USB 9233 are used to acquire 6 accelerometer responses and 1 force of impact hammer. In order to make sure all the signals are obtained with simultaneous way, a DAQ chassis, model NI cDAQ-9172 is connected before all the signals enter into 4 NI-USB 9233, which requires 11-30V power supply. After obtaining the raw data from DAQ hardware system, DASyLab<sup>®</sup> is a software that can be used to post-process the collected

data, then analyzed by MATLAB<sup>®</sup>2017b. The precision of impacts is 2 radial cm around the impact location center. The sampling rate is fixed at 2000Hz and block size includes 4096 samples.



**Fig. 3.5:** Data acquisition module (Khoo et al., 2014)

## 3.2 Impact identification methodology

### 3.2.1 Time-domain feature extraction

After acquiring all the responses data caused by impact force, MATLAB<sup>®</sup>2017b was used to extract all the feature by coding. Based on the previous study (Md Sazzad et al., 2017), MAT was proven as the best input feature for localization that can achieve the lowest error among many features, such as PAT, TC, RMS and peak-to-peak and so on. In addition, peak-to-peak was proven as the input feature for impact force quantification by several researchers (Haywood et al., 2004; Maseras-Gutierrez et al., 1998; Worden & Staszewski, 2000). In this study, peak-to-peak plus MAT were used as the input feature for impact quantification, which improved accuracy of only using peak-to-peak feature.

The signal equation can be expressed as follows:

$$F_{MAT} = t_{\min(v(t))} - t_0 \quad (3.1)$$

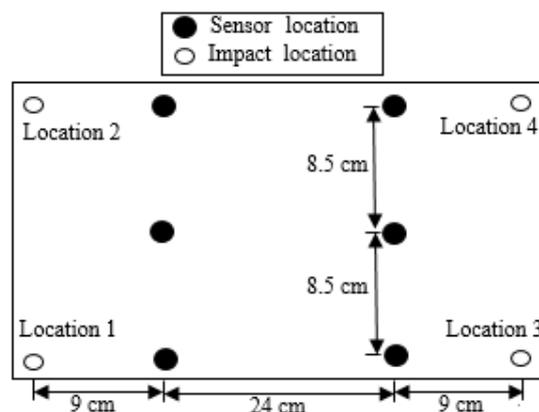
$$F_{peak-to-peak} = \max(|v(t)|) - \min(|v(t)|) \quad (3.2)$$

Where  $v(t)$  is time-domain acceleration signal;  $F$  is feature method and  $n$  is size of the

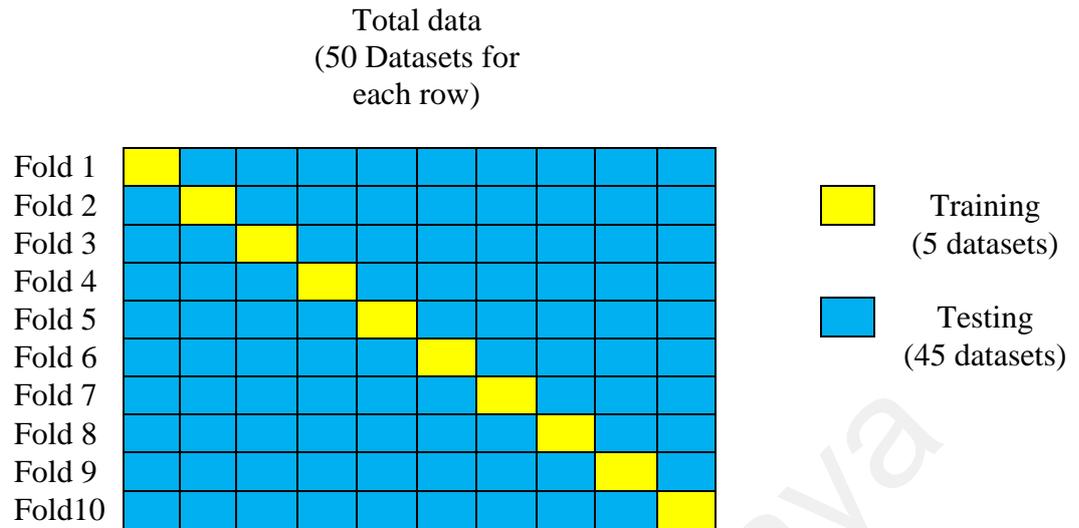
response vector.

### 3.2.2 Arrangements of sensors and trial impacts

Four impact locations were chosen in the corner of the test rig, which represent the actual wheel of vehicle. Sensors were fixed in symmetric arrangement from the center of the test rig. Totally 50 trial datasets were obtained where each dataset includes 6 sensor responses in four impact locations. In order to validate the architecture of the network, 10-fold cross validation was applied to this study. 10 randomly subsamples could be selected by 10-fold cross validation, which reduces the influence of manually selection of training and testing data. 5 randomly chosen datasets were used for training process. The rest 45 randomly chosen datasets were used for testing. Each network would be trained 10 times by using different optimized parameters for 10 subsamples. Training error goal was set to 0.1 for all the networks, avoiding zero error that can cause overfitting. Then testing process was proceeded to observe their overall performances. For localization, testing error was obtained by Euclidean distance between corresponding predicted and measured location. For quantification, testing error was evaluated in terms of relative error  $(|((\text{actual magnitude} - \text{predicted magnitude})| / \text{actual magnitude}) \times 100)$ .



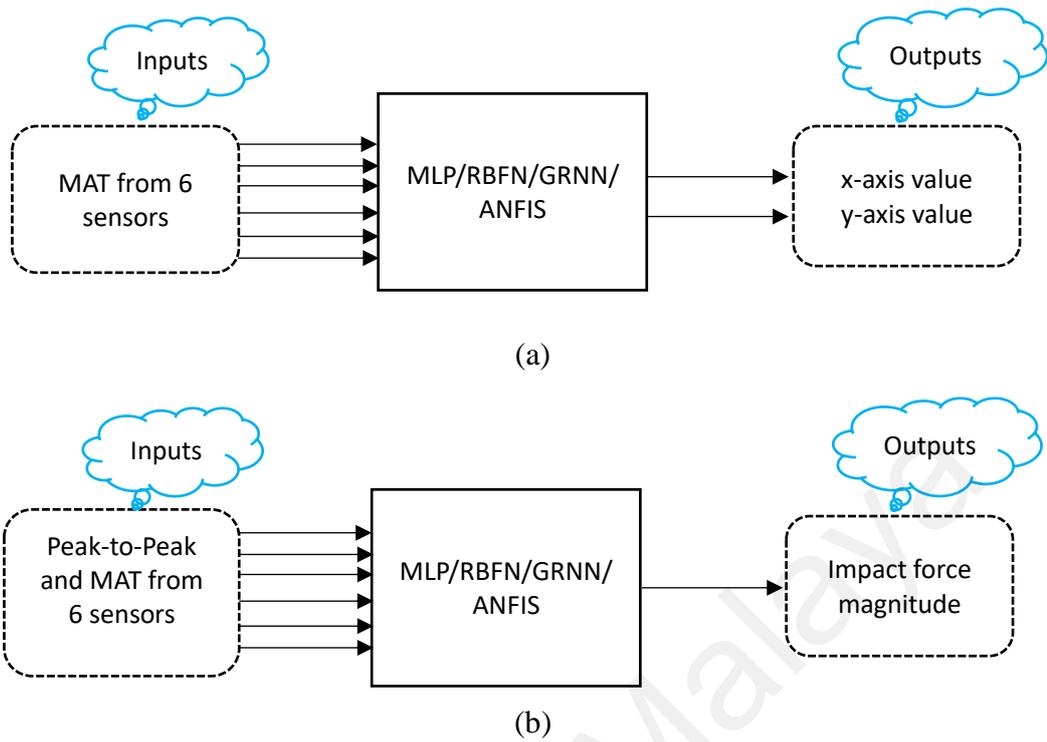
**Fig. 3.6:** Arrangement of the sensors and impact locations



**Fig. 3.7:** Structure of 10-fold cross validation

### 3.2.3 Neural network schemes

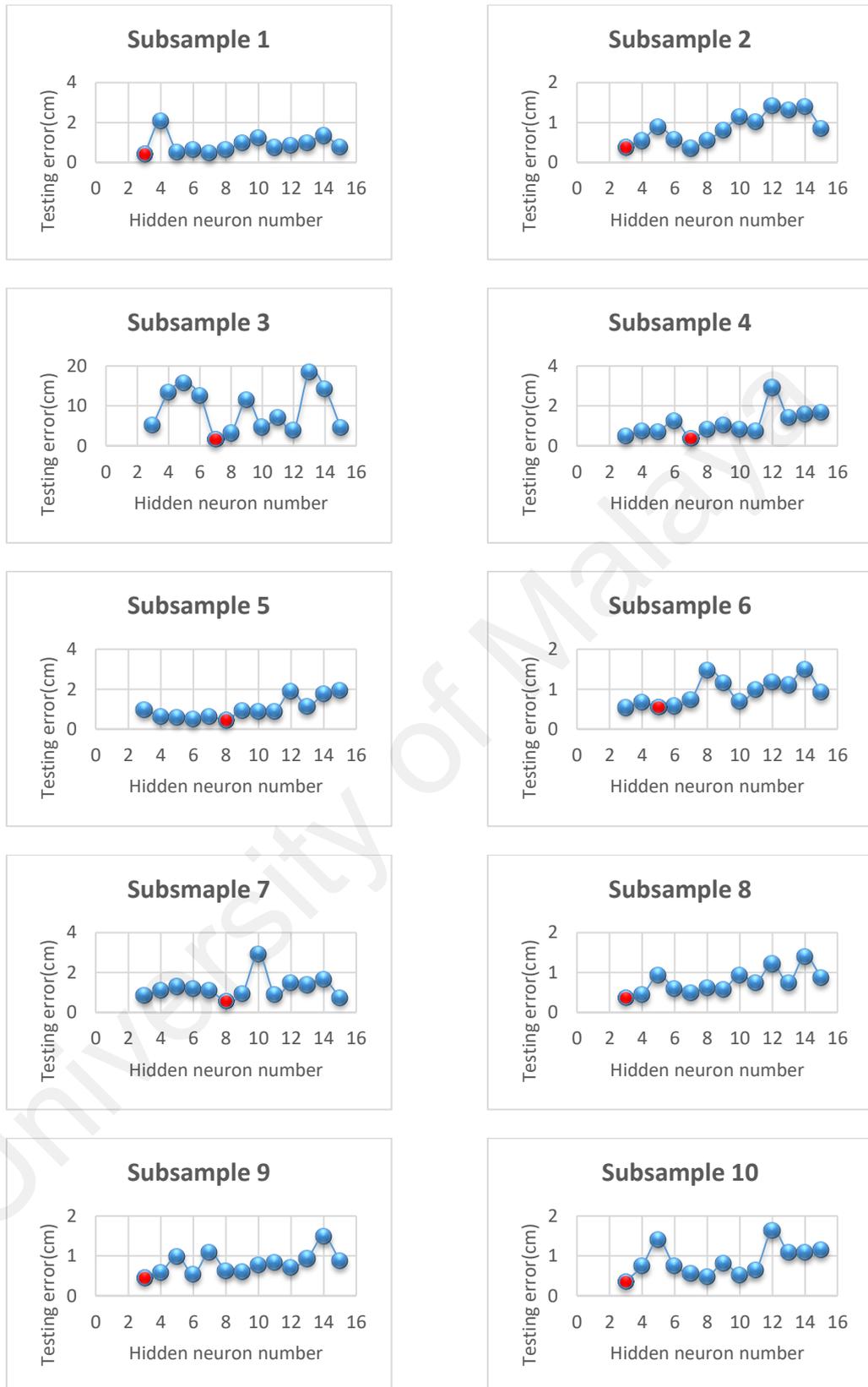
Simple schemes of neural network are shown in this section for localization and quantification of impact, respectively. These two figures clearly show inputs and outputs for both cases of localization and quantification. For MLP, RBFN and GRNN, totally 5 randomly chosen training datasets were arranged with 6 by 20 matrixes. Each training dataset included six sensor responses in four different locations, which was arranged for training process with 6 by 4 matrixes. The outputs for localization and quantification were represented by Cartesian coordinates (x and y axis) and impact force magnitudes, respectively. The arrangement of training datasets for ANFIS was slightly different with other three networks, which arranged all the datasets in one column and each dataset involved six sensor responses in four locations. In case of localization, two models were applied to predict x and y axis value, respectively, then MSE was used to calculate the total error in corresponding locations.



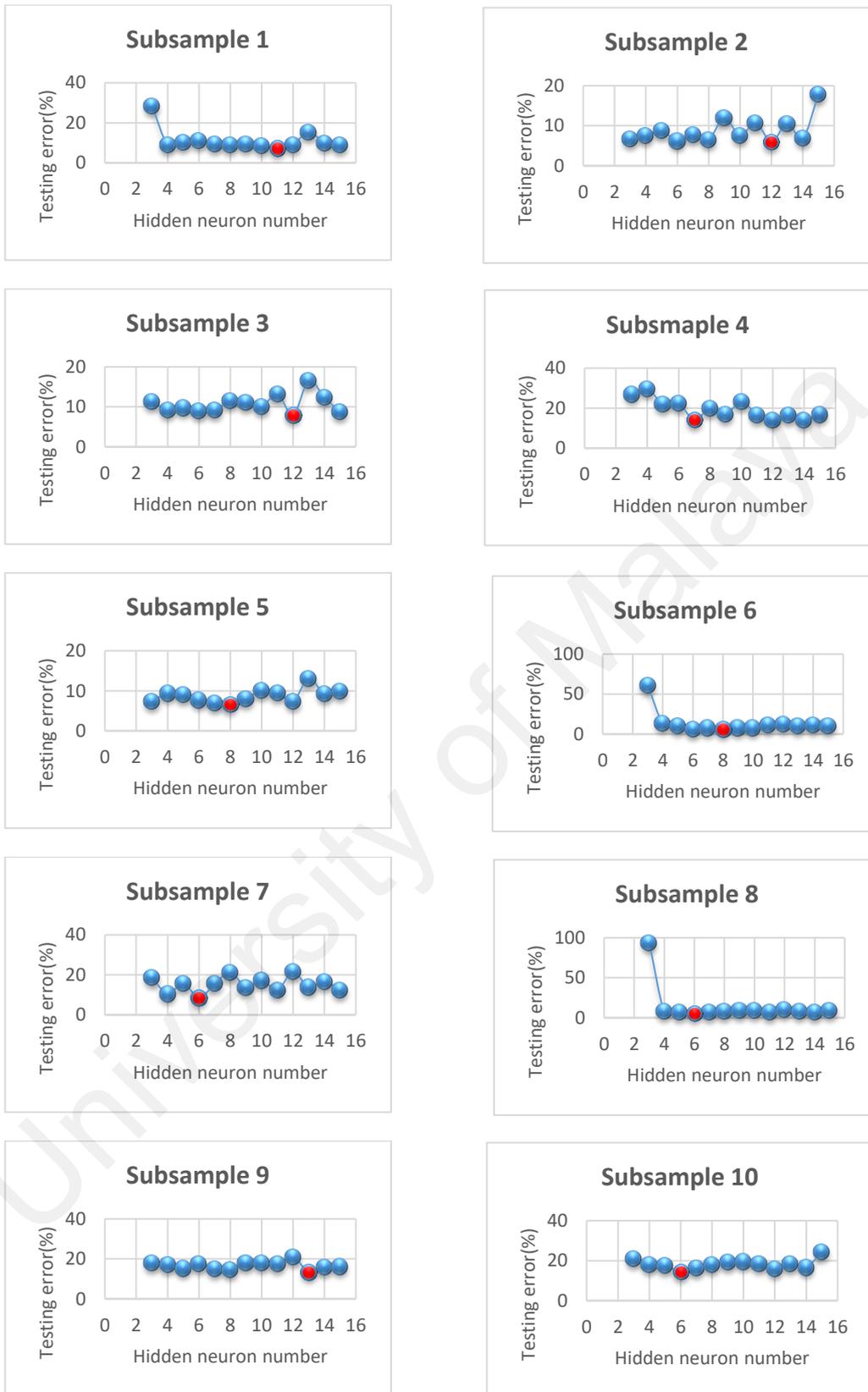
**Fig. 3.8:** Impact (a) localization and (b) quantification schemes using various neural networks

### 3.2.4 Optimize network parameters

For MLP network, hidden neuron number need be optimized for each subsample. Various numbers were chosen to train where testing error was recorded. Fig. 3.9 and Fig. 3.10 show testing error with respect to different hidden neuron numbers for localization and quantification. Red color point presents the minimum testing error at corresponding hidden neuron number, which would be taken as optimized parameters. Relevant data was obtained according to Appendix Table A.1 and Table A.2).

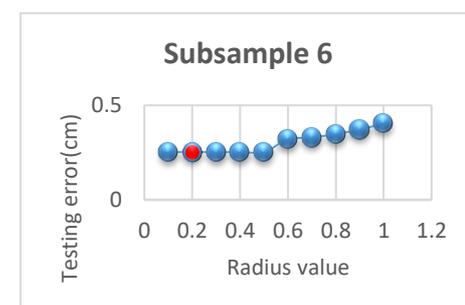
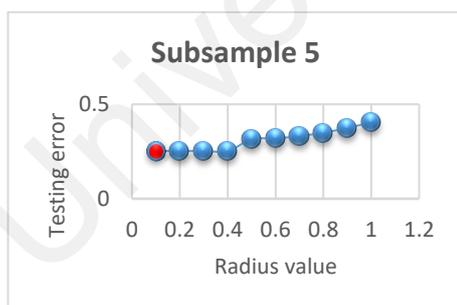
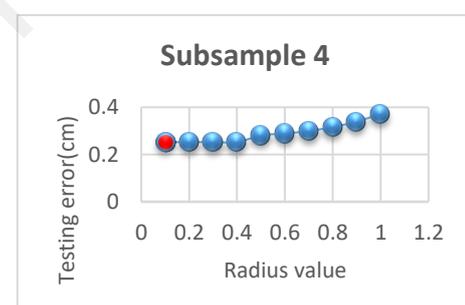
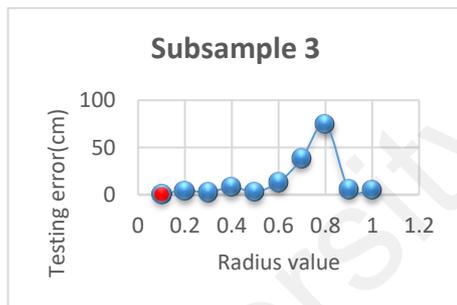
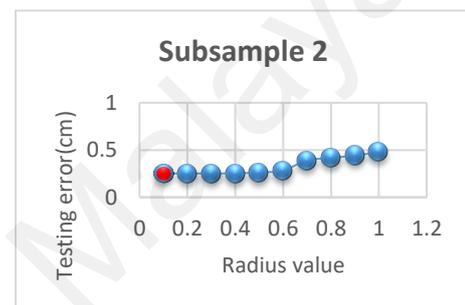
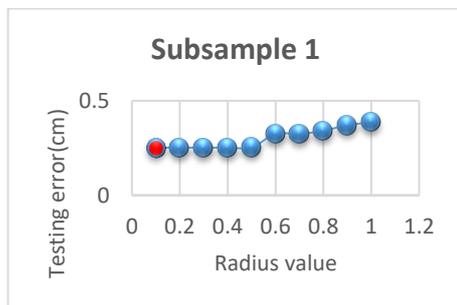


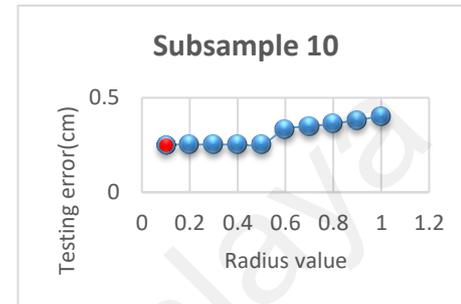
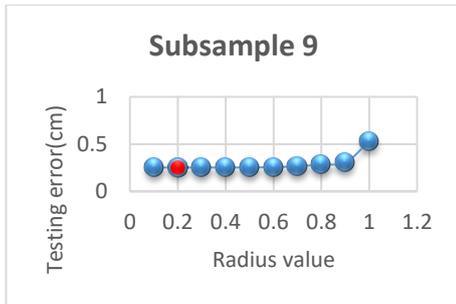
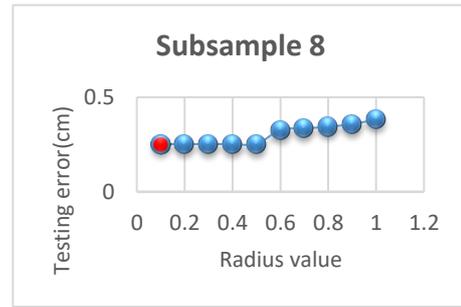
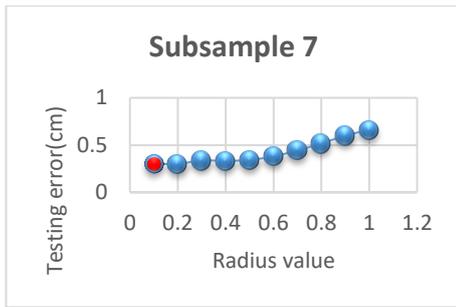
**Fig. 3.9:** Optimize hidden neuron number (MLP) for impact force localization



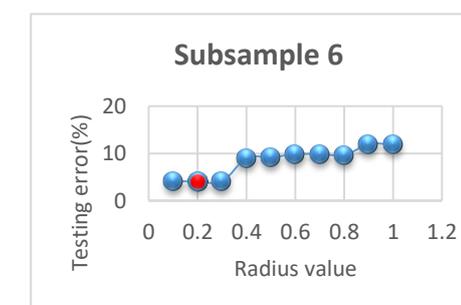
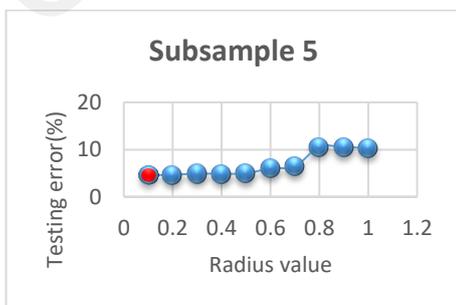
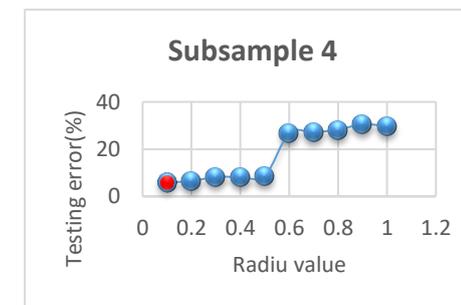
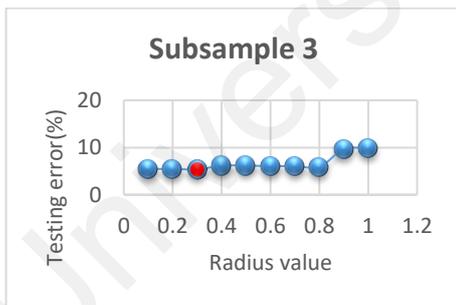
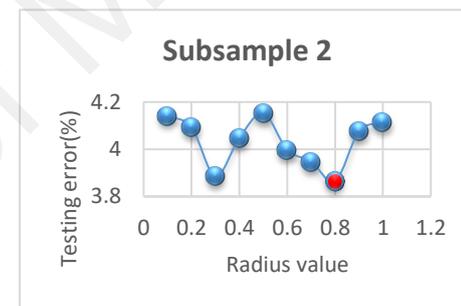
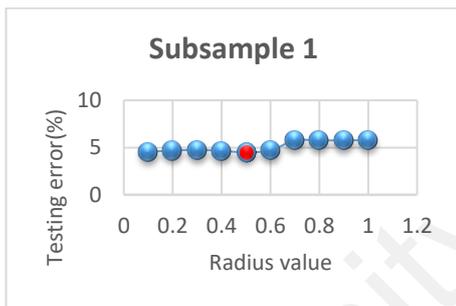
**Fig. 3.10:** Optimize hidden neuron number (MLP) for impact force quantification

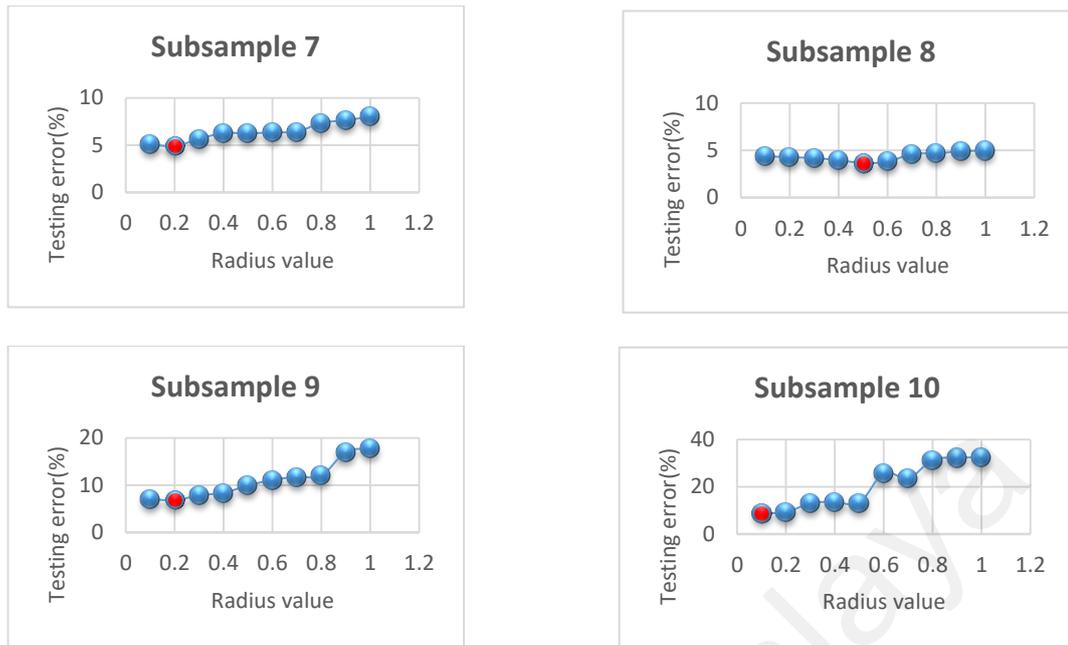
For ANFIS network, radius need be optimized for each subsample. The range of radius value is examined from 0.1 to 1. Radius was initially set at 0.1, then the radius was increased with the interval of 0.1. Each subsample should be trained 10 times. The results were plotted in Fig. 3.10 and Fig. 3.11 for impact localization and quantification, respectively. Detailed data for Figs. 3.10 and 3.11 can be found in Appendix Table A.3 and Table A.4. The red color point shows best radius value for each subsample, which would be taken as optimized parameters for ANFIS.





**Fig. 3.11:** Optimize radius (ANFIS) for impact force localization





**Fig. 3.12:** Optimize radius (ANFIS) for impact force quantification

For RBFN and GRNN network, the optimized spread constant for localization and quantification can be found in Appendix A (Table A.5, Table A.6, Table A.7 and Table A.8), which was marked by red color for each subsample.

In overall, the optimized parameters for each network are presented in Table 3.1 and Table 3.2.

**Table 3.1:** Optimized parameters of different network for impact force localization

Subsamples	1	2	3	4	5	6	7	8	9	10
Hidden neuron no. (MLP)	3	3	7	7	8	3	8	3	3	3
Spread value (RBFN)	0.0039	0.0066	0.0029	0.0053	0.0040	0.0057	0.0063	0.0059	0.0062	0.0042
Spread value (GRNN)	0.002	0.0001	0.0001	0.0001	0.0001	0.0022	0.0009	0.0001	0.0001	0.002
Radius value (ANFIS)	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0.2	0.1

**Table 3.2:** Optimized parameters of different network for impact force quantification

Subsamples	1	2	3	4	5	6	7	8	9	10
Hidden neuron no. (MLP)	11	12	12	14	8	8	6	6	8	6
Spread value (RBFN)	305	320	115	155	165	325	70	425	35	130
Spread value (GRNN)	7	11	19	9	13	17	4	23	8	48
Radius value (ANFIS)	0.5	0.8	0.3	0.1	0.1	0.2	0.2	0.5	0.2	0.1

University of Malaya

## CHAPTER 4: RESULTS AND DISCUSSION

### 4.1 Results

In this section, training results of all examined neural networks for impact localization and quantification, respectively will be presented.

#### 4.1.1 Results for impact localization

Using optimized parameters for each network, training process was conducted by MATLAB. Testing error for subsequent subsamples and average test error can be found in Table 4.1. It shows that ANFIS has the minimum average testing error (0.3084 radial cm) and GRNN has close good performance with ANFIS. RBFN gives worst performance and its result is quite far away from ANFIS. Success rate within 2 radial cm error for each network can be obtained in Table 4.2. It is clear to see that GRNN has best success rate (99.16%) compared with others. Although ANFIS has outstanding performance in average testing error, success rate is not good as GRNN (98%). Moreover, only GRNN shows very high identification accuracy for subsample 3 according to table 4.1. It indicates that this kind of network has very robust ability to tolerate noisy data. A detailed comparison will be discussed in next section.

**Table 4.1:** Impact localization testing errors of different subsamples for RBFN, GRNN, ANFIS and MLP

Types	1	2	3	4	5	6	7	8	9	10	Average
RBFN	0.5972	0.7455	2.3156	0.8844	0.8233	0.8143	1.1139	0.7389	0.6387	0.7739	0.9446
GRNN	0.3810	0.2517	0.2517	0.2517	0.2517	0.4264	0.4683	0.2517	0.2517	0.4153	0.3201
ANFIS	0.2522	0.2524	0.7653	0.2523	0.2523	0.2524	0.3007	0.2522	0.2520	0.2522	0.3084
MLP	0.4241	0.3840	1.7172	0.3773	0.4576	0.5409	0.5806	0.3794	0.4590	0.3644	0.5685

Note: Orange means minimum testing error, blue means maximum error.

**Table 4.2:** Success rate percentage of impact localization for different subsamples

Types	Success rate(%) within 2 radial cm error for subsequent subsample										Average
	1	2	3	4	5	6	7	8	9	10	
RBFN	97.78	98.33	81.67	95	96.11	98.89	91.11	97.78	98.89	95.56	95.11
GRNN	98.89	99.44	99.44	99.44	99.44	98.89	98.33	99.44	99.44	98.89	99.16
ANFIS	99.44	99.44	85.56	99.44	99.44	100.00	98.89	99.44	99.44	98.89	98.00
MLP	98.89	99.44	88.33	99.44	98.89	98.89	98.89	99.44	99.44	98.89	98.05

Note: Orange means maximum success rate, blue minimum success rate.

#### 4.1.2 Results for impact quantification

Using the optimized parameters which have been done in last chapter, training results were obtained for each neural network as follow.

**Table 4.3:** Impact quantification testing errors of different subsamples for RBFN, GRNN, ANFIS and MLP

Types	1	2	3	4	5	6	7	8	9	10	Average
RBFN	4.7723	3.8661	6.354	16.7829	4.8208	4.7082	6.6939	3.9010	14.2716	15.7364	8.1907
GRNN	11.3304	8.2761	8.3896	11.8164	8.1659	9.5150	8.2105	7.2954	7.3967	7.5280	8.7924
ANFIS	4.4898	3.8649	5.3915	5.9812	4.6095	4.0320	4.8894	3.6542	6.7995	8.8230	5.2535
MLP	7.3145	5.9628	7.9485	14.0975	6.5865	6.0974	8.4427	5.5661	14.6063	14.5032	9.1126

Note: Orange means minimum testing error, blue means maximum error.

**Table 4.4:** Success rate percentage of impact quantification for different subsamples

Types	Error percentage (%) within 10% for subsequent subsample										Average
	1	2	3	4	5	6	7	8	9	10	
RBFN	95.56	96.11	93.33	66.11	94.44	93.89	90	96.11	65.56	61.11	85.22
GRNN	80.00	85.56	86.11	75.56	85.00	78.33	83.89	86.67	85.56	91.11	83.78
ANFIS	95.00	96.11	94.44	87.78	95.56	95.56	95.56	96.11	85.00	80.00	92.11
MLP	83.33	92.22	84.44	73.33	91.67	91.67	84.44	94.44	67.22	70.56	83.33

Note: Orange means maximum success rate, blue minimum success rate.

From table 4.3 and 4.4, it is clear to show that ANFIS has outstanding performance among these four networks. The average testing error for GRNN (8.7924 %) is slightly higher than RBFN (8.1907 %). In addition, average success rate of RBFN (85.22%) is higher than GRNN (83.78%), but it has maximum testing error (16.7829%) among all networks for each testing subsample. MLP has highest testing error (9.126 %) and lowest

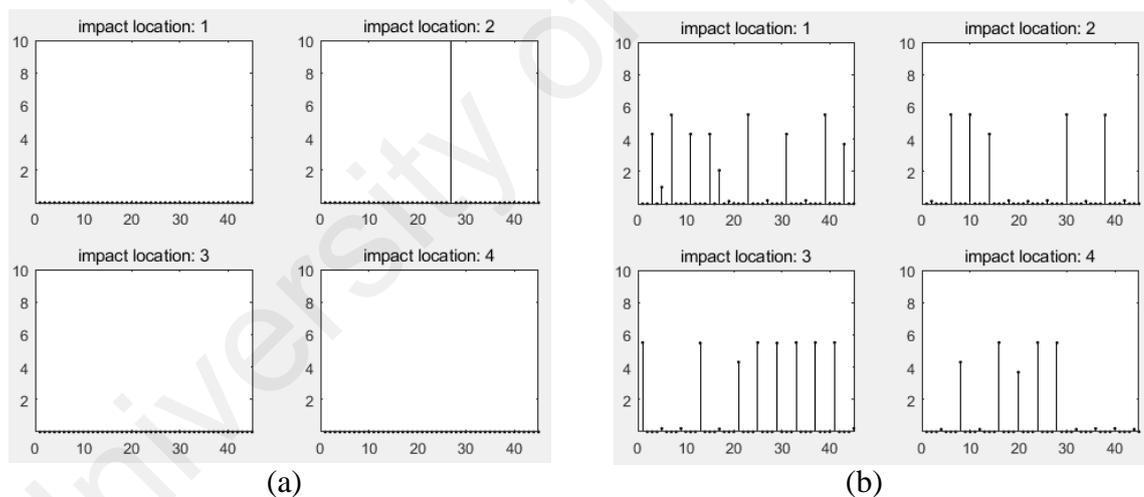
success rate percentage (83.33%) within 10% of error. In order to further study the performance of each network, a detailed analysis has been conducted in next section.

## 4.2 Discussion

In this section, various results for each neural network will be discussed and best neural network will be chosen based on the discussion.

### 4.2.1 Selection of best neural network for impact localization

Subsamples that have minimum and maximum testing error were chosen to analyze the detailed performance. Firstly, subsample 9 and subsample 3 were selected to analyze for ANFIS network, which have best and worst performance, separately. The training results are showed in Fig. 4.1.



**Fig. 4.1:** (a) Subsample 9, which has best identification performance among all the subsamples for ANFIS; (b) Subsample 3, which has worst identification performance among all the subsamples for ANFIS.

Fig. 4.1 shows that the average error range for ANFIS is 0.2520-0.7653 radial cm. Subsample 9 shows good performance and all the errors are less than 0.002 radial cm except testing dataset 27 in location 2, with 45.31 radial cm testing error. Below table

presents the responses of sensor at each location are totally different between training input and testing input in impact #27, thus it gets high identification error. The range of success rate within 2 radial cm error for ANFIS is 85.56%-100%. If sensor gives good response, ANFIS is robust to identify location of impact force.

**Table 4.5:** Comparison of testing impact data #27 with training data at location 2

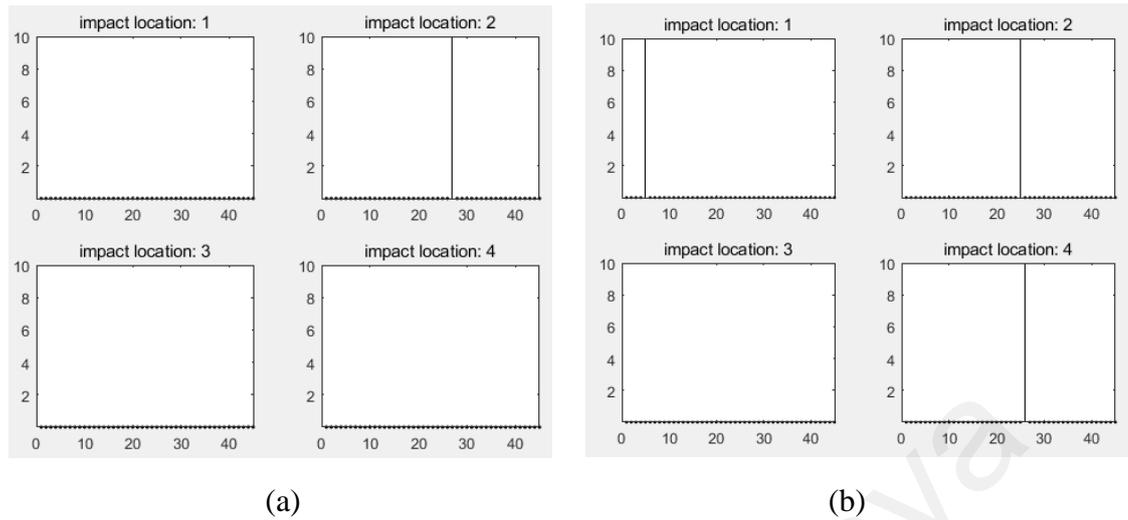
Sensor no.	No. of training trials at location 2					Testing impact #27
	1	2	3	4	5	
1	0.0285	0.0285	0.0280	0.0285	0.0285	0.0310
2	0.0265	0.0265	0.0270	0.0270	0.0265	0.0310
3	0.0255	0.0255	0.0260	0.0260	0.0255	0.0295
4	0.0355	0.0355	0.0355	0.0355	0.0355	0.0285
5	0.0300	0.0300	0.0305	0.0295	0.0300	0.0270
6	0.0290	0.0285	0.0290	0.0290	0.0290	0.0270

In addition, it uses hybrid algorithm by mixing with least mean squares and backpropagation approach, which has advantages compared with other conventional neural networks. It converges much quicker because it decreases the search space dimensions of the original backpropagation method that was used in MLP. For example, subsample 1 was randomly chosen to compare the iterations for each neural network. Table 4.6 shows that ANFIS has much better learning ability, where much smaller convergence error can be achieved using only one iteration.

**Table 4.6:** Iterations and training error to achieve error goal for different networks

Types	ANFIS	GRNN	RBFN	MLP
Iterations	1	1	7	5
Training error	0.00015	0.05675	0.05675	0.01090

Secondly, subsample 2 and subsample 7 were selected to analyze their performances for GRNN, which are the best subsample and worst subsample, respectively.



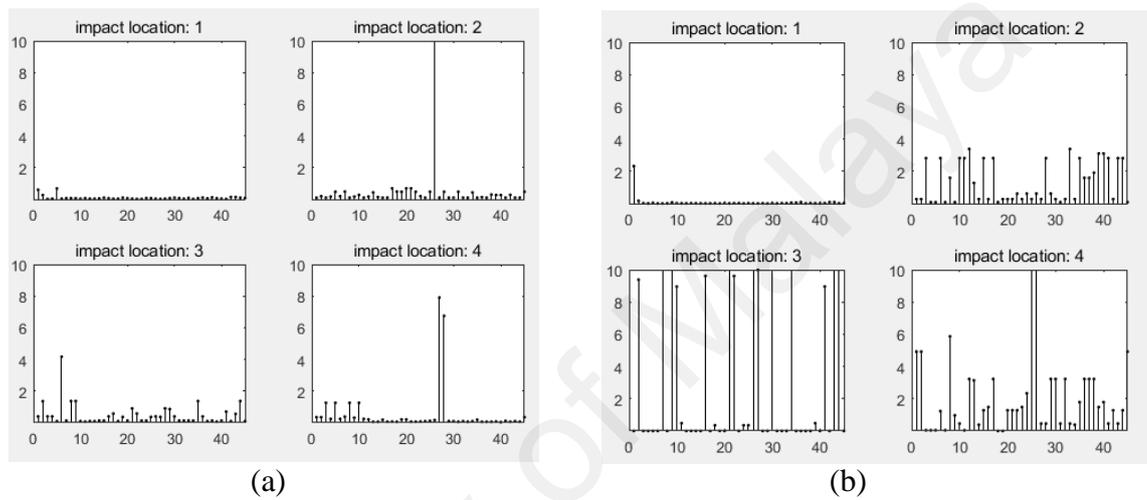
**Fig. 4.2:** (a) Subsample 2, which has best identification performance among all the subsamples for GRNN; (b) Subsample 7, which has worst identification performance among all the subsamples for GRNN.

The error range is 0.2517-0.4683 radial cm for GRNN, which is very precise to identify location. In addition, it has the least testing error among these four networks. Although the average error of GRNN is more than ANFIS, it shows higher average success rate (99.16%) and higher accuracy of success rate (98.33%-99.44%). In addition, the worst performance of GRNN is far away better than ANFIS, where subsample 7 shows average testing error is 0.4683 radial cm for GRNN, while ANFIS shows 0.7653 radial cm testing error in subsample 3. Therefore, GRNN is the best model to identify location among these networks. It has similar architecture with RBFN. They both use Gaussian function in the second layer. By adding another special linear layer in GRNN, the accuracy of identification can be improved. In addition, the output is converged to global and won't be trapped by a local minimum like MLP.

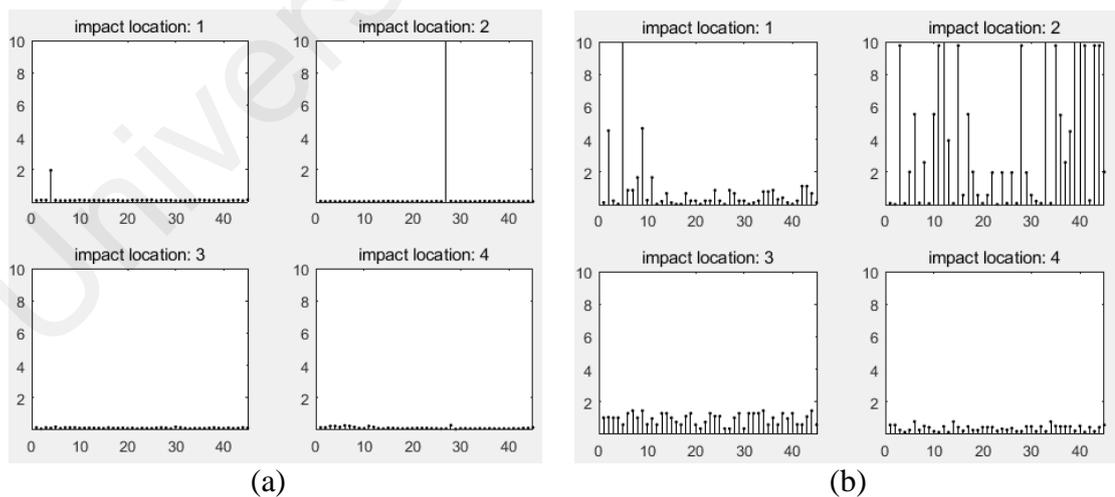
Although GRNN has the best performance, there are three datasets show very poor

results for subsample 7, with 45.31 radial cm, 23.23 radial cm and 15.76 radial cm error, separately. In contrast, the worst performance of subsample 3 of ANFIS is better than GRNN, with maximum error less than 6 radial cm.

Thirdly, subsamples that have best and worst performance were chosen for RBFN and MLP, respectively. Testing results can be presented as follows.



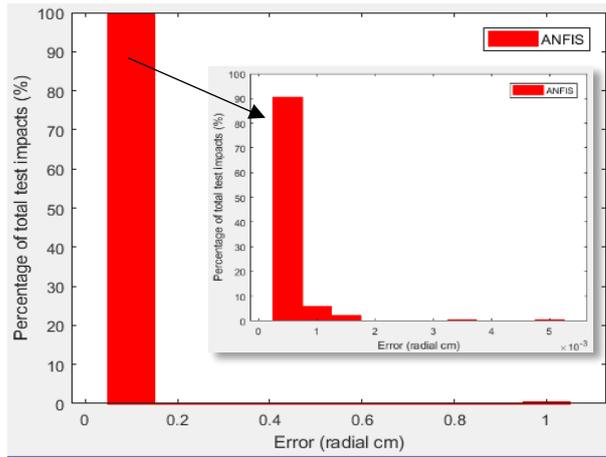
**Fig. 4.3:** (a) Subsample 1, which has best identification performance among all the subsamples for RBFN; (b) Subsample 3, which has worst identification performance among all the subsamples for RBFN.



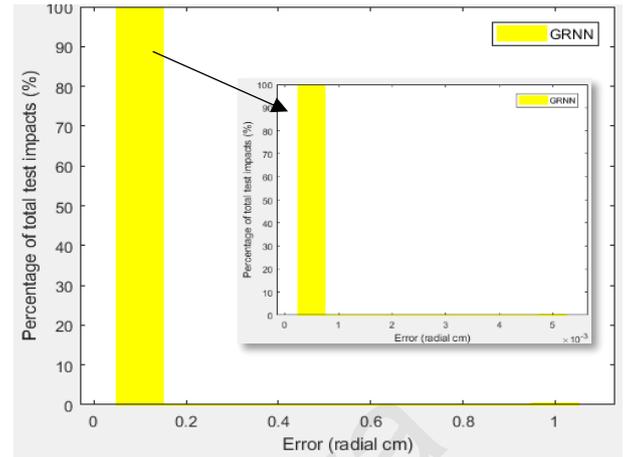
**Fig. 4.4:** (a) Subsample 10, which has best identification performance among all the subsamples for MLP; (b) Subsample 3, which has worst identification performance among all the subsamples for MLP.

From the above figures, it is obvious that RBFN performs worst among these networks. It has broad error range (0.5972-2.3156 radial cm) and success rate of error within 2 radial cm (81.67%-98.89%). Based on the previous study, the approximation ability of RBFN is better than MLP by using feature peak arrival time. Minimum arrival time is proven better than peak arrival time combined MLP network by previous researchers (Md Sazzad et al., 2017). In this study, minimum arrival time is used as input feature. It is found that the localization of MLP is more accurate than RBFN. It is because that MLP can be applied as stochastic optimizer and it only calculates random probability distribution. When it combines high accuracy feature, good performance can be shown, while RBFN is more robust to error, which interpolates between input vectors and it needs consider all the datasets. For subsample 3, RBFN shows good results in location 1 and 2, while MLP shows good results in location 3 and 4. It is because that RBFN is more error-tolerant with less accurate input data.

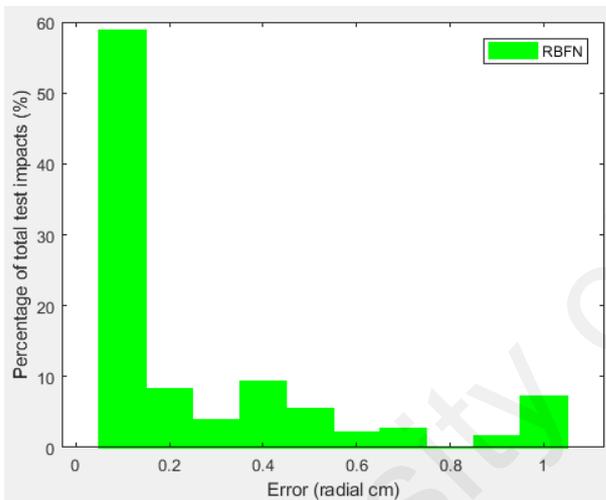
Subsample that has best performance for each network was chosen to compare their prediction accuracy (Fig. 4.5).



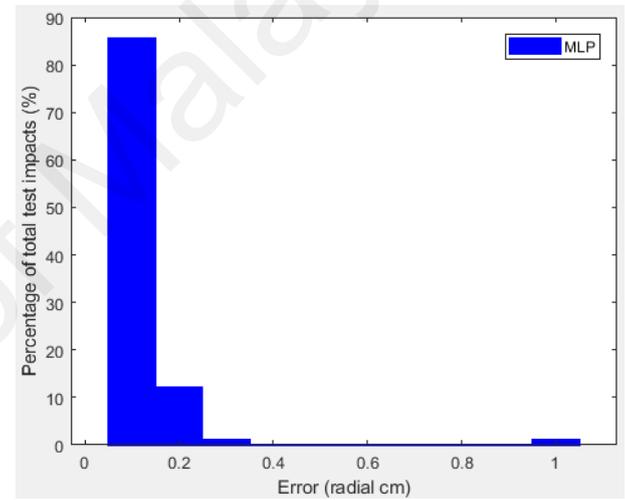
(a)



(b)



(c)



(d)

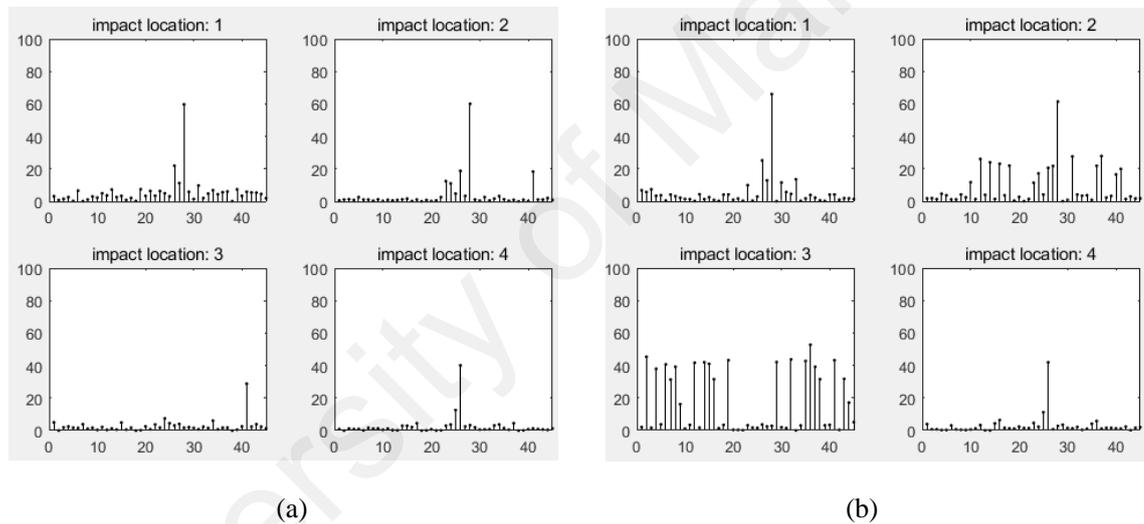
**Fig. 4.5:** (a) Radial error percentage within 0.1 cm for ANFIS (subsample 9); (b) Radial error percentage within 0.1 cm for GRNN (subsample 2); (c) Radial error percentage within 0.1 cm for RBFN (subsample 1); (d) Radial error percentage within 0.1 cm for MLP (subsample 10).

For GRNN and ANFIS network, 99.44% of data has error within 0.1 radial cm. They both show one dataset has high error up to 45.31 radial cm. Then smaller testing error is studied, it is found that 90.56% of data has error within 0.0005 radial cm for ANFIS network, while 99.44% of data has same error range for GRNN. For RBFN and MLP

network, no error is less than 0.0005 radial cm. The error range is wider and 85.56% of data has error within 0.1 radial cm for MLP. RBFN is the worst model and only 62.78% of data has error within 0.1 radial cm. Therefore, the ranking for these four networks in impact localization is GRNN > ANFIS > MLP > RBFN.

#### 4.2.2 Selection of best neural network for impact quantification

According to the training results, subsample 8 and subsample 10 have the best and worst performance for ANFIS network, respectively. These two subsamples were chosen to analyze and training results can be found as follow.

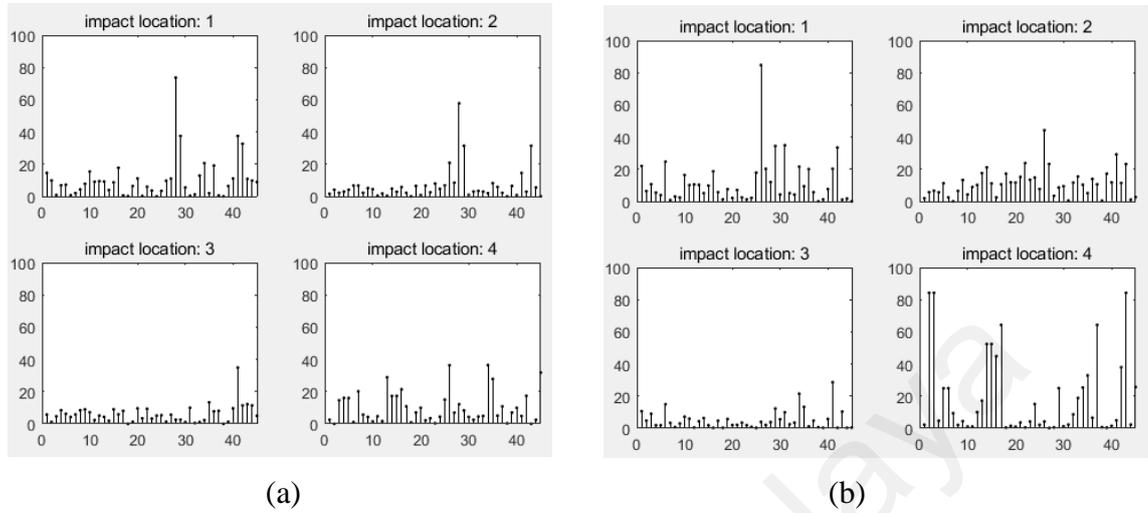


**Fig. 4.6:** (a) Subsample 8, which has best identification performance among all the subsamples for ANFIS; (b) Subsample 10, which has worst identification performance among all the subsamples for ANFIS.

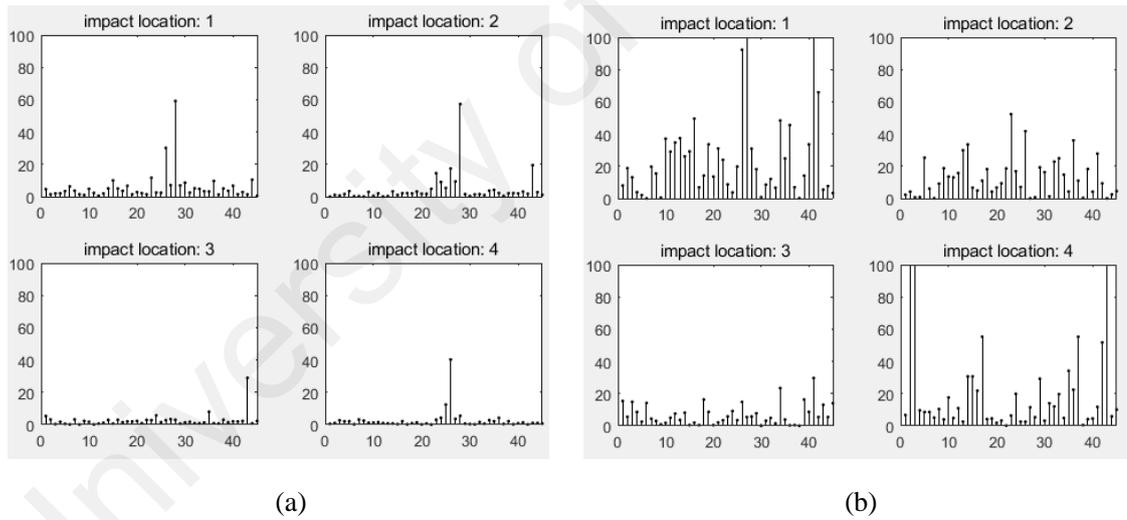
The percentage of error range for ANFIS is from 3.87% to 8.82%. For subsample 8, only one testing dataset has very high error percentage, up to 59.72% and 60.03% in location 1 and 2, respectively. Other datasets have very low error value.

Secondly, subsample 8 and 4 were chosen to study for GRNN and subsample 2 and 4

were chosen to study for RBFN, respectively. The testing results can be obtained as follow.



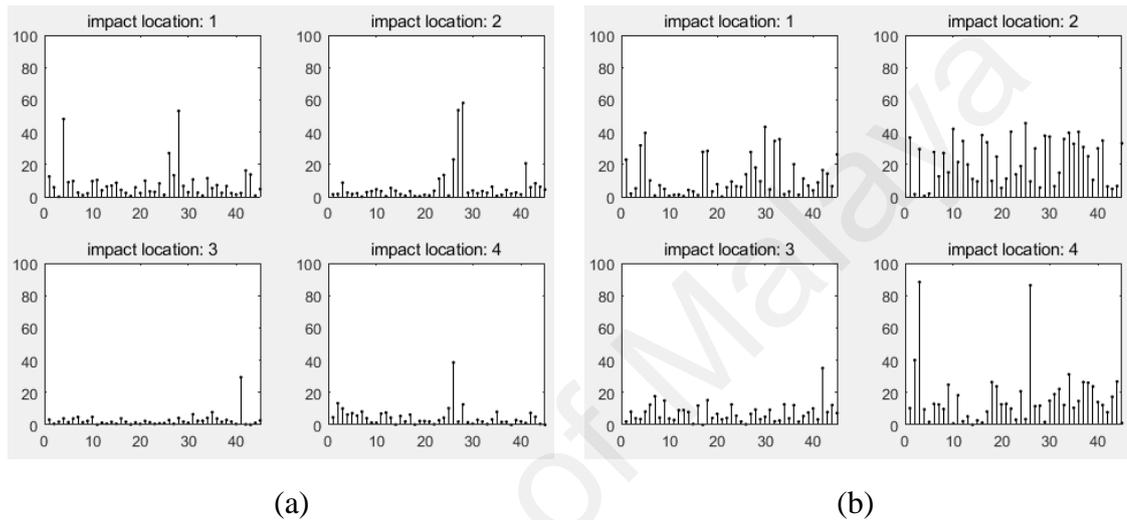
**Fig. 4.7:** (a) Subsample 8, which has the best identification performance among all the subsamples for GRNN; (b) Subsample 4, which has the worst identification performance among all the subsamples for GRNN.



**Fig. 4.8:** (a) Subsample 2, which has best identification performance among all the subsamples for RBFN; (b) Subsample 4, which has worst identification performance among all the subsamples for RBFN.

For RBFN network, the average and minimum testing errors of different subsample are less than GRNN, but the quantification error range (3.87%-16.78%) is wider than GRNN (7.30%-11.82%). RBFN can achieve high precision for magnitude identification, but it is

unexpected that the maximum error percentage is more than 11.82%, which is the maximum error value for GRNN. For GRNN network, the maximum error percentage is 84.72%, while it accounts for 135.10% for RBFN by training subsample 3. Due to the unstable environment, GRNN has priority for application over RBFN, which has higher probability to achieve good results.

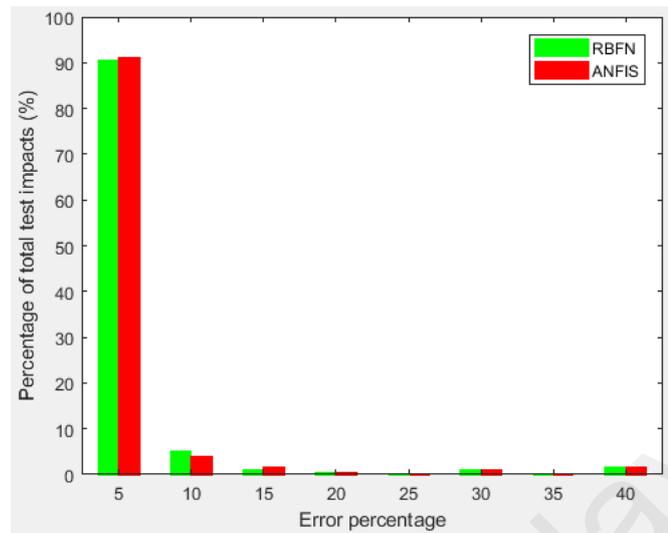


**Fig. 4.9:** (a) Subsample 8, which has best identification performance among all the subsamples for MLP; (b) Subsample 9, which has worst identification performance among all the subsamples for MLP.

Thirdly, Fig. 4.9 shows the best and worst performances for MLP. Although MLP gives worst performance among these four networks according to above graphs, it has the lowest maximum error percentage (58.07%). In addition, the averaged success rate percentage (within 10%) for MLP is close to GRNN.

Due to similar results appear in subsample 2 for RBFN and ANFIS. It is chosen to explore their performance of the success rate percentage of within 5% error. Below figure shows that the success rate percentage within 5% for ANFIS (91.11%) is slightly higher than RBFN (90.56%). Therefore, the ranking for these four networks in impact

quantification is ANFIS > GRNN > RBFN > MLP.



**Fig. 4.10:** Comparison of success rate percentage within 5% for ANFIS and RBFN (subsample 2)

#### 4.2.3 Comparison of impact localization and quantification results with the previous studies

The above section already discussed the findings of this study. It is important to compare them with the previous studies. The best results of previous findings were taken and listed below. Table 4.7 and 4.8 show identification results of various studies regarding to accuracy and precision.

In overall, current study achieved better localization results than previous study according to Table 4.7. It used 10 randomly chosen subsamples, which resulted in the results became more robust and average testing error became smaller. For example, the accuracy of RBFN became more accurate by increasing 1 subsample to 10 subsamples. Although the average testing error in previous thesis (0.7497 radial cm) was lower than current study (0.9446 radial cm) for RBFN, the best testing result for current study could

achieve 0.5972 radial cm. The value of average testing error in previous thesis was between the maximum error and the minimum error of current study for RBFN, which proved that the testing results of current study were reasonable. For these networks that they only had one subsample could not acquire precision. MLP network had different testing result between this study and paper 2 due to different dataset applied. In terms of accuracy, the new proposed networks GRNN and ANFIS had outstanding performance among all the networks. MLP obtained smaller average testing error and higher average success rate than RBFN by combining feature MAT. In terms of precision, GRNN and ANFIS had small error range that close to zero, but ANFIS gave much lower success rate's range (85.56%-100%) compared to GRNN (98.33%-99.44%). In contrast, MLP gave higher success rate' range (88.33%-99.44%) than ANFIS although its error range was very high (0.3644-1.7172 radial cm).

It is clearly to see that ANFIS could achieve best quantification results among all the networks according to table 4.8. For RBFN network, it obtained good results in paper 1 and thesis, whereas the accuracy decreased when 10 subsamples were applied. The average testing errors were 6.21% and 7.49% in paper 1 and thesis, respectively and they were in the range of 3.9010%-16.7829%. It proved that the results of current study were reasonable. Furthermore, the results were more reliable by using 10 subsamples. The accuracy of RBFN was slightly higher than GRNN, but GRNN was much more precise than RBFN. Thus GRNN was proven better than RBFN according to accuracy and precision.

**Table 4.7:** Localization results comparison of current study with previous studies

Various studies	Different networks combined with different features for localization testing	Localization		Ranking in terms of accuracy	Ranking in terms of precision
		Average Error and its range (radial cm)	Average Success rate and its range (%)		
Current study (10 subsamples)	MLP+MAT	0.5685 (0.3644-1.7172)	98.05 (88.33-99.44) (within 2 radial cm error)	3	4
	RBFN+MAT	0.9446 (0.5972-2.3156)	95.11 (81.67-98.89) (within 2 radial cm error)	6	5
	GRNN+MAT	0.3201 (0.2517-0.4683)	99.16 (98.33-99.44) (within 2 radial cm error)	1 (Hint: average error is very small and the average success rate are very high)	1 (Hint: error range and the success rate's range are very high)
	ANFIS+MAT	0.3084 (0.2520-0.7653)	98.00 (85.56-100.00) (within 2 radial cm error)	2	2
Paper 1 (Md Sazzad et al., 2017) (1 subsample)	RBFN+PAT	1.2625	92.00 (within 2 radial cm error)	7	N/A (Hint: cant compute because 1 subsample only)
Paper 2 (Md Sazzad et al., 2017) (10 subsamples)	MLP+MAT	0.76 (0.53-1.02)	96.72 (94.13-99.20) (within 2 radial cm error)	5	3
Thesis (Md Sazzad et al., 2016) (1 subsample)	RBFN+MAT	0.7497	N/A	4	N/A

**Table 4.8:** Quantification results comparison of current study with previous studies

Various studies	Different networks combined with different features for quantification testing	Quantification		Ranking in terms of accuracy	Ranking in terms of precision
		Average Error and its range (%)	Average Success rate and its range (%)		
Current study (10 subsamples)	MLP+(Peak-to-Peak + MAT)	9.1126 (5.9628-14.6063)	83.33 (67.22-94.44) (within 10% error)	6	4
	RBFN+(Peak-to-Peak + MAT)	8.1907 (3.8661-16.7829)	85.22 (61.11-96.11) (within 10% error)	4	3
	GRNN+(Peak-to-Peak + MAT)	8.7924 (7.2954-11.8164)	83.78 (75.56-91.11) (within 10% error)	5	2
	ANFIS+(Peak-to-Peak + MAT)	5.2535 (3.6542-8.8230)	92.11 (80.00-96.11) (within 10% error)	1	1
Paper 1 (Md Sazzad et al., 2017) (1 subsample)	RBFN+Peak-to-Peak	6.21	80.5% (within 10% error)	2	N/A
Paper 2 (Md Sazzad et al., 2017) (10 subsamples)	N/A	N/A	N/A	N/A	N/A
Thesis (Md Sazzad et al., 2016) (1 subsample)	RBFN+(Peak-to-peak+MAT)	7.49	N/A	3	N/A

## CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

### 5.1 Conclusions

This study compared performance of four types of neural networks in terms of impact force identification. Some new results were found based on the objectives and previous researches (Md Sazzad et al., 2017).

In case of location prediction, GRNN and ANFIS had close average testing error, while GRNN showed good result in every subsample. In addition, GRNN had much higher prediction accuracy range (98.33%-99.44%) than ANFIS (81.67%-98.89%). Average success rate for common used network MLP was also very high (98.05%). Although RBFN combined MAT feature and PAT feature were proven better than MLP+PAT, it performed worst among these networks in this study. Therefore, ranking from best to worst network for location prediction of impact force are GRNN, ANFIS, MLP and RBFN.

In terms of magnitude prediction, ANFIS presented outstanding performance in precision and accuracy. Although RBFN showed slightly lower average testing error percentage and higher average success rate percentage within 10% error, one of the subsample had very high testing error (16.7829%) for RBFN. In addition, the lowest success rate percentages for RBFN and GRNN were 61.11% and 75.56%, respectively. GRNN improved accuracy 14.45% than RBFN. MLP was the worst network among these networks, but its performance was not far away other networks. It still can be used for magnitude prediction. Overall, ranking from best to worst network are ANFIS, GRNN, RBFN and MLP.

## 5.2 Recommendations

This study focused on impact force identification by using different types of neural networks and proposed the most proper network to solve the problem. Based on these findings, several recommendations could be come up for further research. Firstly, one or two input feature was used in this study. Because more inputs can provide more information that related to outputs, combination of two or more input features are recommended to use for impact identification. Secondly, fix number and position of sensor were used in this study. Future scope is to optimize sensor quantity and position in order to acquire more accurate results. Thirdly, because success rate percentage was zero for some testing datasets in this study, the original data should be optimized further by using advanced signal filtering processes or better sensor quality against noise. Fourthly, 5 training datasets were used in this study, which was randomly chosen number. Various numbers of training datasets can be studied in order to get lowest testing error. Fifthly, GRNN and ANFIS have been successfully used in impact force identification, they are expected to apply in other similar situations in the future.

## References

- B. Samanta, K. Al-Balushi & S. Al-Araimi (2003), Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection, *Eng. Appl. Artif. Intell.* 16 (7), 657–665.
- Civalek, O. (2004). “Flexural and axial vibration analysis of beams with different support conditions using artificial neural networks”, *Structural Engineering and Mechanics*, 18(3), 303-314. *Journal of Water Supply* 66 (1) 49-61.
- Chan Moon Kim & Manukid Parnichkun (2016). MLP, ANFIS, and GRNN based real-time coagulant dosage determination and accuracy comparison using full-scale data of a water treatment plant. *Journal of Water Supply: Research and Technology*. doi: 10.2166/aqua.2016.022.
- Chandrashekhara, Okafor & Jiang (1997). Estimation of Contact Force on Composite Plates Using Impact-induced Strain and Neural Networks. *Composites Part B* 29B, 363-370.
- Chang, F.-J., & Chang, Y.-T. (2006). Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. *Advances in Water Resources*, 29(1), 1-10. doi: 10.1016/j.advwatres.2005.04.015.
- Demuth, H., Beale, M., & Hagan, M. (2008). Neural network toolbox™ 6. User’s guide.
- D.F. Specht (1991). A general regression neural network. *Trans Neural Netw*, 2 (6), pp. 568-576.
- E. Özkaya & H. Öz (2006). Determination of natural frequencies and stability regions of axially moving beams using artificial neural networks method. *Journal of Sound and Vibration* 252 (4), 782–789.
- Fulkerson, B. (1997). Pattern Recognition and Neural Networks. *Technometrics*, 39(2), 233-234. doi: 10.1080/00401706.1997.10485099.
- Ghajari, M., Sharif-Khodaei, Z., Aliabadi, M. H., & Apicella, A. (2013). Identification of impact force for smart composite stiffened panels. *Smart Materials and Structures*, 22(8), 085014. doi: 10.1088/0964-1726/22/8/085014.
- Ghajari M, Sharif Khodaei Z & Aliabadi M (2012). Impact detection using artificial neural networks. *Key Eng Mater.* 2012;488:767–770.
- Hagan & Menhaj (1994). Training Feedforward Networks with the Marquardt Algorithm. *Transactions on Neural Networks*, Vol. 5, No. 6.
- Hakim & Razak (2013). Adaptive Neuro Fuzzy Inference System (ANFIS) and Artificial

- Neural Networks (ANNs) for structural damage identification. *Structural Engineering and Mechanics*, Vol. 45, No. 6, 779-802.
- Haywood, J., Coverley, P., Staszewski, W., & Worden, K. (2004). An automatic impact monitor for a composite panel employing smart sensor technology. *Smart Materials and Structures*, 14(1), 265.
- Hecht-Nielsen, R. (1987). Kolmogorov's mapping neural network existence theorem. Paper presented at the Proceedings of the international conference on Neural Networks. *HNC, Inc.*
- Hikmet & Murat (2006). Generalized regression neural network in modelling river sediment yield. *Advances in Engineering Software* 37, 63–68.
- H. Md. Azamathulla, Aminuddin Ab. Ghani & Seow Yen Feib (2012). ANFIS-based approach for predicting sediment transport in clean sewer. *Appl Soft Comput.*, 12(3), 1227–1230.
- Hossain, M. S., Ong, Z. C., Ismail, Z., & Khoo, S. Y. (2017). A comparative study of vibrational response based impact force localization and quantification using radial basis function network and multilayer perceptron. *Expert Systems with Applications*, 85, 87-98. doi: 10.1016/j.eswa.2017.05.027.
- Hossain, M. S., Ong, Z. C., Ng, S.-C., Ismail, Z., & Khoo, S. Y. (2017). Inverse identification of impact locations using multilayer perceptron with effective time-domain feature. *Inverse Problems in Science and Engineering*, 26(3), 443-461. doi: 10.1080/17415977.2017.1316496.
- Hossain (2016). *Artificial neural network application in non-collocated impact force identification using appropriate time-domain feature and network type*. (Master's Thesis, University of Malaya, Kuala Lumpur, Malaysia).
- I. Abu-Mahfouz (2003). Drilling wear detection and classification using vibration signals and artificial neural network, *Int. J. Mach. Tools Manuf* 43 (7), 707–720.
- Jang, J.S.R. (1993). ANFIS: Adaptive-network-based fuzzy inference systems, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, pp. 665–685.
- J. LeClerc, et al. (2007). Impact detection in an aircraft composite panel—A neural-network approach, *J. Sound Vib.* 299 (3), 672–682.
- Jones & Sirkis (1997). Detection of Impact Location and Magnitude for Isotropic Plates Using Neural Networks. *Journal of Intelligent Material System and Structures*, Vol. 7.
- Kim, S.-J., & Lee, S.-K. (2009). Experimental identification for inverse problem of a mechanical system with a non-minimum phase based on singular value

decomposition. *Journal of Mechanical Science and Technology*, 22(8), 1504-1509. doi: 10.1007/s12206-008-0312-1.

- K. Worden & W. Staszewski (2000). Impact location and quantification on a composite panel using neural networks and a genetic algorithm, *Strain* 36 (2), 61–68.
- L. Chen & J. Chen, J. Li (2015). Practical compensation for nonlinear dynamic thrust measurement system, *Chin. J. Aeronaut.* 28 (2), 418–426.
- LeClerc, J., Worden, K., Staszewski, W., & Haywood, J. (2007). Impact detection in an aircraft composite panel—A neural-network approach. *Journal of Sound and Vibration*, 299(3), 672-682.
- Lee, J. J., Lee, J. W., Yi, J. H., Yun, C. B., & Jung, H. Y. (2005). Neural networks-based damage detection for bridges considering errors in baseline finite element models. *Journal of Sound and Vibration*, 280(3-5), 555-578. doi: 10.1016/j.jsv.2004.01.003.
- Lin & Huang (2013). An Adaptive-Network-Based Fuzzy Inference System For Predicting Springback OF U-Bending. *Transactions of the Canadian Society for Mechanical Engineering*, Vol. 37, No. 3.
- Liu, Y., & Shepard, W. S. (2005). Dynamic force identification based on enhanced least squares and total least-squares schemes in the frequency domain. *Journal of Sound and Vibration*, 282(1-2), 37-60. doi: 10.1016/j.jsv.2004.02.041.
- Maseras-Gutierrez, Staszewski, Found & Worden (1998). Detection of impacts in composite materials using piezoceramic sensors and neural networks. *Smart Structures and Integrated Systems*, Vol 3329.
- Mahmood Amiri, Hamed Davande, Alireza Sadeghian & Sylvain Chartier (2010). Feedback associative memory based on a new hybrid model of generalized regression and self-feedback neural networks, *Neural Networks* 23 (7), 69–74.
- Mcculloch & Pitts (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics Volume 5*.
- Minsky, Marvin, Papert & Seymour (1969). Perceptrons: An Introduction to Computational Geometry. *MIT Press*. ISBN 0-262-63022-2.
- M.T. Hagan & M.B. Menhaj (1994). Training feedforward networks with the Marquardt algorithm, *Neural Netw. Trans.* 5 (6), 989–993.
- Neha, Ivan & , Arjan (2016). Comparison of adaptive neuro-fuzzy inference system (ANFIS) and Gaussian processes for machine learning (GPML) algorithms for the prediction of skin temperature in lower limb prostheses. *Medical Engineering and Physics* 38, 1083–1089.

- Ng, Andrew, Dean & Jeff (2012). "Building High-level Features Using Large Scale Unsupervised Learning".
- Noorzaei, J., Hakim, S.J.S., Jaafar, M.S., Abang, A.A.A. & Thanoon, W.A.M. (2007), "An optimal architecture of artificial neural network for predicting of compressive strength of concrete", *Indian Concrete Journal*, 81(8), 17-24.
- Qiao, B., Zhang, X., Luo, X., & Chen, X. (2015). A force identification method using cubic B-spline scaling functions. *Journal of Sound and Vibration*, 337, 28-44. doi: 10.1016/j.jsv.2014.09.038.
- Reza Rooki (2016). Application of general regression neural network (GRNN) for indirect measuring pressure loss of Herschel–Bulkley drilling fluids in oil drilling. *Measurement* 85, 184–191.
- S. & R. (2000). Basic Concepts of Artificial Neural network (ANN) Modeling and Its Application in Pharmaceutical Research. *Journal of Pharmaceutical and Biomedical Analysis* 22, 717–727.
- S. Chen, C.F. Cowan & P.M. Grant (1991). Orthogonal least squares learning algorithm for radial basis function networks. *Neural Networks, Transactions on* 2 (2), 302–309.
- Shaikh, Manza & Ramteke (2010). Generalized Regression Neural Network and Radial Basis Function for Heart Disease Diagnosis. *International Journal of Computer Applications* (0975 – 8887).
- Sharif-Khodaei Z, Ghajari M & Aliabadi M (2012). Determination of impact location on composite stiffened panels. *Smart Mater Struct.* 21:105026.
- Smolensky, P. (1986). "Information processing in dynamical systems: Foundations of harmony theory." In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. 1. pp. 194–281.
- Takagi, T. & Sugeno, M. (1985), "Fuzzy identification of systems and its applications to modeling and control", *Trans Syst, Man, Cybernet*, 15, 116-132.
- Thite, A., & Thompson, D. (2003a). The quantification of structure-borne transmission paths by inverse methods. Part 1: Improved singular value rejection methods. *Journal of Sound and Vibration*, 264(2), 411-431.
- Weaver Jr, W., Timoshenko, S. P., & Young, D. H. (1990). *Vibration problems in engineering*: John Wiley & Sons.
- Werbos & P.J. (1975). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*.

- Worden, K., & Staszewski, W. (2000). Impact location and quantification on a composite panel using neural networks and a genetic algorithm. *Strain*, 36(2), 61-68.
- Y.-Y. Liu, et al. (2011). Structure damage diagnosis using neural network and feature fusion. *Engineering applications of artificial intelligence* 24 (1), 87–92.
- Yang, H., Yan, W., & He, H. (2016). Parameters Identification of Moving Load Using ANN and Dynamic Strain. *Shock and Vibration*, 1-13. doi: 10.1155/2016/8249851.
- Z. Zakaria, N.A.M. Isa & S.A. Suandi (2010). A study on neural network training algorithm for multiface detection in static images, *World Acad Sci Eng Technol* 38, 170–173.

University of Malaya