ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING

Dynamic Analysis of Fluid Containing Cylindrical Tanks Using ANN

A Thesis in Structural Engineering

By Estifanos Bekele

April, 2024

Addis Ababa

A Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science

The undersigned have examined the thesis entitled '**DYNAMIC ANALYSIS OF FLUID CONTAINING CYLINDRICAL TANKS USING ANN**' presented by **ESTIFANOS BEKELE**, a candidate for the degree of **Master of Science** and hereby certify that it is worthy of acceptance.

Dr. Bedilu Habte		Apr. 22/24
Advisor	Signature	Date
Dr. Abrham Gebre		22/04/2024
Internal Examiner	Signature	Date
Dr. Fiseha Nega	for	22/04/2024
External Examiner	Signature	Date

Abrham Gebre (Dr.)
Dean, School of Civil & Environmental Engineering

Chairperson	Signature	Date

Director of Postgraduate Program
Addis Ababa University
Addis Ababa Institute of Technology

UNDERTAKING

I certify that this research work titled **"DYNAMIC ANALYSIS OF FLUID CONTAINING CYLINDRICAL TANKS USING ANN"** is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged/ referred.

Estifanos Bekele

ABSTRACT

Cylindrical containers that store liquid are among those delicate structures that are impacted through dynamic loads. These structures that contain fluid are affected due to impulsive and convective pressures created by the liquid inside.

This study utilized an artificial neural networking (ANN) model to analyze the dynamic response of fluid-containing cylindrical tanks. Six input parameters were selected to characterize the geometric and mechanical attributes of the cylindrical tank, including the properties of the fluid it holds. A combined dataset of 6,912 samples from Housner's approximate method was utilized to train and test the ANN model. The training and testing sets yielded R^2 values of 0.9997 and 0.9991, respectively. The ANN model obtained results that were comparable to the results of Housner's approximate method show that ANN simulations can accurately predict the dynamic response of fluid-containing cylindrical tanks. The model can be enhanced to investigate additional parameters that impact the dynamic response of fluid-containing cylindrical tanks, such as the second-order impact caused by axial load, the effect of baffles, etc.

In addition, factors that affect the accuracy and precision of the prediction of the ANN model were investigated and directions were put forward to get a more accurate and precise prediction from available data.

Keywords: Dynamic load, Training, Testing, Neural Network, Cylindrical Tank

ACKNOWLEDGMENTS

First and foremost, the greatest thanks from the depth of my heart is to the Almighty God for his unending blessings. I would like to express my heartfelt gratitude and admiration to my thesis advisor Dr. Bedilu Habte, for his priceless advice, motivation, and clarification during the research period. I would additionally like to thank my entire family for their steadfast patience and dedication, as well as for applauding my success, supporting my goals, and being tolerant of my failures, all of which have helped me become a better person and a problem solver.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

P_i Impulsive pressure

P_s Convective pressure

P_{i1} the impulsive pressure faced by the rigid tank wall

P_{i2} the impulsive pressure caused by the tank wall's flexibility

m_1 impulsive mass

m_2 convective mass

u_1 acceleration corresponds to m_1

u_2 acceleration corresponds to m_2

P_1 the seismic force that corresponds to m_1

P_2 the seismic force that corresponds to m_2

P_3 the seismic force that corresponds to m_3

M_1 moment corresponds to P_1

M_2 moment corresponds to P_2

W_1 weight of fluid transitioning with the rigid wall generating impulsive force

W_2 weight of fluid generating convective force

W_3 weight of constrained fluid

h_1 height causing moment to bend to the shell of gross weight W_1

h_2 height causing moment to bend to the shell of gross weight W_2

h_3 height causing moment to bend to the shell of gross weight W_3

R radius of the cylindrical tank

h	height of fluid level
γ	unit weight of the fluid
W_w	weight of the fluid
h_1^0	the equivalent height level to incorporate the bottom stress of weight W_1
h_2^0	the equivalent height level to incorporate the bottom stress of weight W_2
T	the period of resonating fluid
K	spring stiffness
ω	natural circular frequency of the fluid
g	Acceleration due to gravity
OTM	the overturning moment
BM	the bending moment
A	the maximum acceleration of the sloshing mass
A_0	the maximum acceleration of the ground
A*	the maximum acceleration of a system with one degree of freedom
H	fluid height
L	the length (height) of the shell
h,t	the thickness of the shell
(r, z)	the cylindrical coordinate system employed with the central point of the base serving as the origin
(w, u)	the axial and radial displacement elements of a point on the middle surface of the shell
ϕ(r, z, t)	potential function of velocity

I_0	the modified function of Bessel of the first type with order zero
E	the elasticity modulus of the material
υ	the material's poisson's ratio
t	time
ρ_l	the density of the fluid
ρ_s	the density of the shell
P(r,z,t)	the pressure imposed to the container's wall at any given time t
ω_n	the natural frequency of the shell
D	the extensional stiffness of the shell material
K	the bending stiffness of the shell material
U(t)	the strain energy absorbed by the shell material
T(t)	the kinetic energy emanating from the shell material
N_z	membrane force generated along axial direction
N_θ	membrane force generated along hoop direction
M_z	the axial bending moment generated
ε_z	the normal strain in the axial direction
ε_θ	the normal strain in the hoop direction
K_z	the mid surface variation in the curvature
m(z)	the shell's mass for each unit area
{d}	displacement vector
{ε}	strain vector
[P]	The differential operator

S_{as}	The spectral measurement of the pseudo-acceleration that matches up to the fundamental sloshing frequency
Σ	Symbolizes the direct integration to integrate in accordance with the matrix a_e
m_s	the effective sloshing mass in an operational model
H_s	the height where the effective sloshing mass operates
I_j	input layer neuron activation level
H_k	hidden layer neuron activation level
f	the activation function of an item in the layer that is hidden
bias	the weight from a permanently active item
μ	the fixed coefficient that dictates the impact of previous shifts in weight on the current direction of motion in the weight space
O_l	activation of a singular item used in the output layer
W_{jk}	link among input units and hidden layers
W_{kl}	link among this unit and the hidden layer unit
δ	connection between the input and hidden layers
lrate	learning rate
R^2	coefficient of determination

CHAPTER 1 INTRODUCTION

1.1 Background of the Research

Numerous studies have explored liquid-filled cylindrical tanks, employing mathematical and experimental methods for modeling. Among notable contributors, George W. Housner's significant contributions stand out. He devised a simplified analytical method rooted in experimental studies to characterize hydrodynamic pressure, enabling the calculation of forces and overturning moments exerted at the tank's base [19].

Analytical approaches are inadequate for deriving closed form solutions due to the complexity of the circumstance. Advancements in numerical techniques, powerful computers, and the implementation of the finite element method have facilitated the analysis of liquid-containing tanks. Different modeling approaches that represent the issue could be used. Among them include the creation of a liquid finite element model, the use of displacement-based finite elements to represent the shell region, and the application of analytical approaches to model the liquid.

The examination of liquid-containing structures is, as previously stated, quite challenging. The issue becomes complex as a result of the dynamic effect and fluid structure interaction. This paper aims to use existing neural fitting software to create an artificial neural network that includes the effect of the liquid within the tanks and to examine the behavior of the tanks.

In this study, a neural network model was developed which allows structural designers to use it for the analysis of ground supported cylindrical tanks as well as for the design of fluid containing cylindrical tanks.

1.2 Statement of the Problem

Tanks with a cylindrical shape are used for a variety of things, including the storage of liquids like water, petroleum products, chemicals of various types, etc. These tanks range in size from small to large and can be used for both industrial and domestic purposes. These structures respond to earthquake loads much differently than other structures because they are in contact with liquid. The seismic force also transmits hydrodynamic pressure in addition to hydrostatic pressure. For the design of cylindrical tanks, this liquid-structure interaction is important, and due consideration should be given to it. The assessment of structures that include liquids is quite complex. The issue becomes more challenging as a result of the dynamic effect and the fluid-structure interaction.

The purpose of researching this issue is to examine the behavior of the tanks and to create an artificial neural network using an existing neural network fitting software that integrates the effect of the liquid inside the tanks.

1.3 Objectives

1.3.1 General Objective

The general objective of this research is to investigate how dynamically loaded, ground-supported cylindrical tanks behave while taking into account hydrodynamic forces using Artificial Neural Networks.

1.3.2 Specific Objectives

The specific objectives of the thesis are listed as follows:

- To develop a neural network model that gives a more precise and accurate prediction.
- Analysis of the dynamic response of cylindrical tanks with different aspect ratios and heights.
- Investigation of the minimum freeboard requirement of fluid containing cylindrical tanks under dynamic loading.

1.4 Research Questions

- Which analysis method gives more precise, accurate, and reliable results for shallow tanks and slender tanks?
- What factors affect the accuracy and precision of neural networks?
- What is the effect of variation of the aspect ratio on the dynamic response of cylindrical tanks?
- What is the effect of variation in the density of fluids on the dynamic response of cylindrical tanks?

1.5 Significance of the Study

This research will play a constructive role in helping review different methods of dynamic analysis of fluid-containing cylindrical tanks. It will also play a crucial part in understanding the use of ANNs in predicting structural responses.

1.6 Scope of the Study

This study is performed on cylindrical tanks that carry a fluid volume ranging from 100 m^3 to 10,000 m^3. Where the height of the cylindrical tanks ranges from 1.5 m to 9 m and the radius of the cylindrical tanks ranges from 6 m to 22.5 m. The cylindrical tanks are flat bottomed and ground supported. The steel cylindrical tank walls are considered to be rigid. The current study develops a code for predicting the dynamic response of fluid containing cylindrical tanks using the Levenberg-Marquradt algorithm, which is one of the top performers in the neural network fitting MATLAB toolbox.

CHAPTER 2 LITERATURE REVIEW

2.1 General

For the storage of liquids like water, oil, and natural gas, among others, liquid storage tanks are crucial structures. Due to the flowing nature of the material they hold, they are susceptible to damage during seismic events. Early theories on seismic response assumed that cylindrical tanks were inflexible and ignored the flexibility of the tank shell.

Due to the complexity of the issue, no closed form solutions are often accessible, and those who take into account this effect do so while ignoring the effects of sloshing and flexibility.

For several years, cylindrical tanks that contain liquid have been a matter of concern. They were engaged in both experimental and analytical studies of how a liquid moved inside cylinder and rectangle shaped tanks when they were subjected to horizontal ground motion.

Housner investigated the hydrodynamic pressures that are generated inside these tanks when they experience ground motions in 1957 [1, 3]. For expressing force and moment effects on the rigid tank due to liquid motion, the results were given as equivalent masses together with their locations [6].

The state-of-the-art of liquids in moving containers were the subject of review studies provided by Cooper in 1960 and Abramson in 1963 [1], with a lot of the work matters linked to space vehicle technology. The first significant damage to liquid storage tanks was caused by the 1964 Alaska earthquake, which also inspired numerous studies into the dynamic properties of the containers [2]. In 1976, Howard I. Epstein revised the findings of Housner's mechanical model [1] introducing extra factors to take the containers' flexibility into account.

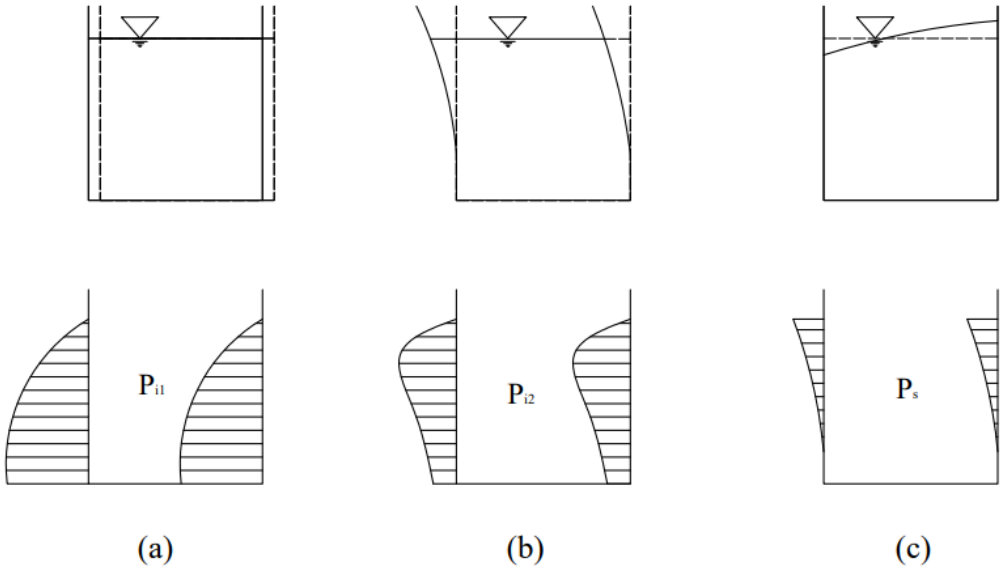
Medhat A. Haroun is responsible for a huge percentage of the works on tanks that were produced in the 1980s. In 1982, G.W. Housner and Medhat A. Haroun conducted a detailed analysis of the issue while taking the interaction problem into account [2]. Furthermore, between 1983 and 1985, he and his colleagues developed a technique for

conducting analytical and numerical approaches to the issue. The approaches implemented will be briefly addressed below.

2.2 Review of Basic Concepts

Structures in contact with liquid exhibit substantially distinct behavior during during load from seismic events. The impulsive pressure, P_i, is the force per unit area that the fluid's inertia applies to the tank walls while in horizontal earthquake excitation of a fluid container. Along with the impulsive pressure, the excitation will also upset the initial liquid level, causing sloshing. In addition convective pressure, P_s, builds up on the container's walls as a result of the sloshing action.

There are two components to the impulsive pressure that is brought on by inertia forces. The primary is caused by the pressure pushing on the container's hard enclosure and the secondary is brought on by the tank wall's flexibility. The latter decreases as tank stiffness increases. Less pressure is generated by a flexible tank the stiffer the tank is [3].



a) Impulsive pressure posing on the rigid wall b) Impulsive pressure posing on the flexible wall c) Convective pressure

Figure 2.1 Dynamic liquid pressures imposed on the container wall [8]

2.3 Housner's Approximate Method

2.3.1 Assumptions

Housner's approximate technique hinges on the following assumptions:

1. Flat-bottom
2. Rigid container walls
3. Containers having a consistent circular or rectangular cross-section
4. Solely horizontal earthquake excitation

Results from both experimental and analytical methods served as the foundation for Housner's approximation method. His lumped mass approach to modeling the issue was created from the findings of the investigations. Due to the liquid's inertia and sloshing, the horizontal ground acceleration creates convective and impulsive forces. Accordingly, the mass of the fluid was divided into the convective mass m_2 and the impulsive mass m_1.

The accelerations u_1 and u_2 act on the masses m_1 and m_2, producing the equivalent seismic forces P_1 and P_2, respectively. When these figures are multiplied to the heights h_1 and h_2, they form the moments M_1 and M_2 on the vessel's bottom. The masses m_1 and m_2 will be chosen in order to produce stresses that are comparable to those produced experimentally through the fluid within the seismically overwhelmed container [3].

Due to the presumption that the walls are stiff, the tanks' absolute displacement is the same as that of the ground. The geometry of the tank has an impact on how impulsive mass is estimated. According to its geometry, the tanks fall into two categories known as slender and shallow tank. The involvement of the convective and impulsive mass can be ascertained with the use of this classification.

Housner showed how mathematical models may be used to calculate the forces brought on by the horizontal acceleration of the fluid in the vessel.

2.3.2 Classification of Tanks

In the case of a cylindrical storage container with radius R, Holding a fluid that is incompressible with unit weight γ, saturated to the depth H, the total weight of the fluid W_w is given by:

$$W_w = \pi R^2 H \gamma \quad (2.1)$$

A cylindrical container with a height to radius proportion of 1.5 or below is considered shallow. Figure 2.2 depicts the lumped mass model for this scenario. It comprises a mass of the weight, W_1, moving in synchronization alongside the container's rigid enclosure to generate the impulsive force. Additionally, a mass of the weight, W_2, produces the convective force.

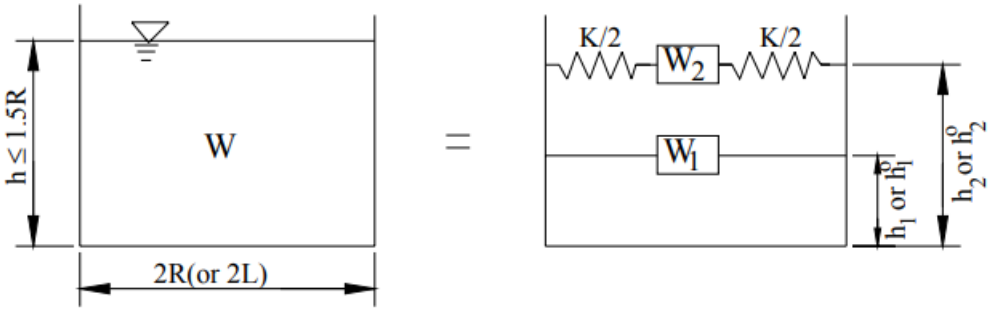


Figure 2.2 Shallow Tank- Lumped mass approach [8]

Where W is the total weight of the liquid

W_1 is weight of fluid transitioning with the rigid wall generating impulsive force

W_2 is weight of fluid generating convective force

K is the equivalent spring stiffness

h_1 is height causing moment to bend to the shell of gross weight W_1

h_2 is height causing moment to bend to the shell of gross weight W_2

h_1^0 is the equivalent height level to incorporate the bottom stress of weight W_1

h_2^0 is the equivalent height level to incorporate the bottom stress of weight W_2

The heights of these weights were determined by obtaining the correct moment about the base. As shown in Fig. 2.2, the corresponding heights are denoted by the symbols h_1 and h_2. Bending moments about the structure's base are computed using the heights h_1 and h_2. The shell member resists the bending moment slightly above the level of the base.

The same weights were employed to establish the dynamic pressure distribution on the container's bottom, which is used to determine the overturning moment [22]. But in order to take the bottom pressure into account, new corresponding heights were utilized. According to Figure 2.2, the heights h_1^0 and h_2^0 are used to compute the structure's overturning moment. The following equations yield the expressions for the heights:

$$h_1 = \frac{3}{8}\mathrm{H} \tag{2.2}$$

$$h_2 = H[1 - \frac{\cosh\left(1.84\frac{H}{R}\right) - 1}{1.84\left(\frac{H}{R}\right).\sinh\left(1.84\frac{H}{R}\right)}] \tag{2.3}$$

$$h_1^0 = \left(\frac{H}{8}\right)[\frac{4}{\left[\frac{\tanh\left(\sqrt{3}.\left(\frac{R}{H}\right)\right)}{\sqrt{3}.\left(\frac{R}{H}\right)}\right]} - 1] \tag{2.4}$$

$$h_2^0 = H[1 - \frac{\cosh\left(\frac{1.84H}{R}\right) - 2.01}{1.84\left(\frac{H}{R}\right).\sinh\left(1.84\frac{H}{R}\right)}] \tag{2.5}$$

The period of the resonating fluid, T, is given by:

$$\mathrm{T} = \frac{2\pi}{\omega} \tag{2.6}$$

Where ω is natural circular frequency of the fluid

The natural circular frequency can be calculated as follows:

$$\omega^2 = \frac{1.84g}{R}\tanh(1.84\frac{H}{R}) \tag{2.7}$$

Where g represents the acceleration due to gravity

The maximum forces that the two masses impose on a rigid tank in the case of a vessel exposed to a single-dimensional horizontal seismic base excitation are:

$$\mathrm{P_1} = \frac{\mathrm{W_1}}{\mathrm{g}}\,\mathrm{A_0} \tag{2.8}$$

$$\mathrm{P_2} = \frac{\mathrm{W_2}}{\mathrm{g}}\,\mathrm{A} \tag{2.9}$$

Where P_1 is the seismic force that corresponds to m_1

P_2 is the seismic force that corresponds to m_2

A_0 is the maximum ground acceleration

A is the maximum acceleration of the sloshing mass, which can be calculated using the seismic response spectrum using the period previously mentioned. Given that the weights are determined by:

$$W_1 = W_w[\frac{\tanh\left(\sqrt{3}.\left(\frac{R}{H}\right)\right)}{\sqrt{3}.\left(\frac{R}{H}\right)}] \quad (2.10)$$

$$W_2 = W_w.0.18\left(\frac{R}{H}\right).\tanh\left(1.84\frac{H}{R}\right) \quad (2.11)$$

Given that the forces act simultaneously, the maximum bending moment (BM) can be determined by:

$$BM = P_1h_1 + P_2h_2 \quad (2.12)$$

The overturning moment (OTM) is provided by:

$$OTM = P_1h_1^0 + P_2h_2^0 \quad (2.13)$$

The generation of liquid velocity relative to the tank in the case of tall (slender) tanks is constrained to an estimated height of 1.5R from the top free surface. Slim tanks are the ones with a greater than 1.5 height to radius ratio. Figure 2.3 depicts the mathematical model suitable to this scenario. The situation (i.e., a H/R ratio of 1.5) could be pictured as the tank being divided into two sections by a fixed rigid membrane. The restricted liquid is represented by an extra mass of weight W_3.

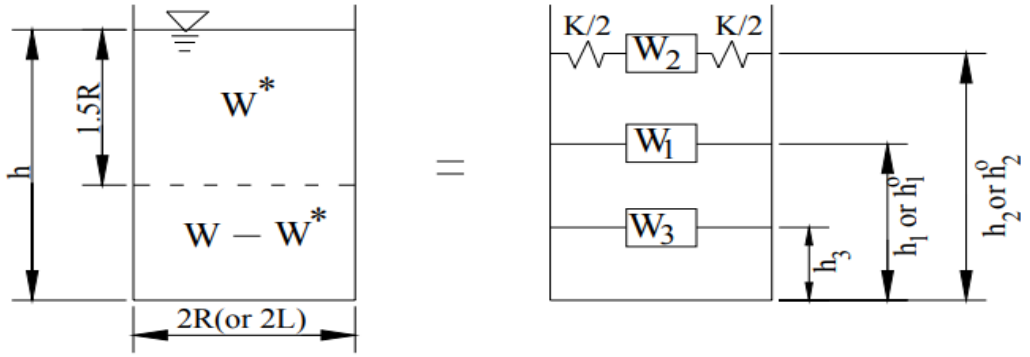


Figure 2.3 Slender Tank- Lumped mass approach [8]

Where W_3 is the weight of constrained liquid

h_3 is height causing moment to bend to the shell of gross weight W_3

The maximum forces that correspond to each mass can be calculated as follows:

$$P_1 = \frac{W_1}{g} A_0 \quad (2.14)$$

$$P_2 = \frac{W_2}{g} A^* \quad (2.15)$$

$$P_3 = \frac{W_3}{g} A_0 \quad (2.16)$$

Where P_3 is the equivalent seismic force corresponding to W_3

A* is the maximum acceleration of a system with one degree of freedom system for period of T*, while T* can be identified by:

$$T^* = 4.65(\frac{R}{g})^{1/2} \quad (2.17)$$

The weights are provided by:

$$W_1 = \gamma.(1.5R^3)\pi.(0.7095) \quad (2.18)$$

$$W_2 = W_w.0.318.\left(\frac{R}{H}\right)\tanh(1.84\frac{H}{R}) \quad (2.19)$$

$$W_3 = \gamma(H - 1.5R).R^2\pi \quad (2.20)$$

The bending moment generated by fluid is provided by:

$$BM = P_1h_1 + P_2h_2 + P_3h_3 \quad (2.21)$$

The heights are provided by:

$$h_1 = \frac{3}{8}.(1.5R) + (H - 1.5R) \quad (2.22)$$

$$h_2 = H[1 - \frac{\cosh\left(\frac{1.84H}{R}\right) - 1}{1.84\left(\frac{H}{R}\right).\sinh\left(\frac{1.84H}{R}\right)} \quad (2.23)$$

$$h_3 = \frac{H - 1.5R}{2} \quad (2.24)$$

$$h_1^0 = \left(\frac{1.5R}{8}\right)\left[\frac{4}{0.7095} - 1\right] + H - 1.5R \quad (2.25)$$

$$h_2^0 = H[1 - \frac{\cosh\left(\frac{1.84H}{R}\right) - 2.01}{1.84\left(\frac{H}{R}\right).\sinh\left(\frac{1.84H}{R}\right)}] \quad (2.26)$$

The well-known streamlined Housner strategy, which is being employed today, was utilized to design the majority of tanks in the past. The method excludes the effect of a flexible wall and only takes into account the hydrodynamic pressure on a rigid wall. The

first significant damage to liquid storage tanks was brought on by the 1964 Alaska earthquake [2]. The hydrodynamic pressure that actually existed was found to be higher than the one predicted by Housner's method. Additional corrective factors were taken into consideration so as to account for the wall's flexibility [8].

2.4 Artificial Neural Networks

Artificial neural networks (ANNs) are a particular kind of machine learning algorithm that are roughly modeled after the structure and activity of the human brain. ANNs are made up of interconnected nodes or neurons that process input data and produce output data. The interconnections among neurons are weighted and adjusted during the learning process to improve the network's ability to make accurate predictions.

There are several types of ANNs, including feedforward neural networks, convolutional neural networks, and deep neural networks. Every network type works most effectively for a specific task, which may involve natural language processing, regression, classification, or image recognition.

The development of ANNs can be traced back to 1940s and 1950s, when researchers began to explore the basic principles of neural computation. One of the earliest ANNs was the perceptron, a type of feedforward network developed by Frank Rosenblatt in 1958. The perceptron was able to learn simple patterns and used in applications such as character recognition.

In the 1960s and 1970s, research on ANNs declined as researchers focused on other areas of artificial intelligence (AI). However, the 1980s, the backpropagation algorithm was developed, which allowed for the efficient training of multi-layered feedforward networks. This led to a resurgence of interest in ANNs, and they were used in several forms of applications including image processing, speech recognition and robotics.

In the 1980s and 1990s, convolutional neural networks (CNNs) were developed and researched by Yann Le Cun, which were able to learn hierarchical representation of visual data CNNs were used in applications such as face recognition, object recognition, and medical image analysis.

In the 2000s recurrent neural networks (RNNs) were developed which were able to model sequential data such as speech and text. RNNs were utilized in tasks such as speech recognition, natural language processing, and machine translation.

In the 2010s, deep learning, which refers to the use of ANNs with many layers, became increasingly popular. Deep learning enabled breakthrough in computer vision, natural language processing, and game playing among other areas.

Today, ANNs continue to be an active area of research, with ongoing efforts to improve their performance, interpretability, and efficiency.

ANNs are particularly useful for complex problems where traditional analytical methods may be difficult or impossible to use. In Civil Engineering, ANNs have been used to predict the behaviors of structures under various loading conditions.

Overall, ANNs have proven to be powerful and versatile tool for structural analysis, and their continued development and refinement is likely to have a significant impact on the field of engineering in the years to come.

An artificial neural network (ANN) is made up of many interconnected processing units. Each unit of processing or node, established as a neuron, acquires input from other units to which it is linked, executes a process, and then delivers the output to other units for processing. Each node in a neural network executes a single computation irrespective of the others. Hence, neural networks have a parallel framework, allowing them to operate on concurrently processing computer systems [11].

ANNs have been deployed and served different sectors of the industrial science such as geology & geophysics, drilling & completion, production and facilities, reservoir engineering, and also in third generation technologies such as genetic algorithms and molecular modelling [21].

The typical arrangement of a biological neuron is illustrated in Fig. 2.4. The central portion is referred to as the cell body. It produces several extensions designated as dendrites. The axon, a single tubular fiber, spreads into several small branches. The dendrites serve as receivers for impulses from neurons that are nearby. The axon passes

on neural signals to different neural cells. Synapses are the connections between the neural cells.

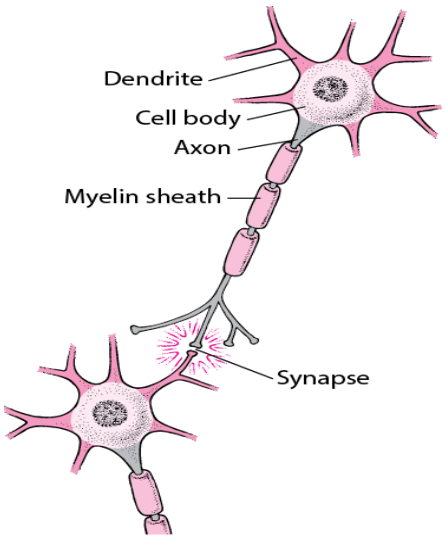


Figure 2.4 Illustration of a biological neuron [11]

Table 2.1 Relationship between Components of Biological and Artificial Neural Networks

Biological Neural Network	**Artificial Neural Network**
Dendrite	Inputs
Cell body	Node
Synapse	Weight
Axon	Outputs

Classification, recognition of speech, image processing, optimization problems, medical and commercial applications, and robotics and control have all been tackled with varying degrees of success using ANNs. Problems suitable for applications involving ANN possess the following aspects [11]:

- Data-intensive applications and rely on various parameters
- The problem zone contains extensive historical data as well as illustrations
- The dataset is inadequate, distorted, and/or has inaccuracies

2.4.1 Artificial Neural Network Architecture

The most commonly known ANN topologies are the Boltzmann machine, the competitive learning network, the Backpropagation network, and the Hopfield network.

The third type obtained its name from the way it learns, which is to back-propagate faults observed at the output nodes. Most of neural network application developers use the back-propagation network model.

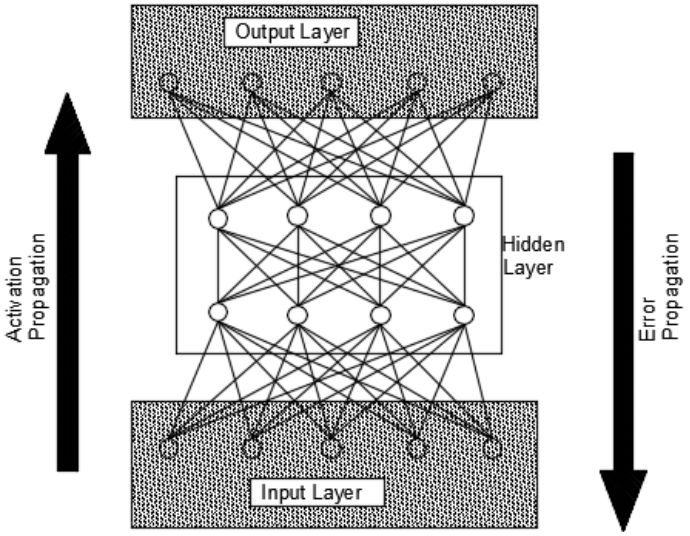


Figure 2.5 Design of a backward propagation neural network [12]

2.4.2 Classification of ANNs

i) Based on Information Flow

- **Feedforward Neural Networks (FNNs):-** Information is transmitted from the input to the output layers in a single direction. These are the most basic types of artificial neural networks, often employed in tasks involving regression and classification.
- **Recurrent Neural Networks (RNNs):-** Information is able to move in both directions, including loops, which makes them ideal for handling sequential data such as text or time series.
- **Modular Neural Networks:-** Integrating multiple smaller ANNs to carry out specific tasks benefits flexibility and reusability.

ii) Based on Functions and Applications

- **Multilayer Perceptrons (MLPs):-** A type of FNN with multiple hidden layers, ideal for complex pattern recognition and classification.

- **Convolutional Neural Networks (CNNs):-** is a kind of FNN which performs well at image recognition and processing due to its shared weight structure.
- **Radial Basis Function Networks (RBFNs):-** Use radial basis functions for interpolation and approximation, which are commonly used in control systems.
- **Generative Adversarial Networks (GANs):-** Two competing artificial neural networks generate and critique data, allowing for realistic image generation and other creative tasks.

Table 2.2 Similarities and Differences of Various Types of ANNs

ANN Type	Similarities	Differences
MLPs	✓ Must be trained with training examples before becoming fully fuctional ✓ Need to be provided with input sets to produce output results ✓ MLPs and CNNs are preferred for image and character recognition	✓ takes vector as input ✓ have the same number of input and output layers ✓ can be trained faster
CNNs		✓ takes tensor as input ✓ can comprehend more effectively the spatial relations between pixels in images
RBFNs		✓ implemented for classification, regression, and time-series prediction
GANs		✓ create new data circumstances that closely mimic the training data

2.4.3 Machine Learning in ANNs

Due to the fact there are so many machine learning strategies available, let's take a look at them individually:

- **Supervised Learning:-** have an initial grasp of what the correct output values should be.
- **Unsupervised Learning:-** is a type of machine learning in which algorithms are given data with no labels or detailed guidance on what to do with it.
- **Reinforced Learning:-** is a subfield of machine learning that allows AI-based systems to learn using trial-and-error methods in a dynamic environment.

2.4.4 Neural Networks Useful Properties and Capabilities

Neural networks offer the following beneficial attributes and traits:-

1. **Nonlinearity:-** An artificial neuron may qualify as linear or nonlinear. A neural network made up of nonlinear neurons is inherently nonlinear. Nonlinearity is a vital property, particularly if the actual mechanism that generates the input signal is innately nonlinear.
2. **Input–Output Mapping:-** The learning process consists of adjusting the synaptic weights of NN using a set of identified training scenarios. Each scenario includes a unique input signal and the expected response. The networks are trained for a large number of scenarios in the set until their performance reach a state that is stable. The previously used training examples may be deployed in a different order during the training session. As a result, the network acquires knowledge from scenarios by creating an input-output linking for the particular situation at being performed.
3. **Adaptivity:-** NNs can adapt their synaptic weights to changes in their surroundings. And they could be programmed to modify their synaptic weights in real time. The network's adaptability allows it to be an effective tool for versatile pattern categorization, adaptive control, and adjustable signal processing. As a general principle, the more adaptable we make the system while keeping it stable, the better its efficiency will be when it is needed to function in a fluctuating setting.
4. **Evidential Response:-** In the context of pattern classification, a neural network can be developed to provide information not only about the specific pattern to be chosen, but also about the certainty of the decision. This last piece of information can be used to eliminate puzzling patterns, thereby improving the network's capacity for classification.
5. **Contextual Information:-** A neural network's structure and activation state encode knowledge. The joint activity of all neurons in the network has the potential to impact each neuron in the entire system.
6. **Fault Tolerance:-** When deployed in hardware, a neural network could have the ability to be innately tolerant of errors or capable of tough calculations, which means that its efficiency declines smoothly under adverse circumstances during operation. For instance, if a neuron or one of its tying links gets harmed,

its retrieval of a previously stored pattern diminishes the level of quality. Yet, due to the dispersed nature of data stored in a system, the harm caused must be substantial before an entire network's responsiveness is seriously compromised. Instead of a complete breakdown, the neural network exhibits a gradual decline in efficiency. Though a certain empirical proof supports robust processing, it is commonly uncontrolled [15].

7. **VLSI Implementability:-** In response to its massive simultaneous execution, a neural network can perform faster at computing specific operations. As a result, neural networks are ideal for carrying out very large-scale integrated (VLSI) technology.
8. **Uniformity of Analysis and Design:-** As information processors, neural networks are basically universal. We say this as a matter of the same notation is utilized in all fields where neural networks are used. This characteristic presents itself in multiple ways:-

- Neurons, in some form or another, are components shared by all neural networks.
- Because of this similarity, theories and learning methods in various neural network applications can be shared.
- Modular networks can be established through linking of modules.

9. **Neurobiological Comparison:-** Neural networks are inspired by the brain, which exemplifies fault-tolerant parallel processing that is both agile and powerful. Neurobiologists have a keen interest in artificial neural networks as an investigative instrument for translating neurobiological phenomena. Engineers, on the contrary, examine neurobiology for novel approaches that solve issues that are more complicated than those based on regular encoded concepts.

2.4.5 Pros and Cons of ANNs

Among the advantages of artificial neural networks some are listed below:-

- The capacity to extract information from noisy and incomplete data.
- Gain experience and intelligence by means of self-training and organization.
- High potential for a quick optimization.
- Their applicability for situations with complicated algorithmic solutions.

The drawbacks of ANNs can be described as follows:-

- They provide solutions and outcomes without explaining how they arrived at their conclusions.
- The accuracy of ANN performance largely depends on the caliber of training examples.

2.5 Analytical Method

The controlling differential equations for the shell surfaces and the fluid serve as a basis for the analytical model that is used for analysis. Different analytical techniques were developed, but the one that incorporates use of the theory of thin shell motion is the most common [7].

2.5.1 Cylindrical Tank and Coordinate System

The cylindrical tank under examination is depicted in Fig 2.4. The tank is a circular, thin-walled cylindrical fluid container with a radius of R, a length of L, and a thickness of h which is supported by the ground. H denotes the liquid height. The origin is the center of the base in a cylindrical coordinate system. u and w represent the axial and radial displacement components of a point on the middle surface of the shell, respectively [7].

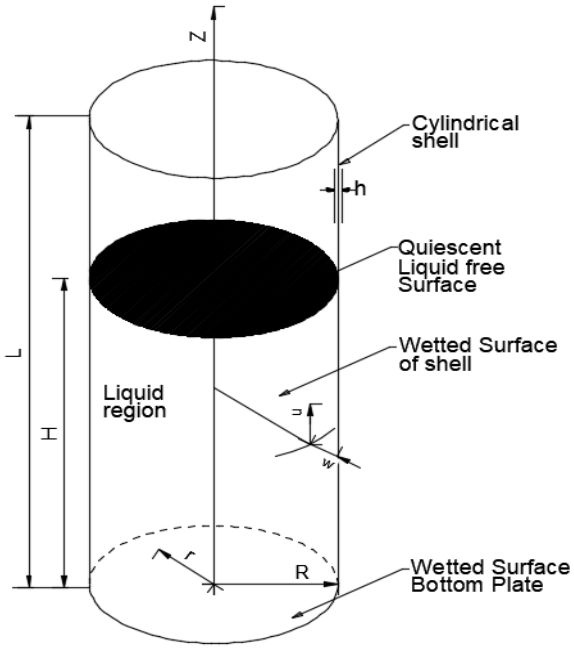


Figure 2.6 Coordinate System of the Cylindrical Tank [2]

2.5.2 Governing Equations of Liquid Motion

In the irrotational flow of an incompressible fluid, the velocity potential function ϕ (r, z, t), follows the Laplace equation within the liquid's region [7].

$$\nabla^2\phi = 0 \tag{2.27}$$

The above equation shows that Laplace equation is satisfied.

Apart from being an harmonious function, ϕ should satisfy certain boundary requirements, which can be summarized as follows:

1. At the rigid container bottom, the vertical liquid velocity is zero.

$$\frac{\partial\phi}{\partial z}(r,0,t) = 0 \tag{2.28}$$

2. The liquid near the elastic shell's wall (r=R) must move radially with the same velocity as the shell.

$$\frac{\partial\phi}{\partial r}(R,z,t) = \frac{\partial w}{\partial t}(z,t) \tag{2.29}$$

   Where w(z, t) is the radial displacement element of a point in the middle surface of the shell

3. If sloshing modes are not ignored, two boundary conditions(i.e. velocity and pressure) must be established at the liquid free surface. Only one surface condition has to be provided if sloshing modes are neglected.

$$\frac{\partial\phi}{\partial t}(r,H,t) = 0 \tag{2.30}$$

The solution that conforms to the above equations can be stated as:

$$\phi(r,z,t) = \sum_{i=1}^{\infty} A_i\, I_0(\alpha_i R)\cos(\alpha_i z) \tag{2.31}$$

where I_0 = the modified function of Bessel of the first type with order zero

$$\alpha_i = \frac{(2i-1)\pi}{2H} \tag{2.32}$$

$$A_i = \frac{2\int_0^H w(z,t)\cos(\alpha_i z)\,dz}{\alpha_i H I_0(\alpha_i R)}$$

The pressure dispersion, P(r, z, t), could be estimated by applying the Bernoulli equation and can be obtained by:

$$P(r,z,t) = -\rho\frac{\partial\phi}{\partial t} + \rho_1 g(H-z) \tag{2.33}$$

2.5.3 Governing Equations of Shell Motion

A cylindrical shell experiencing axisymmetrical resonance is primarily governed by two differential equations: The first equation is of the second order and guides the dynamic equilibrium in the axial direction, is indicated in Eq. 2.35. The second equation is of the fourth order, is indicated in Eq. 2.36, guides dynamic equilibrium in the radial direction [7].

The equations that govern can be summarized as such:

$$\frac{\partial^2 u}{\partial z^2} + \frac{\upsilon}{r}\frac{\partial w}{\partial z} - \frac{\rho_s h}{D}\frac{\partial^2 u}{\partial t^2} = 0 \tag{2.34}$$

$$K\frac{\partial^4 w}{\partial z^4} + \frac{Eh}{(1-\upsilon^2)}\frac{w}{R^2} + \frac{\upsilon Eh}{(1-\upsilon^2)R}\frac{\partial u}{\partial z} + \rho_s h\frac{\partial^2 w}{\partial t^2} = \begin{cases} P(z,t) \\ 0 \end{cases} \tag{2.35}$$

Where E= the elasticity modulus of the material

υ = the material's poisson's ratio

P (z, t) = the pressure imposed to the container's wall at any time t

where 0 < z < H

K and D represent the shell's bending and extensional rigidities respectively.

The mathematical equations for K and D are provided as:

$$D = \frac{Eh}{(1-\upsilon^2)} \tag{2.36}$$

$$K = \frac{Eh^3}{12(1-\upsilon^2)} \tag{2.37}$$

The two equations of motion decouple if υ is set to zero. Furthermore, equations 2.35 and 2.36 demonstrate two different behaviors of the shell in the scenario of an empty tank, P(z, t) = 0. The natural frequencies can be obtained using the first one of these equations, which is equivalent to the one articulating the free vibration of a rod coming across axial resonances.

$$\omega_n = \frac{(2n-1)\pi}{2L}\sqrt{\frac{E}{\rho_s}} \tag{2.38}$$

The uncoupling and vacant vessel assumptions, the second governing differential equation can be simplified as follows:

$$\frac{d^4w}{dz^4} - \beta^4\omega = 0 \tag{2.39}$$

The value of β is given as follows:

$$\beta^4 = \frac{\rho_s h\varpi^2 - \frac{Eh}{R^2}}{Eh^3/_{12}} \tag{2.40}$$

Given that the above equation clearly governs the free vibration of a flexural beam, which is uniform, the natural frequencies of vibration can be estimated by:

$$\varpi_i = \left(\frac{\frac{Eh^2}{R^2} + \beta_i^4\frac{Eh^3}{12}}{\rho_s h}\right)^{1/2} \tag{2.41}$$

Whenever the shell's bending rigidity is ignored, the equation simplifies to:

$$\varpi_i = \frac{\left(\frac{E}{\rho_s}\right)^{1/2}}{R} \tag{2.42}$$

The preceding equation represents the circular natural frequency of a closed ring undergoing uniform radial expansion. It is worth noting that, in practice, v is not equal to zero. However, the equations still provide a decent estimation of the natural frequencies of an empty vessel [7].

2.6 Finite Element Method

The finite element approach is utilized to condense infinite degrees of freedom structures to a network of finite elements containing particular nodal points. Lateral movement of any particular structure is expressed as a function of finite number of displacement coordinates, making use of the lumped mass and generalized coordinate concepts. This strategy is almost universally applicable to all structural system problems [8].

The finite element method (FEM) can be used to solve a wide variety of situations. It ranges from an arrangement of one-dimensional finite elements to three-dimensional complex problems. It is possible to engage in both static and dynamic analysis [4, 5].

Finite element analysis provides a more detailed numerical analysis method for studying the behavior of fluid-structure interactions with one another. The liquid mass can be calculated using the Lagrangian, Eulerian, or Arbitrary Lagrangian Eulerian (ALE) formulation [3, 6].

In fluid mechanics, the Eulerian approach is commonly utilized. In this scenario, the computational mesh remains fixed while the liquid shifts in relation to the grid. The Eulerian method considers a velocity potential function, and the fluid's property is determined by pressure or velocity parameters at the element nodes. Nevertheless, it is difficult to display the structure arrangement using this set up. Because the structure arrangement necessitates the use of displacement variables [3, 6].

Lagrangian elements and fluid elements with displacement as an element of fluid variable can be used to overcome the above difficulty. During motion, each node of the computational analysis mesh in the Lagrangian algorithms follows the related material component. These algorithms are often employed in the field of structural mechanics in conjunction with both solid and structural elements. It also simplifies the tracking of unrestricted and material layouts [3, 6].

2.7 Numerical Method

The mass and stiffness matrices are generated from the formulas representing the potential and kinetic energies within the numerical method. The energy stored in the shell due to both stretching and bending can be expressed as:

$$U(t) = \frac{1}{2}\int_0^L (N_z \varepsilon_z + N_\theta \varepsilon_\theta + M_z K_z)\pi \qquad (2.43)$$

Where N_z and N_θ = membrane force generated

M_z = the axial bending moment generated

ε_z = the normal strain along the axial direction

ε_θ = the normal strain in the axial direction

K_z = the mid surface variation in the curvature

Under the assumption of homogeneity, isotropy, and linear elasticity of the shell material, the force and moment outcomes can be represented in relation to the normal stresses on the middle surface and the curvature changes on the middle surface in the following manner:

$$\{\sigma\} = [D]\{\varepsilon\} \tag{2.44}$$

$$\{\sigma\} = \{N_z \;\; N_\theta \;\; M_z\}^T \tag{2.45}$$

$$\{\varepsilon\} = \{\varepsilon_z \;\; \varepsilon_\theta \;\; K_z\}^T \tag{2.46}$$

$$[D] = \frac{Eh}{1-\upsilon^2}\begin{bmatrix} 1 & \upsilon & 0 \\ \upsilon & 1 & 0 \\ 0 & 0 & h^2/_{12} \end{bmatrix} \tag{2.47}$$

The generalized strain vector $\{\varepsilon\}$ can be expressed in relation to the displacement vector $\{d\}$ in the following manner:

$$\{\varepsilon\} = [P]\{d\} \tag{2.48}$$

in which [P] represents the matrix of differential operators that establishes the relationship between strain and displacement,

$$\{d\} = \{u \;\; w\}^T \tag{2.49}$$

By utilizing the aforementioned equations, it is possible to formulate the potential energy as follows:

$$U(t) = \pi R \int_0^L [([P]\{d\})^T [D]([P]\{d\})]dz \tag{2.50}$$

The kinetic energy of the shell, excluding the effects of rotary inertia, can be described as:

$$T(t) = \pi R \int_0^L [m(z)\{\dot{d}\}^T\{\dot{d}\}]dz \tag{2.51}$$

Where m (z) = the shell's mass for each unit area;

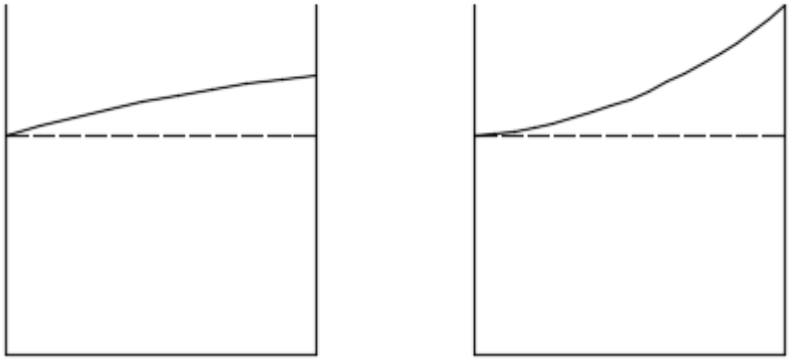
$\{d\}$ = the vector representing displacement

$(\dot{\cdot})$ means differentation with respect to time, t.

2.8 Sloshing in cylindrical tanks

In situations where lateral motion is imposed on a tank, the liquid inside sloshes at the free surface. Previous earthquakes have resulted in damage to the shells of liquid storage tanks. The overturning moment calculated theoretically is significantly smaller compared to the impulsive pressure during small amplitude sloshing. Hence, greater attention has been directed towards assessing the stresses on the shell induced by impulsive pressure and considering the flexibility of the shell [9].

While it's evident that the extended-period seismic motion can induce significant amplitude sloshing, practical computations of sloshing heights have primarily relied on small amplitude theory. Allowing inadequate freeboard for fixed roof systems can result in sloshing motion generating high localized impact pressures on the roof, thereby causing structural damage [16]. Figure 2.7 illustrates two modes linked with liquid sloshing.



(a) Small-scale oscillations (b) Significant-scale oscillations

Figure 2.7 Modes linked with liquid sloshing. [9]

A simplified version of the free surface provides information about the natural frequencies and corresponding modes for moderate amplitude oscillations [9]. The following assumptions are utilized in the linearized sloshing theory. It is assumed that the liquid is homogeneous and incompressible. By assuming that the tank wall is stiff, the convective dynamic pressure may be calculated with fair precision [9].

Hence, the peak convective pressure resulting from the primary sloshing mode, $|P_s(R,\theta,z)|_{max}$, can be calculated as follows:

$$|P_s(R,\theta,z)|_{max} = 0.837\rho_1 R\cos(\theta)\left[\cosh\left(\frac{1.84z}{R}\right) \Big/ \cosh\left(\frac{1.84H}{R}\right)\right]S_{as} \quad (2.52)$$

Where ρ_l represents the density of the liquid

S_{as} represents the spectral pseudo-acceleration value associated with the primary sloshing frequency [20]

The maximum vertical displacement of the free surface, denoted as ε_{max}, occurs at the intersection with the shell and along the axis of excitation. This value is determined by:

$$\varepsilon_{max} = 0.837R\frac{S_{as}}{g} \tag{2.53}$$

To calculate the effective mass, m_s, due to sloshing, the commonly known mechanical model is used. It is given by:

$$m_s = 0.455\pi\rho_1 R^3 \tanh\left(\frac{1.84H}{R}\right) \tag{2.54}$$

At a height of Hs, Medhat A. Haroun supplied the values for this height alongside the related natural frequency across different height to radius ratios [9].

CHAPTER 3 ANALYSIS OF CYLINDRICAL TANK USING ARTIFICIAL NEURAL NETWORKS

Neural networks (NNs) aim to replicate the biological neural structure of the human brain electronically, with the goal of integrating capabilities like learning and pattern recognition. Throughout the entire training process, learning involves generating weight matrices that optimize the matching of expected seismic structural responses with the observed responses [13].

3.1 Introduction

Backpropagation stands as the most adaptable learning algorithm for a layered feedforward network. It redistributes or back-propagates the errors in the network's output by adjusting the weight matrices accordingly. This algorithm is applied in categorizing structural responses caused by different earthquakes due to its rapid learning rate. For simplicity, a backpropagation model with one input, one hidden, and one output layer is depicted in Figure 3.1.

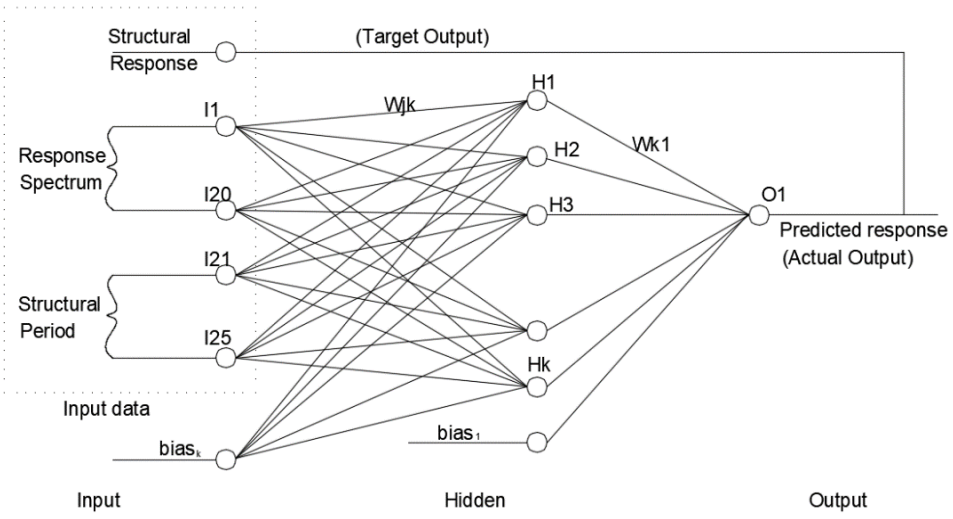


Figure 3.1 Architecture of NN [13]

3.2 Neural Network Fitting

During the training phase, the input data for each pattern includes details regarding the seismic and structural characteristics at the input units of the neural network, alongside the desired categorization of the response at the output units of the network.

The activation of each input unit, denoted as I_j, consistently corresponds to the input values converted from the feasible structural or seismic behaviors. The hidden layer encapsulates the characteristics of the relationship between the input data and the output categorization. The activation of each hidden unit is expressed as:

$$H_k = f(bias_k) + \sum_{j=1}^{J} W_{jk}.I_j \tag{3.1}$$

Where k ranges from 1 to K

In this equation, I and H denote the activation levels of neurons in the input and hidden layers, respectively. The symbols J and K indicate the number of neurons, with subscripts j and k representing the neurons in the input and hidden layers, respectively. The k^{th} bias, denoted as $bias_k$, can be viewed as a weight originating from an always active unit, while W_{jk} is an element of the weight matrix connecting units in the input and hidden layers. Meanwhile, *f* represents the activation function of a hidden layer unit, given by:

$$f(net) = \frac{1}{1 + e^{-net}} \tag{3.2}$$

Equivalent relationships apply to the network's output layer, resulting in equation (4.1) transforming to:

$$O_l = f(bias_l) + \sum_{k=1}^{K} W_{kl}.H_k \tag{3.3}$$

Where O_l represents the activation of the single unit utilized in the output layer to offer a distinct description of the structural response. W_{kl} serves as a connection between this unit and a unit in the hidden layer. The predicted categorization, O_l, derived from equation (4.3), is compared with the target categorization of the structural response. In cases where the two categorizations diverge, the network is trained by adjusting weights and biases to minimize the disparity.

In the (n + 1)th iteration, the values of W_{kl} and bias from equation (3) are determined based on their values in the previous epoch by employing:

$$W_{kl}(n+1) = W_{kl}(n) + \Delta W_{kl}(n+1) \tag{3.4}$$

$$bias_l(n+1) = bias_l(n) + \Delta\, bias_l(n+1) \tag{3.5}$$

$$\Delta W_{kl}(n+1) = lrate.\,\delta_l.\,H_k + \mu.\,\Delta W_{kl}(n) \tag{3.6}$$

$$\Delta bias_l(n+1) = lrate.\,\delta_l + \mu.\,\Delta bias_l(n) \tag{3.7}$$

$$\delta_l = (t_l - O_l).\,O_l.\,(1 - O_l) \tag{3.8}$$

Where lrate is rate of learning

μ is a constant coefficient that governs the impact of past weight adjustments on the current direction of movement within the weight space.

δ the link between the input and hidden layers

Subscript k is simply substituted by j for a connection between the input and hidden layers, l by k, and δ_k is given by:

$$\delta_k = H_k.\,(1 - H_k).\,W_{kl}.\,\delta_l \tag{3.9}$$

As the discrepancy between the target value (t) and the output value (Ol) decreases, the network progressively learns. However, similar to other gradient descent techniques, there's no assurance of reaching a global minimum. However, a neural network with numerous hidden units and weights may traverse different dimensions to navigate from various local minima.

3.3 Procedural steps to develop NN models

The MATLAB software was fed neural network training data which were obtained via finite element software analysis and Housner method of analysis. Housner's simplified approach is a validated analytical method supported by experimental evidence [16, 18]. Around 768 training samples were deployed in the first round of training the neural

network. Then 6,912 different sets of samples were deployed in the final round of neural network training.

The corresponding input and output samples are subdivided into three kinds of samples during neural network fitting:

- **Training Samples**:- During the training process, these inputs are provided to the network, and the network is modified based on its error.
- **Validation Samples**:- These samples are utilized to assess the network's ability to generalize and to cease training when further improvements in generalization cease.
- **Testing Samples**:- These samples do not influence the training process; hence, they offer an impartial evaluation of performance during and after training.

Training data is a set of observations. These observations constitute the conditions from which the program learns. Every observation in a supervised learning task is comprised of an observed response variable, in addition to that it features one or more observed explanatory variables. The test data consists of a similar set of observations. The test set is utilized to assess the model's performance using some performance criterion. It is critical that no observations from the training data show up within the test data. If the test data contains examples from the training data, investigation of the performance of the algorithm will be challenging because it will be difficult to know if it has learned to generalize or if it has memorized from the training data.

A program that demonstrates successful generalization will be able to effectively perform a task with new data. Conversely, an algorithm that memorizes the training data by acquiring an excessively intricate model can accurately predict the values of the response variable for the training data. However, it struggles to forecast the value of the response variable for new cases. Overfitting is the process of memorizing the training data. A program that memorizes its observations might be unable to achieve its objective due to the fact that it might recall relations and structures that are random in the training data. Many machine learning algorithms struggle with balancing generalization and memorization.

A third set of observations, referred to as validation data, is required in addition to the training and test data. The validation data is used to adjust variables known as hyperparameters, which dictate how the algorithm learns from training data. The program is assessed on the test data to give an estimation of its actual performance. Since the program has been adapted to learn from the training data in a way that optimizes its result on the validation data, the validation data should not be used to estimate actual performance.

A single set of supervised observations is commonly divided into training, validation, and test sets. There are no criteria for partition sizes, and they may vary depending on the amount of data available. It is commonly accepted to distribute between 50 and 75 percent of the data to the training set, 10 to 25 percent to the test set, and the remaining to the validation set.

The following procedural steps has to be followed to develop, validate and test the neural network models as shown in the chart below:

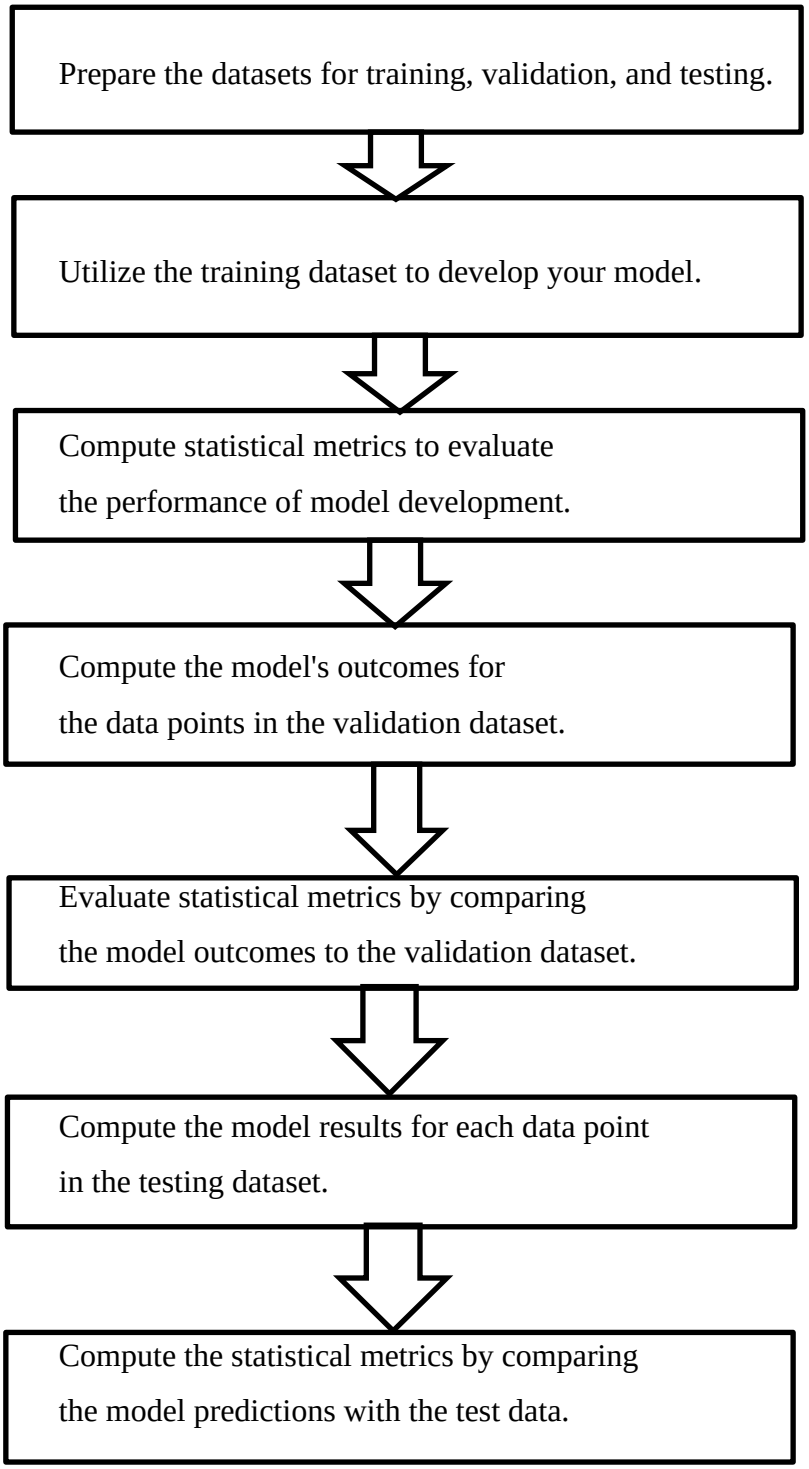


Figure 3.2 Steps for developing, validating, and testing models

In this research, where numerous models are being considered, the conventional approach is to choose the model that exhibits the closest fit to the validation data and proceed accordingly. Nonetheless, there is another potential complication to consider. Just because a model closely aligns with the validation data does not necessarily mean it accurately reflects reality. Even if a model performs exceptionally well in a specific test, it could still be incorrect in its representation of the problem [15].

The last and crucial step involves comparing the top-performing model from the validation phase with the test data. This test dataset, again, constitutes a subset of the original data source and includes points that were not utilized in training or validation. A model is deemed suitable for application when its statistical evaluations closely match those of the test data.

Table 3.1 Input and Output elements deployed for the first NN fitting

Input Elements	**Output Elements**
Height of Cylindrical Tank	Sloshing Frequency
Radius of Cylindrical Tank	Impulsive Force
Density of Fluid	Convective Force
Peak Ground Acceleration	Base Shear
Damping Correction factor	Bending Moment
Soil Factor	Overturning Moment

Table 3.2 Input and Output elements deployed for the second NN fitting

Input Elements	**Output Elements**
Height of Cylindrical Tank	Maximum Vertical Displacement of Free Surface
Radius of Cylindrical Tank	
Density of Fluid	Effective Mass due to Sloshing
Peak Ground Acceleration	
Damping Correction factor	Maximum Convective Pressure due to Sloshing mode
Soil Factor	

Out of the available training algorithms for neural network fitting in MATLAB software Levenberg-Marquradt, Bayesian Regularization, and Scaled Conjugate Gradient; the first one was chosen due to the fact that it requires less time and training automatically stops when generalization stops improving.

The six input layers were selected to represent the geometric and mechanical attributes of the cylindrical tank as well as the fluid it holds. Whereas the six output layers were

designated for the reason that they represent the dynamic response of the fluid containing cylindrical tank.

The height of the cylindrical tanks ranges from 1.5 m to 9 m, while the radius of the cylindrical tanks ranges from 6 m to 22.5 m. Cylindrical tanks within these ranges of dimensions are applicable in domestic as well as industrial areas. Water and oil were used to study the effect of the density of fluid to the dynamic response of the cylindrical tank. 10% of g, 25% of g and 50% of g peak ground acclerations were used to train the neural network. The corresponding damping correction factors for 5%, 10% and 15% damping rations were used to train the neural network. Ground type A and Ground type B conditions were deployed for the neural network fitting. The following tables show the first 32 sets of samples of the 6,912 different sets of samples that were deployed in the final round of neural network training.

Table 3.3 Input Data Deployed for the First NN Training

No.	H (m)	R (m)	ρ (Kg/m^3)	A_0 (m/s^2)	η	S
1	1.5	6	920	0.1g	1	1
2	1.5	9	1,000	0.25g	0.7071	1
3	2	7.5	920	0.5g	0.7071	1.2
4	2	10.5	1,000	0.1g	0.8165	1.2
5	2.5	9	920	0.25g	0.8165	1
6	2.5	12	1,000	0.5g	1	1.2
7	3	10.5	920	0.1g	1	1
8	3	13.5	1,000	0.25g	0.7071	1.2
9	3.5	12	920	0.5g	0.7071	1
10	3.5	15	1,000	0.1g	0.8165	1
11	4	13.5	920	0.25g	0.8165	1.2
12	4	16.5	1,000	0.5g	1	1.2
13	4.5	15	920	0.1g	1	1
14	4.5	18	1,000	0.25g	0.7071	1.2
15	5	16.5	920	0.5g	0.7071	1
16	5	19.5	1,000	0.1g	0.8165	1.2
17	5.5	18	920	0.25g	0.8165	1
18	5.5	21	1,000	0.5g	1	1
19	6	19.5	920	0.1g	1	1.2
20	6	22.5	1,000	0.25g	0.7071	1.2
21	6.5	21	920	0.5g	0.7071	1
22	6.5	6	1,000	0.1g	0.8165	1.2
23	7	22.5	920	0.25g	0.8165	1
24	7	7.5	1,000	0.5g	1	1.2

25	7.5	6	920	0.1g	1	1
26	7.5	9	1,000	0.25g	0.7071	1
27	8	7.5	920	0.5g	0.7071	1.2
28	8	10.5	1,000	0.1g	0.8165	1.2
29	8.5	9	920	0.25g	0.8165	1
30	8.5	12	1,000	0.5g	1	1.2
31	9	10.5	920	0.1g	1	1
32	9	13.5	1,000	0.25g	0.7071	1.2

Where H is fluid height, R is radius of the cylindrical tank, ρ is the density of the fluid, A_0 is the maximum acceleration of the ground, η is the damping correction factor, and S is the soil factor.

Table 3.4 Output Data Deployed for the First NN Training

No.	ω (rad/s)	P_i (KN)	P_s (KN)	BS (KN)	BM (KNm)	OTM (KNm)
1	1.1375	22.0993	3.8847	25.9840	15.3947	140.5080
2	0.7723	90.0789	8.0307	98.11	56.7392	818.6392
3	1.0462	245.5474	28.7897	274.3371	213.5143	1,794.9792
4	0.7609	74.7322	7.7699	82.5020	63.8975	793.9214
5	0.9717	230.2000	25.6567	255.8567	248.5638	1,992.5705
6	0.6688	427.0409	48.4573	475.4982	369.1142	5,404.6995
7	0.9104	154.6939	14.9484	169.6423	196.9558	1,528.1756
8	0.7198	540.4735	36.8160	577.289	664.0134	6,821.1491
9	0.8589	1,203.1711	71.4444	1,274.6154	1,707.1065	1,2941.6275
10	0.6979	326.9530	19.0854	346.038	463.0290	4,493.1176
11	0.8151	883.9599	64.3035	948.2634	1,457.6403	10,856.7627
12	0.6768	2,348.7221	181.5479	2,530.2700	3,892.0819	36,323.3413
13	0.7772	497.2262	33.0801	530.3063	915.3334	6,725.0882
14	0.6567	1491.7045	74.1652	1,565.8697	2,687.0046	24,122.2724
15	0.7441	3,376.2190	143.8696	3,520.0886	6,699.1221	48,723.9600
16	0.6379	867.4245	45.6642	913.0887	1742.6530	15,221.3033
17	0.7149	2,228.2998	100.2047	2328.504	4,878.4674	35,158.7513
18	0.6204	5,651.5993	280.3320	5,931.9314	12,441.9175	106,136.2627
19	0.6888	1,149.1375	69.9252	1,219.0627	2,800.7643	20,013.8547
20	0.6040	3,603.1413	140.8193	3,743.9607	8,537.8024	71,344.1909
21	0.6653	7,261.8975	241.2494	7,503.1469	18,505.4631	131,605.2641
22	1.7026	415.6492	20.7659	436.4150	1,096.6468	2,108.5618
23	0.6441	4,511.8225	161.2294	4,673.052	12,422.7582	87,844.9656
24	1.5021	3,113.5305	188.1104	3,301.6410	8,956.4593	19,582.6852
25	1.7171	487.4060	20.1737	507.5798	1,468.3374	2,524.8149
26	1.3517	2,182.5482	75.3520	2,257.9002	6,465.8341	16,002.2015

27	1.5210	3,634.7988	128.6507	3,763.4495	11,538.3984	22,669.1610
28	1.2340	1,170.6247	53.7414	1,224.3661	3,756.2747	10,109.0328
29	1.3730	2,528.6337	85.2225	2,613.856	8,492.2791	18,639.9118
30	1.1391	7,598.3177	407.6334	8,005.9511	26,161.7694	75,604.0532
31	1.2563	1,344.1757	54.2196	1,398.3953	4,821.1048	11,567.3640
32	1.0608	4,810.6848	173.6399	4,984.3247	17,102.6249	53,150.0426

Where ω is natural circular frequency of the fluid, P_i is impulsive force, P_s is convective force, BS is base shear, BM is bending moment, and OTM is overturning moment.

Table 3.5 Statistical Parameters of the Scaled Input-Patterns of the Dataset used for the First NN Training

Parameter	Mean	Standard deviation	Lower Extreme	Upper Extreme
H (m)	5.25	2.3049	1.5	9
R (m)	14.25	5.1781	6	22.5
ρ (Kg/m^3)	960	40	920	1,000
A_0 (m/s^2)	0.2833g	0.165g	0.1g	0.5g
η	0.8412	0.1208	0.7071	1
S	1.1	0.1	1	1.2

Table 3.6 Input and Output Data Deployed for the Second NN Training

No.	H (m)	R (m)	ρ (Kg/m^3)	A_0 (m/s^2)	η	S	ϵ_{max} (cm)	M_s (Kg)	$P_{s,\,max}$ (N/m^2)
1	1.5	6	920	4.905	0.8165	1.2	20.16	132,790.96	1,819.33
2	2	12	1,000	4.9050	1	1.2	17.07	73,4595.60	1,674.77
3	3	15	1,000	4.9050	1	1.2	20.22	16,99319.61	1,983.59
4	4	15	1,000	4.9050	1	1.2	26.10	2,193,831.96	2,560.83
5	5	18	1,000	4.9050	1	1.2	27.03	3,924,866.57	2,651.30
6	6	21	1,000	4.9050	1	1.2	27.67	6,381,969.39	2,714.87
7	7	6	1,000	4.905	0.8165	1.2	45.61	300,434.60	4,474.08
8	8	9	1,000	4.905	0.8165	1.2	43.44	965,823.21	4,261.65
9	2	9	920	2.4525	0.8165	1.2	9.08	403,824.65	819.65
10	3	6	920	2.4525	0.8165	1.2	17.01	224,124.98	1535.33
11	4	18	920	2.4525	0.8165	1.2	9.08	3,230,597.21	819.65
12	5	18	920	2.4525	0.8165	1.2	11.03	3,924,866.57	995.80
13	6	6	920	2.4525	0.8165	1.2	22.28	293,563.44	2,011.01
14	7	21	920	2.4525	0.8165	1.2	12.81	7,234,106.38	1,155.83
15	8	9	920	2.4525	0.8165	1.2	21.72	965,823.21	1,960.36
16	2	18	920	2.4525	0.8165	1	3.94	1,680,976.39	355.41
17	3	15	920	2.4525	0.8165	1	6.88	1,699,319.61	620.85
18	4	6	920	2.4525	0.8165	1	16.44	259,851.83	1,483.39
19	5	21	1,000	2.4525	1	1	9.86	5,454,867.31	966.87
20	6	15	1,000	2.4525	1	1	14.99	3,023,501.89	1,470.54
21	7	12	1,000	2.4525	1	1	18.91	1,953,099.20	1,855.33

22	8	18	1,000	2.4525	1	1	16.12	5,617,557.49	1,581.14
23	1.5	6	1,000	4.905	1	1	20.57	132,790.96	2,018.30
24	2	9	920	2.4525	1	1	9.27	403,824.65	836.55

Where ϵ_{max} is the maximum vertical displacement of the free surface, M_s is effective mass due to sloshing, and $P_{s,\,max}$ is maximum convective pressure due to sloshing mode

Table 3.7 Statistical Parameters of the Scaled Input-Patterns of the Dataset used for the Second NN Training

Parameter	Mean	Standard deviation	Lower Extreme	Upper Extreme
H (m)	4.5625	2.20	1.5	8
R (m)	13.5	5.1235	6	21
ρ (Kg/m^3)	960	40	920	1,000
A_0 (m/s^2)	0.375g	0.125g	0.25g	0.5g
η	0.90825	0.09175	0.8165	1
S	1.1	0.1	1	1.2

The following steps has to be followed to make dynamic response prediction of the fluid containing cylindrical tanks as shown in the chart below:

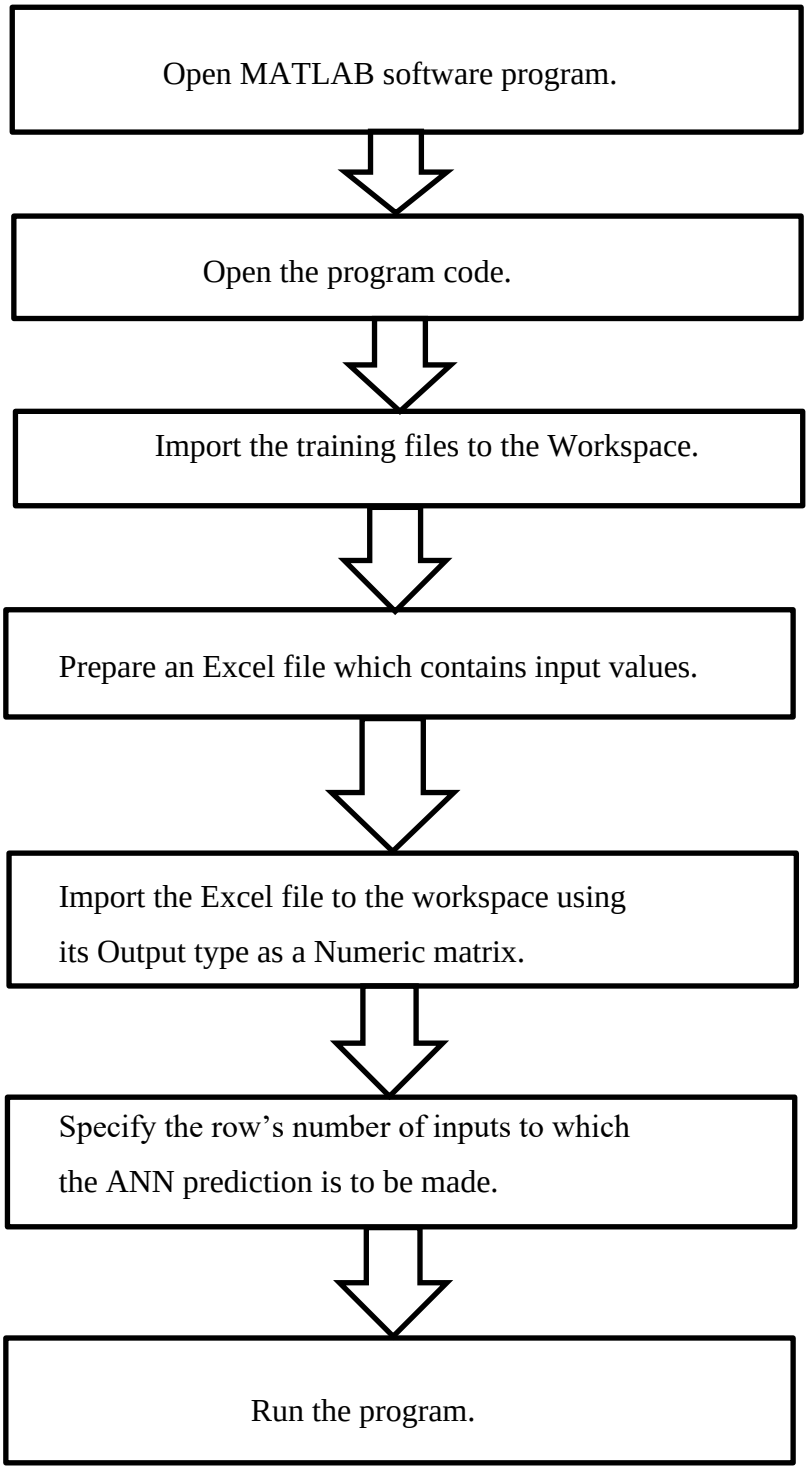


Figure 3.3 Steps for utilizing the program code to predict the desired output element.

The program codes are provided in the Appendix section of this paper.

3.4 Verification of Neural Network Fitting Software

MATLAB, a programming and data analyzing software, is used to train neural networks for the prediction of the dynamic response of fluid containing cylindrical tanks. The accuracy and precision of the neural network fitting software are investigated in this section.

M. A. El-Gohary employed MATLAB software to simulate a mathematical model representing a quarter car suspension system. The accuracy of the results was verified through experimentation using a test rig. The artificial neural network (ANN) comprises five inputs, including vibration dose value (VDV), root mean square (RMS) acceleration, maximum transient vibration value (MTVV), RMS suspension displacement, and RMS tire deflection. Additionally, the model features two outputs, namely spring stiffness and damping coefficient [17].

Table 3.8 Correlation of MATLAB Predicted results with experimental results [17]

	Experimental Value	**MATLAB Prediction**	**% Correlation**
VDV ($m/s^{1.75}$)	1.5673	1.3331	85.06%
Aw (m/s^2)	0.5695	0.542125	95.19%
MTVV (m/s^2)	2.2782	2.056311	90.26%

An artificial neural network (ANN) is utilized to forecast the optimal suspension parameters employing the nftool function within MATLAB. The ANN outcomes exhibit a correlation ranging from 85% to 95% with experimental findings.

CHAPTER 4 RESULTS AND DISCUSSION

4.1 General

Analysis of the cylindrical tank was conducted as shown in the third chapter. Various input values were applied during the execution of the analysis process.

Table 4.1 shows the material properties used in the system. The water temperature is at 20 ^{0}C, and these material properties were kept constant throughout the analysis.

The properties of materials used for the system are shown in Table 4.1

Table 4.1 Material Properties

Material Property	
Water	
Density	1000 kg/m^3
Bulk modulus	2.0684e9 Pascal
Viscosity	1.13e-3 N.S/m^2
Oil	
Density	920 kg/m^3
Bulk modulus	1.60e9 Pascal
Viscosity	3.11e1 N.S/m^2
Steel	
Density	7850 kg/m^3
Poisson ratio	0.3
Modulus of elasticity	20.67e10 Pascal

Then these results were used as output elements to train a neural network as discussed in the previous chapter. After that investigations were conducted on the neural network's capacity to predict the dynamic response of the cylindrical tanks by varying the proportion of sample sets, number of hidden layers, and retraining of neural networks. Following that the trained artificial neural network's accuracy and precision were compared with the previously obtained results of Housner's approximate method as well as codes were provided.

4.2 Effect of Variation of Sample Set Proportions

The dataset used for the actual training process is termed as the training data. Through fine-tuning the model after each epoch, the validation data assists in enhancing model

performance. The test dataset provides insights into the model's ultimate accuracy achieved during the training phase.

The commonly recommended split for train-validation-test datasets is 60-80% for training data, 10-20% for validation data, and 10-20% for test data.

4.2.1 70% Training data, 15% Validation data and 15% Testing data

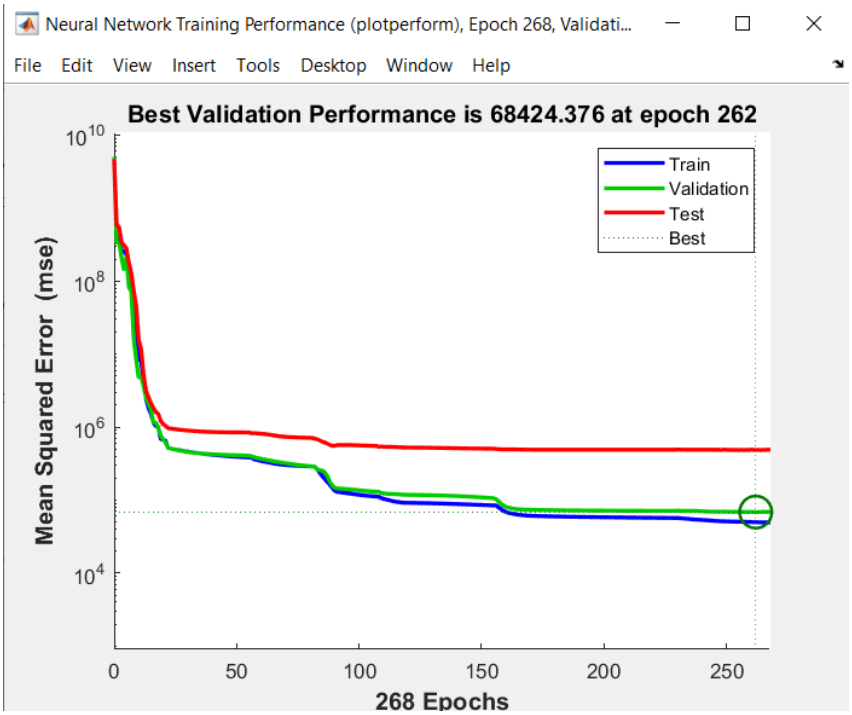


Figure 4.1 Neural Network Training Performance

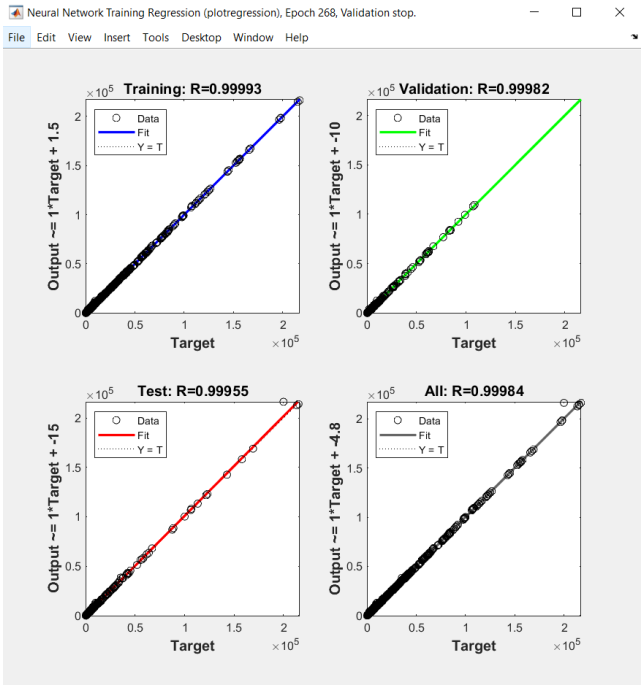


Figure 4.2 Neural Network Training Regression

4.2.2 75% Training data, 10% Validation data and 15% Testing data

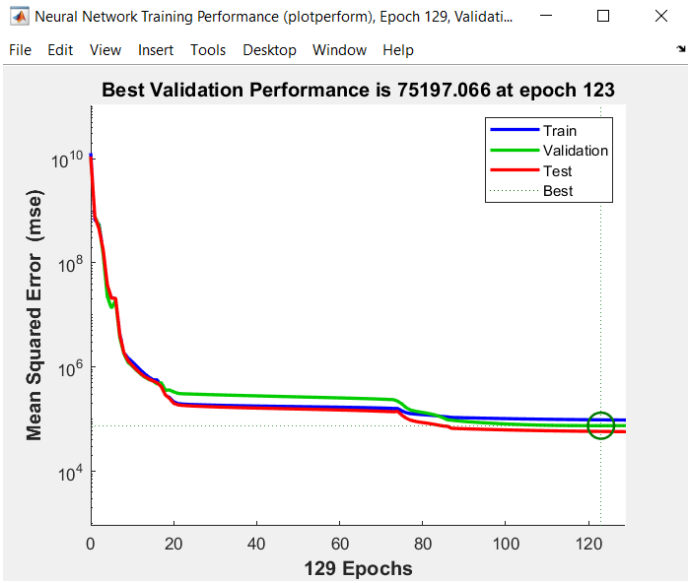


Figure 4.3 Neural Network Training Performance

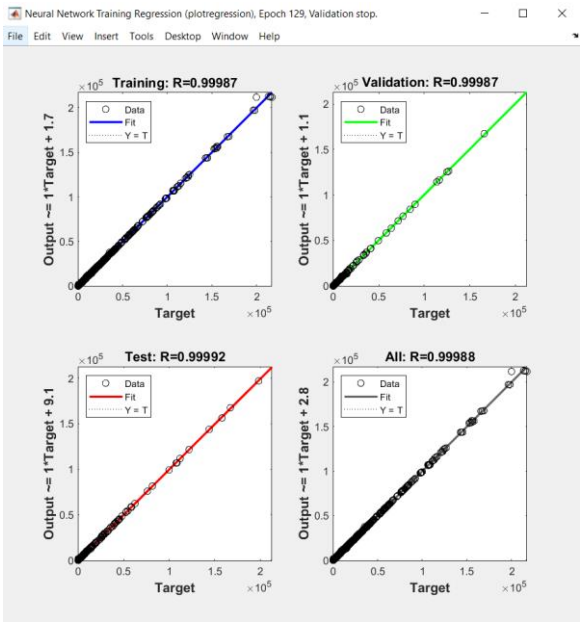


Figure 4.4 Neural Network Training Regression

4.2.3 75% Training data, 15% Validation data and 10% Testing data

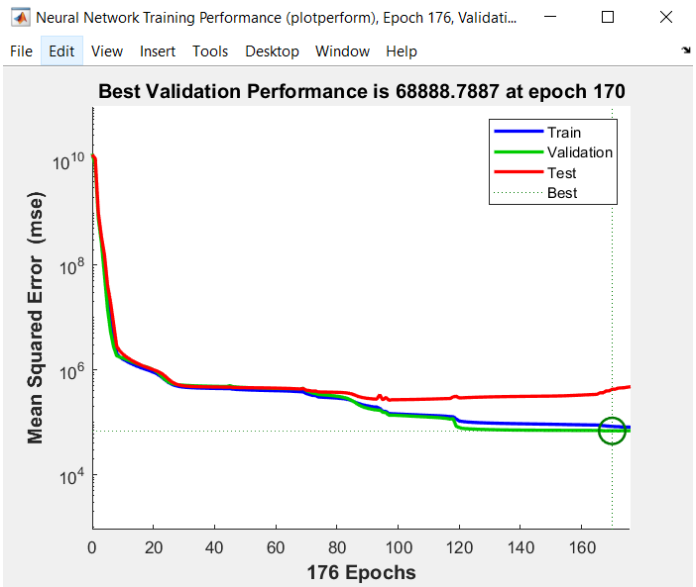


Figure 4.5 Neural Network Training Performance

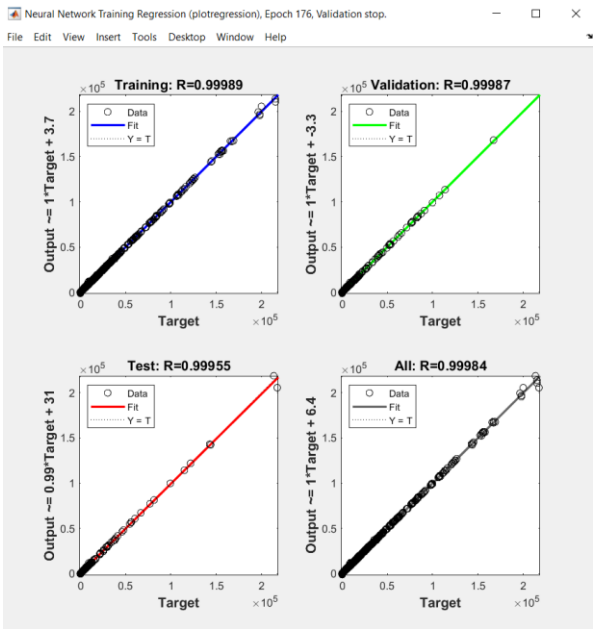


Figure 4.6 Neural Network Training Regression

4.2.4 80% Training data, 10% Validation data and 10% Testing data

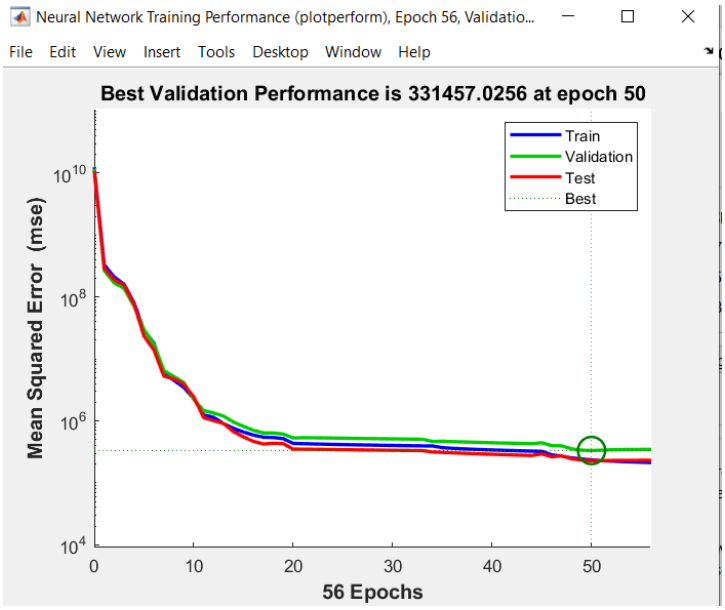


Figure 4.7 Neural Network Training Performance

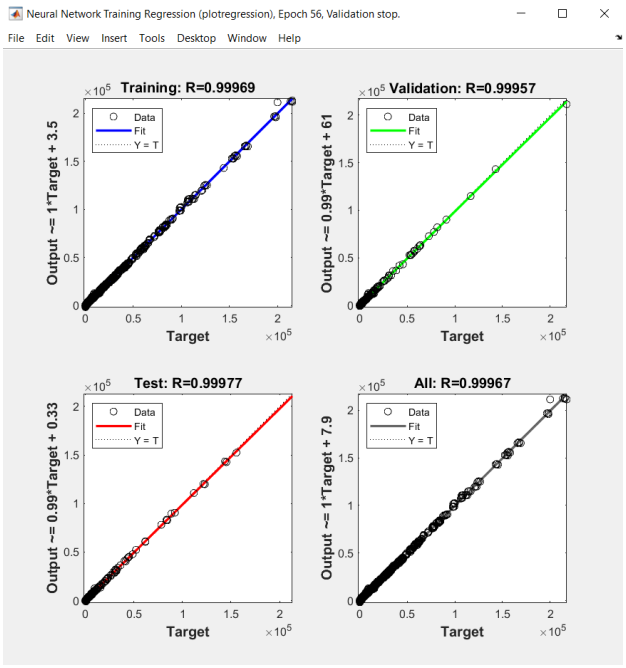


Figure 4.8 Neural Network Training Regression

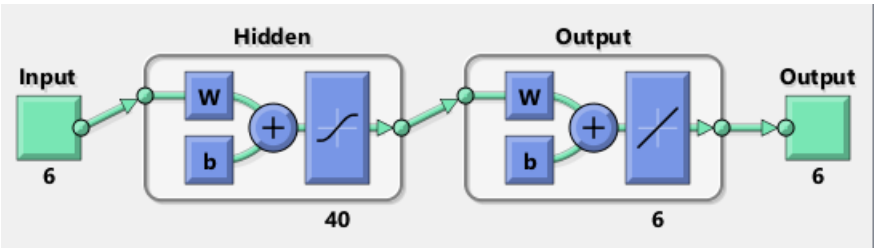


Figure 4.9 Neural Network Diagram

4.3 Effect of the Number of Hidden Layers Deployed

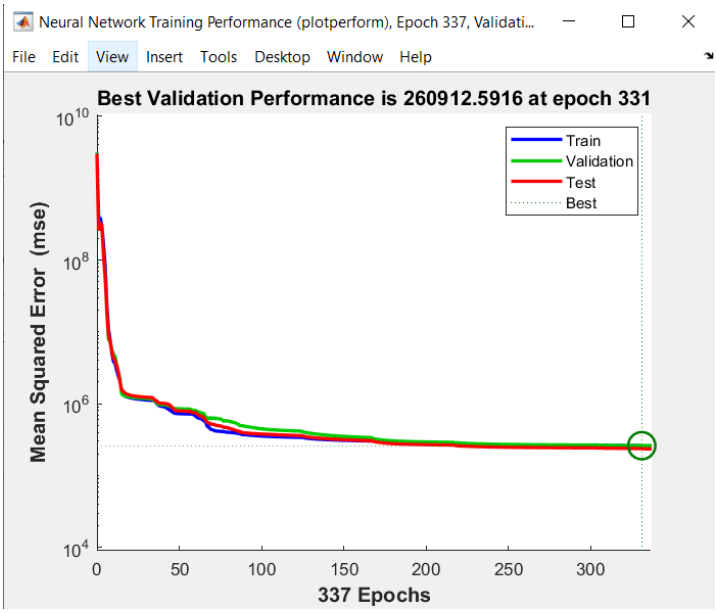


Figure 4.10 Neural Network Performance using 5 Hidden Layers

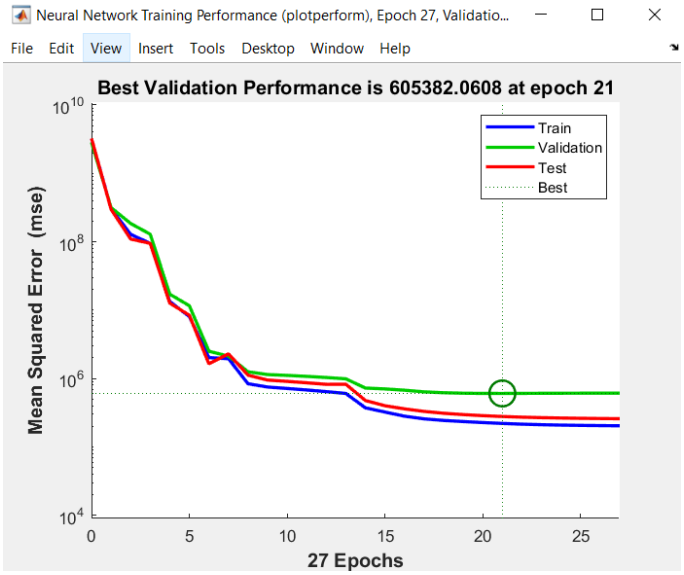


Figure 4.11 Neural Network Performance using 10 Hidden Layers

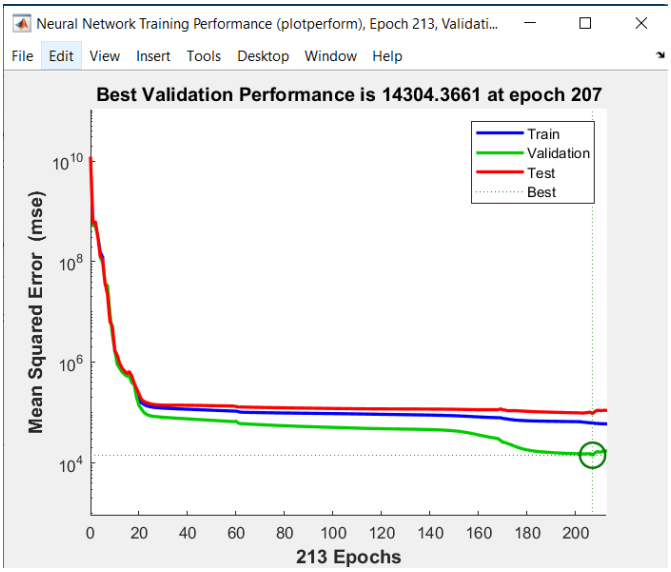


Figure 4.12 Neural Network Performance using 15 Hidden Layers

4.4 Effect of Retraining Neural Networks

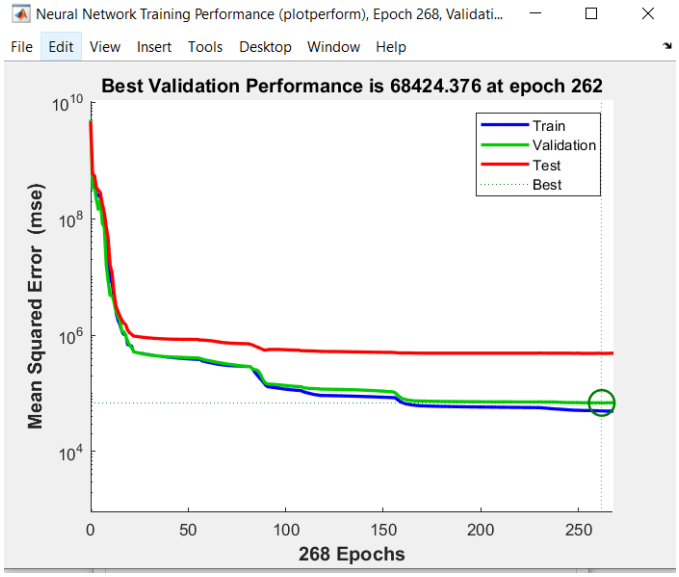


Figure 4.13 Neural Network Performance in the First Training

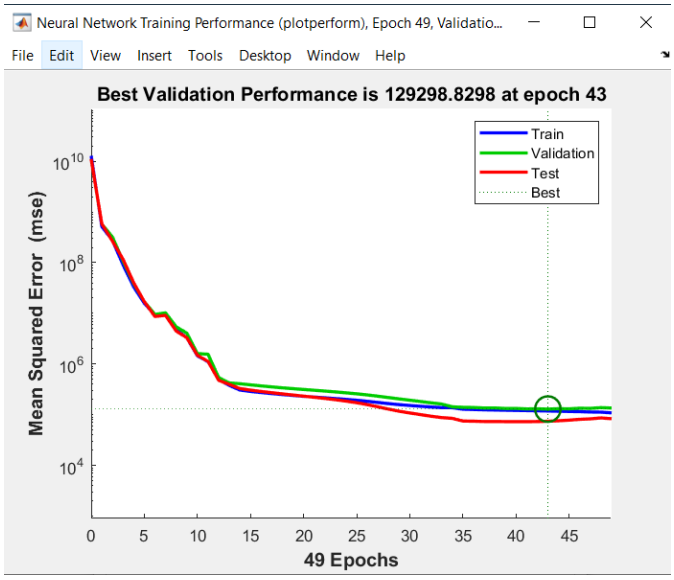


Figure 4.14 Neural Network Performance in the Second Training

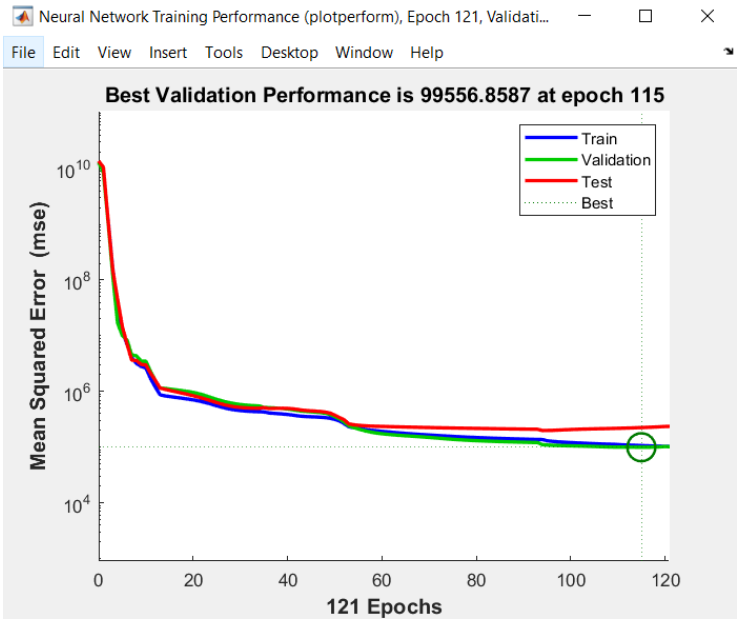


Figure 4.15 Neural Network Performance in the Third Training

4.5 Comparison of Output results with varying Input values

4.5.1 Comparison of the Natural Frequency of Sloshing

Figures 4.20 and 4.21 depict the natural frequencies of sloshing for various height to radius ratios of the contained liquid, with a comparison made to the results obtained from an artificial neural network (ANN).

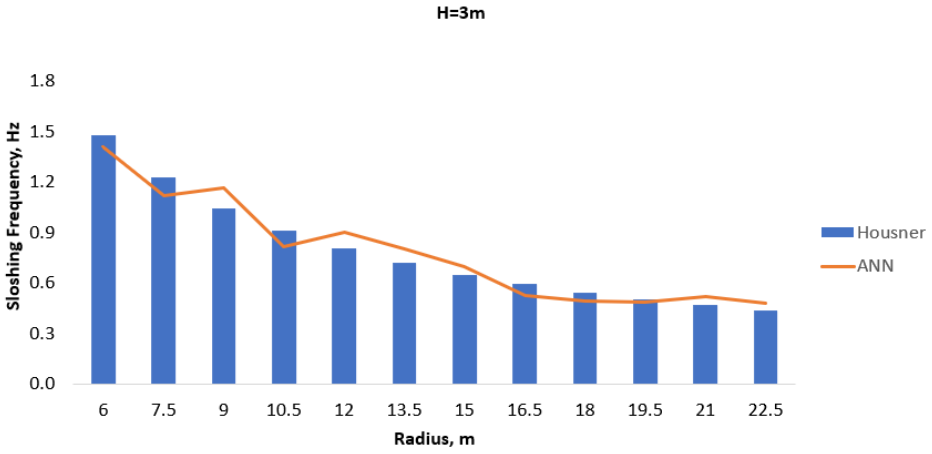


Figure 4.16 Sloshing frequencies for the constant height of three meters and variable radius

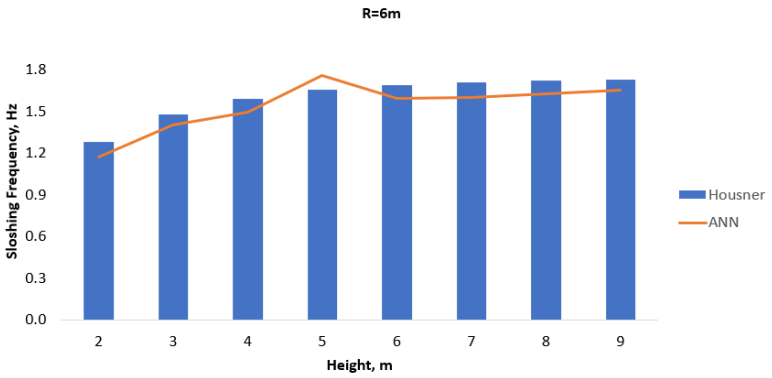


Figure 4.17 Sloshing frequencies for the constant radius of six meters and variable height

The visual outcomes illustrated above indicate that the ANN is capable of producing sloshing modes that closely align with the results obtained from the Housner approximation.

4.5.2 Comparison of the Impulsive Force acting on the Tank Wall

The Impulsive Forces resulting from the Housner approximation method as well as the artificial neural network method were compared graphically.

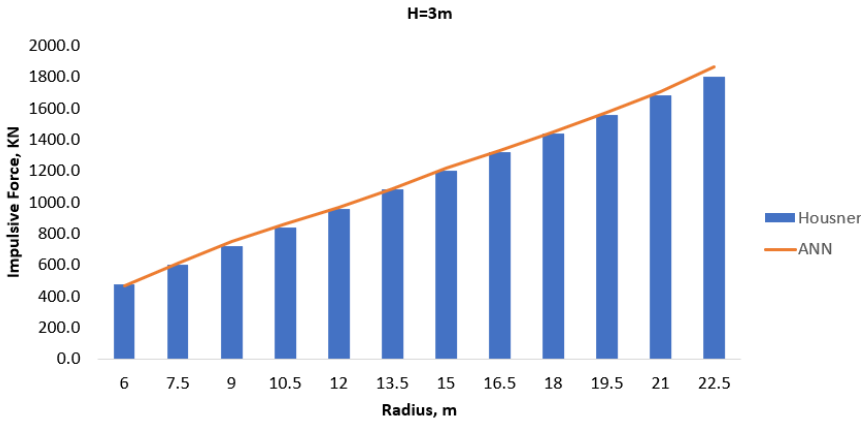


Figure 4.18 Impulsive Force for the constant height of three meters and variable radius

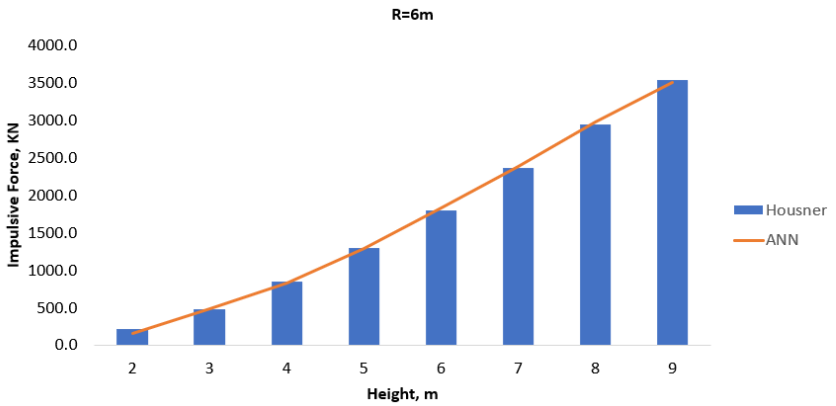


Figure 4.19 Impulsive Force for the constant radius of six meters and variable height

4.5.3 Comparison of the Convective Forces acting on the Tank Wall

The Convective Forces resulting from the Housner approximation method as well as the artificial neural network method were compared graphically.

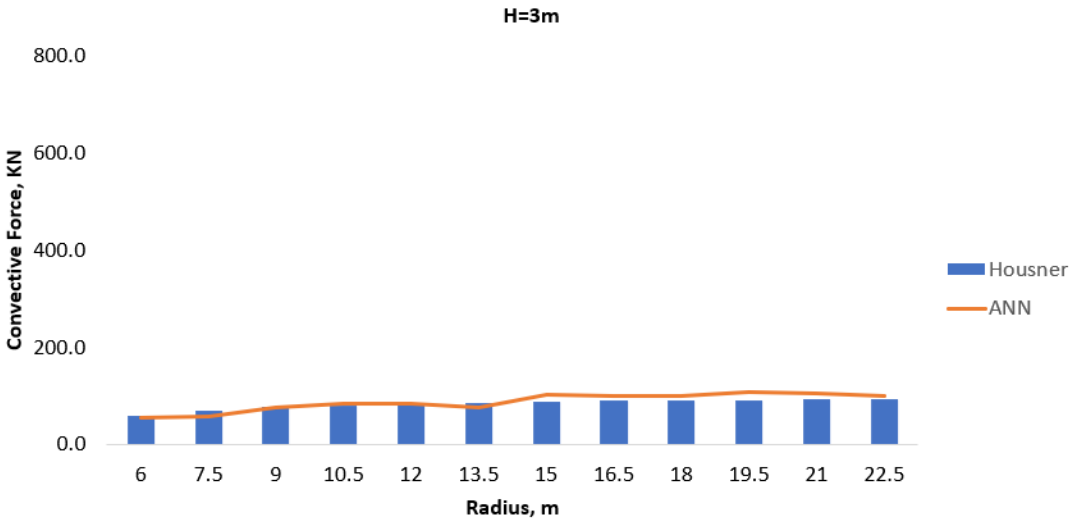


Figure 4.20 Convective Force for the constant height of three meters and variable radius

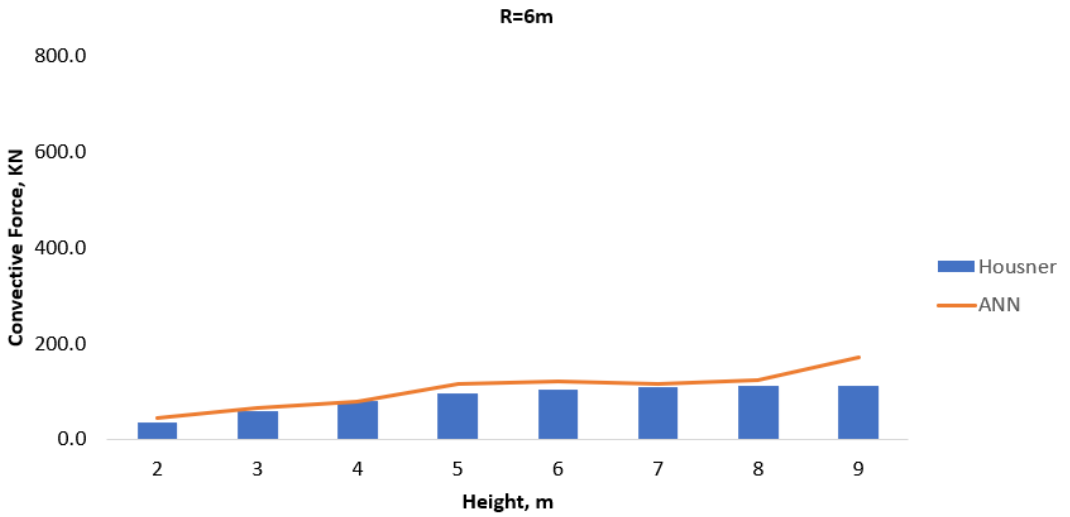


Figure 4.21 Convective Force for the constant radius of six meters and variable height

4.5.4 Comparison of the Base Shear on the Tank Wall

The Base Shear resulting from the Housner approximation method as well as the artificial neural network method were compared graphically.

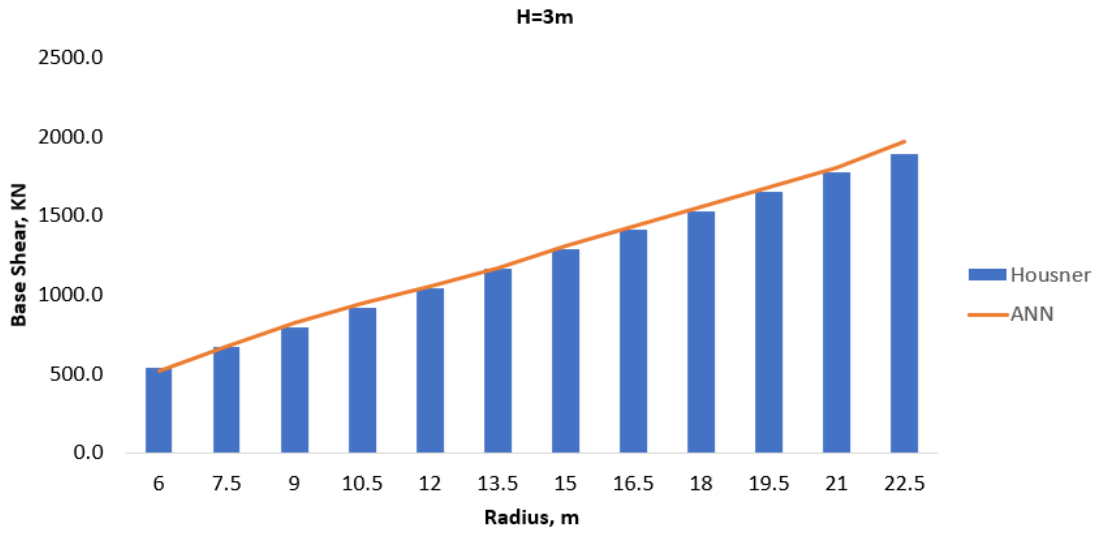


Figure 4.22 Base Shear for the constant height of three meters and variable radius

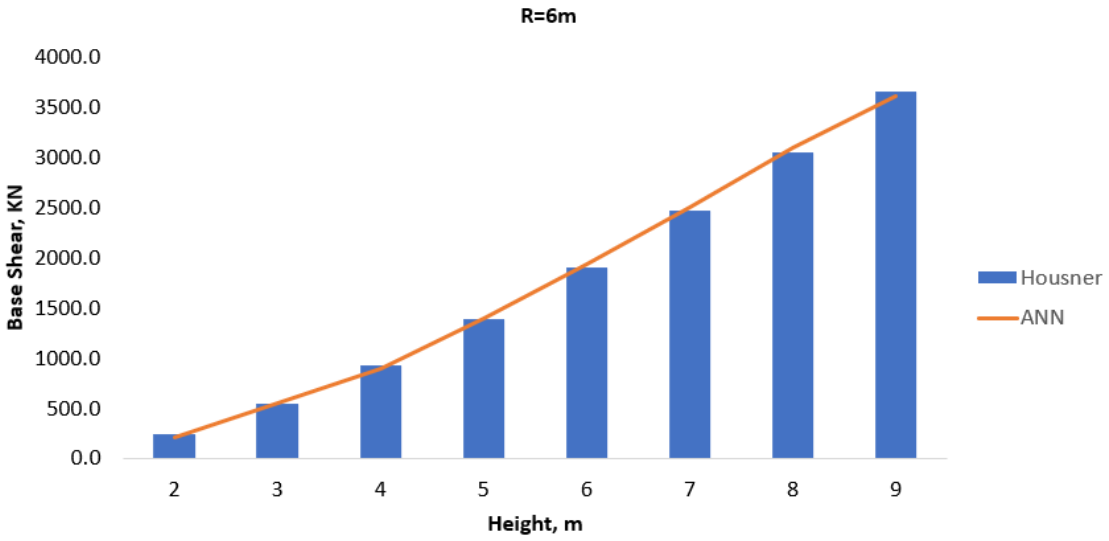


Figure 4.23 Base Shear for the constant radius of six meters and variable height

4.5.5 Comparison of the Bending Moment on the Tank Wall

The Bending Moment resulting from the Housner approximation method as well as the artificial neural network method were compared graphically.

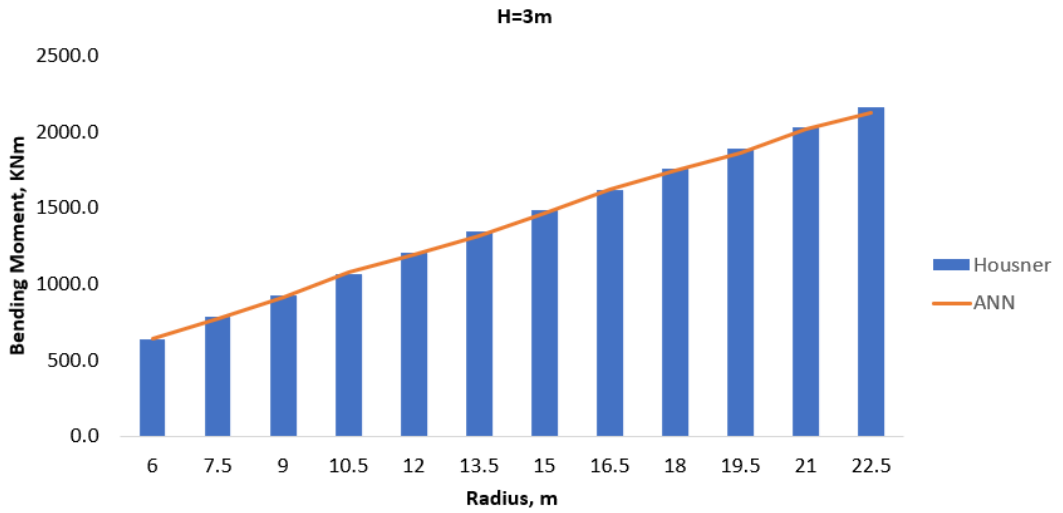


Figure 4.24 Bending Moment for the constant height of three meters and variable radius

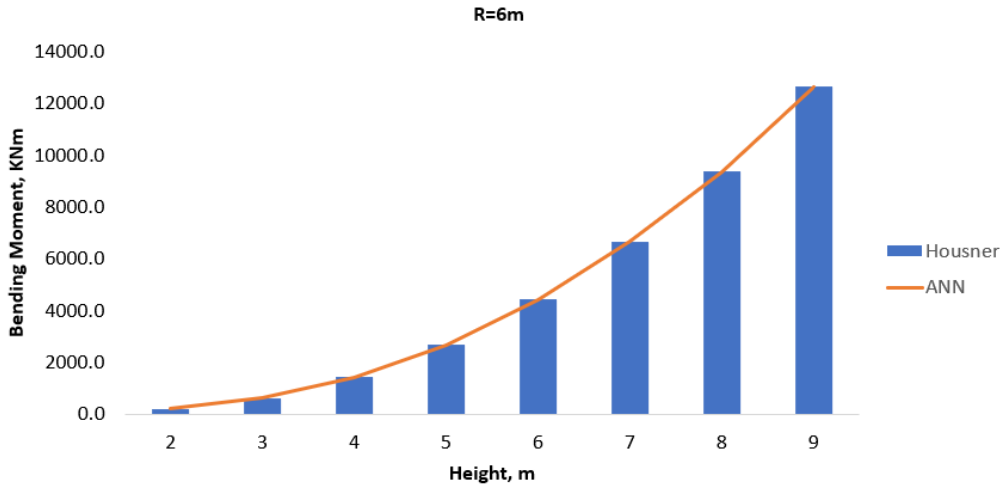


Figure 4.25 Bending Moment for the constant radius of six meters and variable height

4.5.6 Comparison of the Overturning Moment

The Overturning Moment resulting from the Housner approximation method as well as the artificial neural network method were compared graphically.

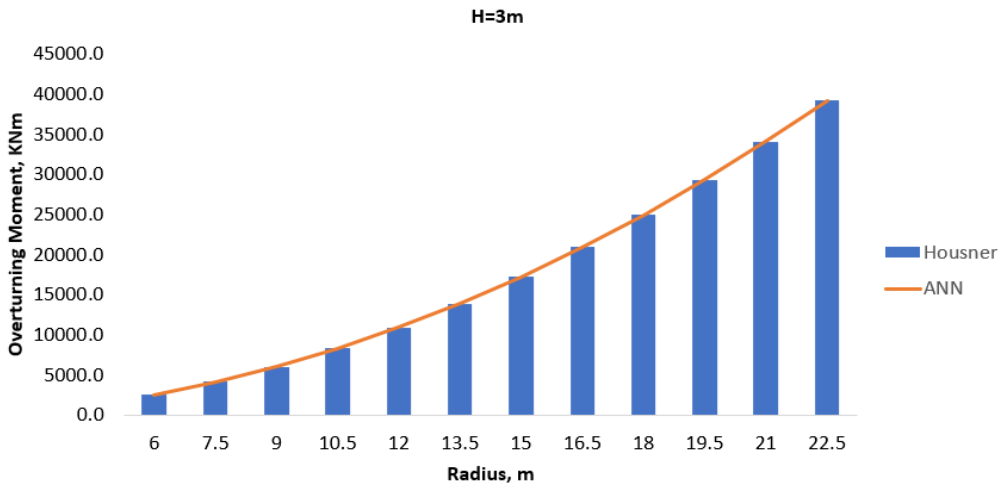


Figure 4.26 Overturning Moment for the constant height of three meters and variable radius

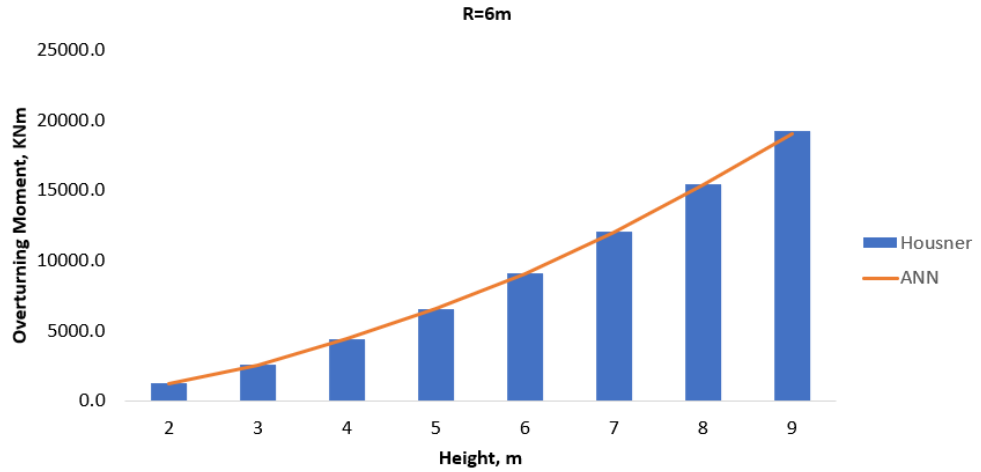


Figure 4.27 Overturning Moment for the constant radius of six meters and variable height

4.5.7 Comparison of the Maximum Vertical Displacement of Free Surface

The Maximum Vertical Displacement of Free Surface resulting from the Housner approximation method as well as the artificial neural network method were compared graphically.

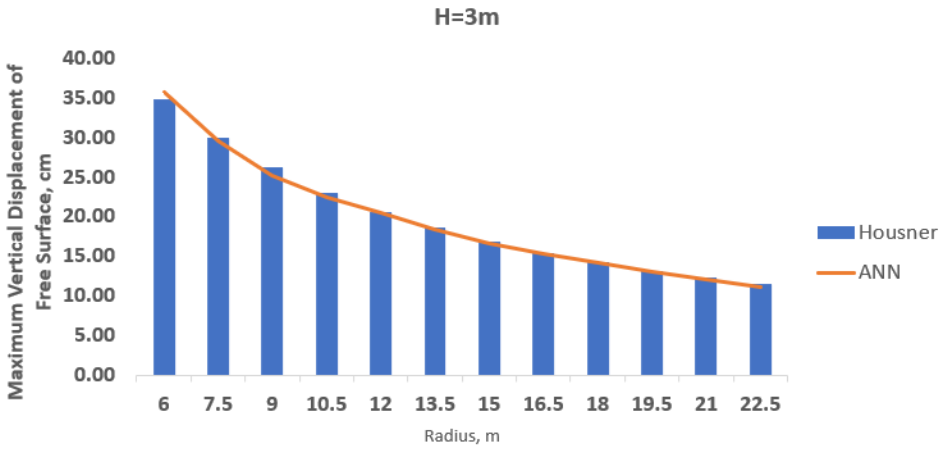


Figure 4.28 Maximum Vertical Displacement of Free Surface for the constant height of three meters and variable radius

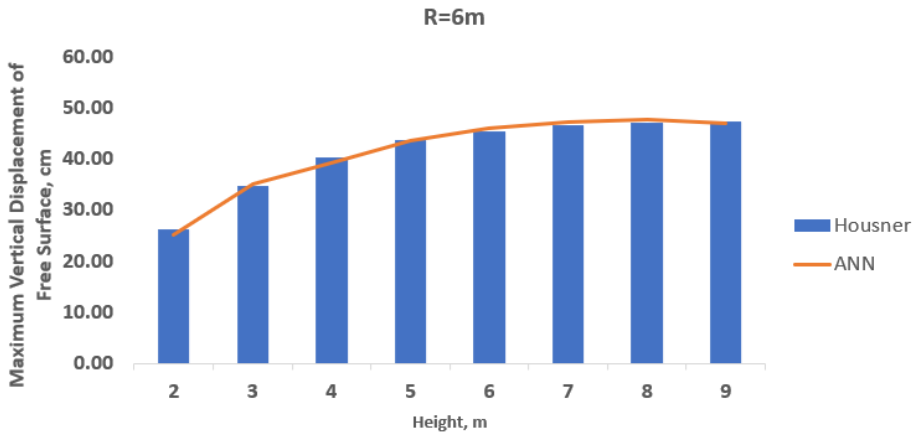


Figure 4.29 Maximum Vertical Displacement of Free Surface for the constant radius of six meters and variable height

4.5.8 Comparison of the Maximum Convective Pressure due to sloshing mode

The Peak Convective Pressure due to sloshing mode resulting from the Housner approximation method as well as the artificial neural network method were compared graphically.

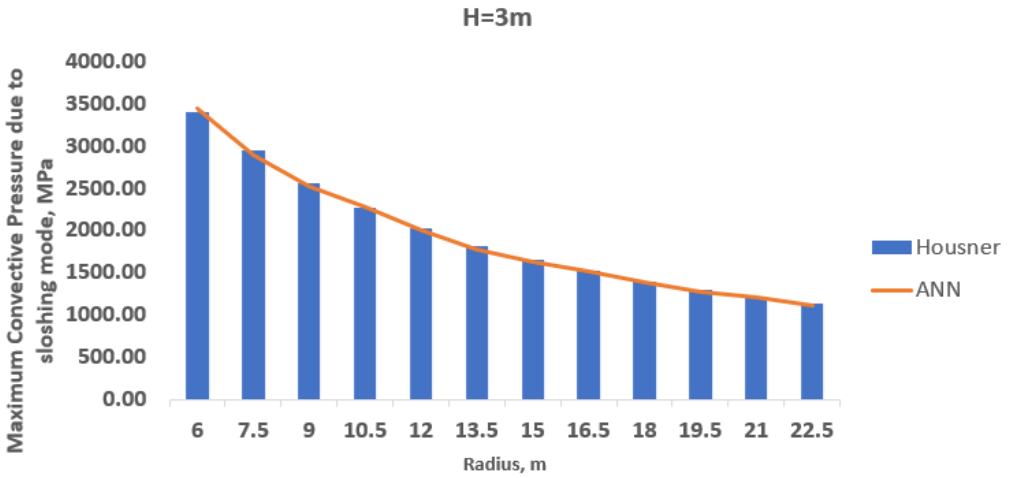


Figure 4.30 Maximum Convective Pressure due to sloshing mode for the constant height of three meters and variable radius

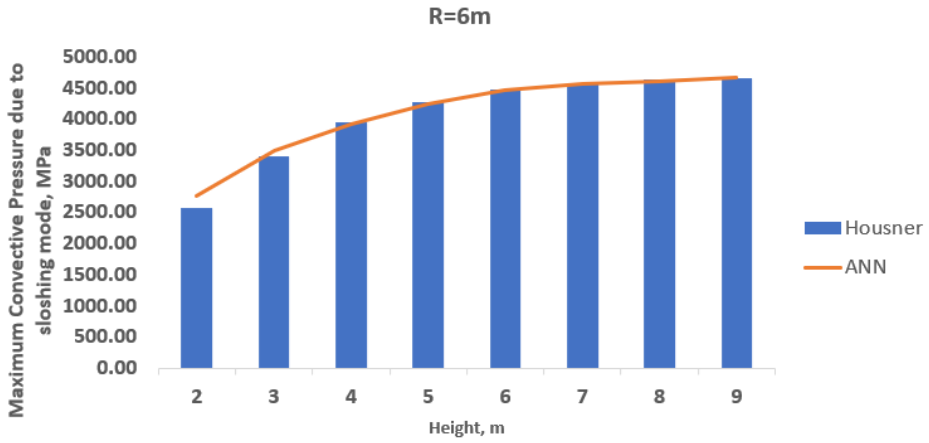


Figure 4.31 Maximum Convective Pressure due to sloshing mode for the constant radius of six meters and variable height

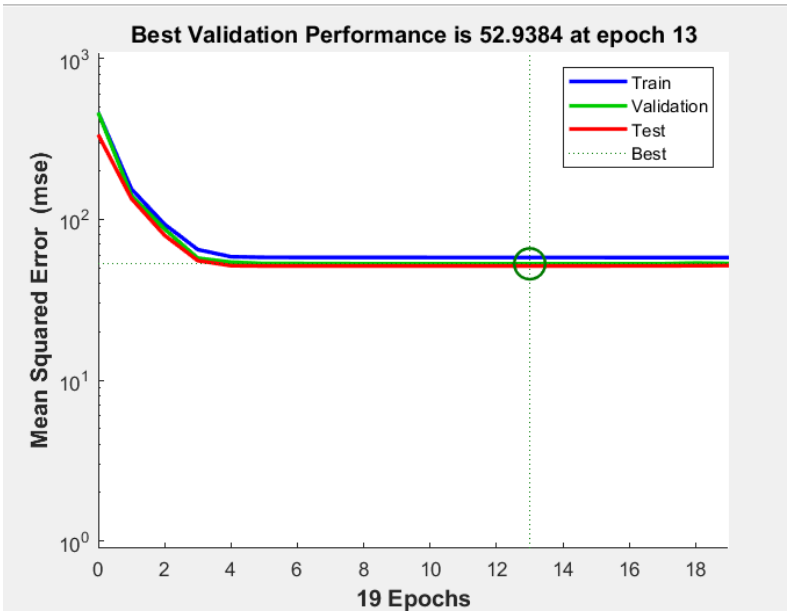


Figure 4.32 Neural Network Training Performance

4.6 Summary of the Analysis Results

Results show that the accuracy and precision of the neural network's response prediction capacity are affected by the variation in the number of hidden layers deployed, the ratio of training, validation & test samples, and the number of times the neural network is retrained.

The result of the analysis shows that keeping the height of the cylindrical tank and increasing the radius of the tank results in lower sloshing frequency where whereas increasing the height of the tank by keeping the radius constant results in higher sloshing frequency.

The impulsive force, convective force, base shear, bending moment, and overturning moment of the cylindrical tank are directly proportional to the radius of the tank, height of the tank, and density of the fluid contained in the tank.

The minimum freeboard requirement of the cylindrical tank decreases as the radius of the cylindrical tank increases with a constant height. However, the minimum freeboard requirement increases as the height of the cylindrical tank increases while the radius of the cylindrical tank is constant.

The maximum convective pressure due to sloshing decreases with increasing radius for a constant height of the cylindrical tank. In the reverse, the maximum convective pressure increases with increasing height while the radius of the cylindrical tank is kept constant.

The precision and accuracy of the trained artificial neural network model can be demonstrated using the tables provided below.

Table 4.2 Comparison between Housner's Approximation result and ANN result of Sloshing Frequency

H	R	ρ	A_o	η	S	Using Housner Approximation	Using ANN	Efficiency
5.5	12.5	920	2.4525	0.8165	1	0.9832	0.99	99.31%
6.3	13	1000	2.4525	1	1.2	0.9944	1.00	99.44%
7.2	17.3	920	2.4525	1	1	0.8200	0.83	98.78%

Table 4.3 Comparison between Housner's Approximation result and ANN result of Impulsive Force imposed on the cylindrical tank

H	R	ρ	A_o	η	S	Using Housner Approximation	Using ANN	Efficiency
5.8	13.4	1,000	4.905	1	1.2	4,007.32	4,004	99.91%
6.8	14	920	0.981	0.8165	1	1,058.03	1,063	99.53%
7.8	18.5	1,000	0.981	0.8165	1.2	2,001.63	1,997	99.77%

Table 4.4 Comparison between Housner's Approximation result and ANN result of Convective Force imposed on the cylindrical tank

H	R	ρ	A_o	η	S	Using Housner Approximation	Using ANN	Efficiency
5.5	12.5	920	2.4525	0.8165	1	83.37	83	99.56%
6.8	14	920	0.981	0.8165	1	47.49	48	98.93%
8.2	10.3	920	2.4525	0.8165	1.2	122.41	123	99.52%

Table 4.5 Comparison between Housner's Approximation result and ANN result of Base Shear imposed on the cylindrical tank

H	R	ρ	A_o	η	S	Using Housner Approximation	Using ANN	Efficiency
6.3	13	1,000	2.4525	1	1.2	2,454.69	2,439	99.36%
5.8	13.4	1,000	4.905	1	1.2	4,307.10	4,295	99.72%
7.8	18.5	1,000	0.981	0.8165	1.2	2,091.50	2,089	99.88%

Table 4.6 Comparison between Housner's Approximation result and ANN result of Bending Moment imposed on the cylindrical tank

H	R	ρ	A_o	η	S	Using Housner Approximation	Using ANN	Efficiency
6.8	14	920	0.981	0.8165	1	2,869.40	2,869	99.99%
8.6	14	920	4.905	1	1.2	29,262.37	29,229	99.89%
9	13.4	1,000	4.905	1	1.2	34,650.78	34,650	99.99%

Table 4.7 Comparison between Housner's Approximation result and ANN result of Overturning Moment imposed on the cylindrical tank

H	R	ρ	A_o	η	S	Using Housner Approximation	Using ANN	Efficiency
6.3	13	1,000	2.4525	1	1.2	25,727.78	25,700	99.89%
7.8	18.5	1,000	0.981	0.8165	1.2	31,568.84	31,584	99.95%
9	12.5	1,000	0.981	1	1	18,179.34	18,110	99.62%

CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The finite element method requires more computational effort and processing time than neural network methods, which enable quick mapping between input data and desired output data. The capability of neural networks lies in the hidden layers. These layers are more commonly referred to as internal representations since they map between the input and output patterns. A neural network is a potentially powerful tool for encoding and manipulating complex mathematical equations that are not explicitly defined as compared to other computational methods.

Dynamically loaded, ground supported cylindrical tanks respond vigorously to seismic actions due to the convective pressure caused by the liquid sloshing. ANN models can give a precise and more accurate prediction of the dynamic response of cylindrical tanks for the reason that ANNs are capable of using training data & algorithms to mimic the way humans learn, step by step improving its accuracy. To improve the accuracy and efficiency of the neural network one can always retrain the program further till the desired accuracy and precision are achieved. Adjustment of sample ratios among training, validation, and test samples can also be done to improve accuracy of results.

Aspect ratio and height of cylindrical tank are among the major variables that influence the dynamic response of fluid containing cylindrical tanks. As the heights and aspect ratios of the cylindrical tanks increase the base shear and overturning moment of the system also increases. Seismic loading can have an impact on the normal operation of tank-liquid systems, jeopardizing their safety and stability. During a seismic event, the liquid on the tank walls and bottom induces time-dependent hydrodynamic forces, pressures, and stresses. Knowledge of hydrodynamic effects is crucial since they influence tank responses and must be considered in the design.

Earthquake damage to steel storage tanks can be revealed in a variety of ways. A sloshing effect of the upper portion of the liquid is one of the phenomena observed at the free liquid surface. The phenomenon of oscillation of the convective liquid in the storage tanks can cause deformations and ruptures of the tank walls due to impact. Sloshing may

also result in detrimental circumstances such as damage to the top cover and walls of closed tanks or can result in liquid spilling in open tanks. In seismically vulnerable areas, it is necessary to allocate a minimum of 17% of the fluid's total height as freeboard to prevent such destructions.

The intent of this paper was to conduct a seismic analysis of the liquid storage tank with an emphasis on the sloshing response to a particular earthquake loading using artificial neural networks. Despite the constraints imposed during implementation on the training and testing sets, the findings unequivocally demonstrate that the application of neural networks represents a notable advancement in remeshing structural elements. The developed artificial neural network models exhibit an average efficiency of 99.61% in forecasting the dynamic response of cylindrical tanks containing fluid.

5.2 Recommendations

Below are the proposed recommendations for further utilization of the study:

- Developing a neural network model that integrates the second-order effects of axial load while considering modal superposition.
- Developing a neural network model that incorporates different types of fluids at the same time and how this would affect the sloshing response.
- Include the influence of the prescence baffles in the dynamic response of cylindrical tanks.

REFERENCES

1. Howard I. Epstein," Seismic Design of Liquid-Storage Tanks," Journal of the Structural Division, ASCE, Vol. 102, No. ST9, September, 1976: pp. 1659-1673.

2. Medhat A. Haroun & George W.Housner," Dynamic Characteristics of Liquid Storage Tanks," Journal of Engineering Mechanics Division, ASCE, Vol. 108, No.EM5, September, 1982: pp.783-800.

3. Konstantin Meslouris, Structural Dynamics; Models, Method and Experiment, Erst & Sohn, 2000.

4. Edward L. Wilson, Three-Dimensional Static And Dynamic Analysis of Structures, 3rd edition, Computers and Structure Inc., Berekley,2002.

5. Anil K. Chopra, Dynamics of Structure,5th Edition, Pearson, New Jersey, 2017.

6. Jean Donea and Antonio Huerta, Finite Element Methods for Flow Problems, John Wiley & Sons Ltd, Chichester, 2003.

7. Medhat A.Haroun and Magdy A.Tayel, "Axisymmetrical Vibrations of Tanks-Numerical," Journal of Engineering Mechanics, ASCE, Vol. 111, No. 3, March, 1985: pp.329-345.

8. Konstantin Meslouris, Structural Dynamics; Models, Method and Experiment, Erst & Sohn, 2000.

9. Medhat A. Haroun, Engineering Mechanics Division Specialty Conference (Liquid Sloshing in Cylindrical Tanks), University of Wyoming, Laramie,1984.

10. Muller, B. J., and Reinhardt, J., Neural Networks: An Introduction, Springer-Verlag, Berlin, 1990.

11. D. A. Berrais, Artificial Neural Networks in Structural Engineering, Abha, Saudi Arabia: Abha College of Technology, 1999.

12. M.A. Goudarzi and S.R. Sabbagh-Yazdi, Evaluating 3D Earthquake Effects on Sloshing Wave Height of Liquid Storage Tanks Using Finite Element Method, Tehran, Iran: K.N. Toosi University of Technology, 2008.

13. M. Cheng and N. Popplewell, Neural Network for Earthquake Selection in Structural Time History Analysis, Manitoba, Canada: John Wiley & Sons, Ltd., 1994.

14. ANSYS Release 14.0, ANSYS, Inc.,2011.

15. Khalifa S. Al-Jabri, and Saleh M. Al-Alawi, An Advanced ANN Model for Predicting the Rotational Behaviour of Semi-rigid Composite Joints in Fire Using the Back-Propagation Paradigm, Muscat, Oman: Sultan Qaboos University, 2010.

16. S. Seyoum, Dynamic Analysis of Liquid Containing Cylindrical Tanks, Addis Ababa, Ethiopia: Addis Ababa University, 2005.

17. Mahesh P. Nagarkar, M. A. El Gohary, and Yogesh J. Bhalerao, Artificial Neural Network Prediction and Validation of Optimum Suspension Parameters of A Passive Suspension System, Ahmednagar, India: MIT Academy of Engineering, 2019.

18. Qiao Jin, Xin Li, Ning Sun, Jing Zhou, and Jiong Guan, Experimental and Numerical Study on Tuned Liquid Dampers for Controlling Earthquake Response of Jacket Offshore Platform, Dalian, China: Dalian University of Technology, 2007.

19. R. A. Ibrahim, Liquid Sloshing Dynamics Theory and Applications, New York: Cambridge University Press, 2005.

20. Design of Structures for Earthquake Resistance - Part 1: General rules - seismic actions and rules for buildings, Ethiopian Standards Agency, 2015.

21. J. Ali, Neural Networks: A New Tool for the Petroleum Industry?, Aberdeen, United Kingdom: Society of Petroleum Engineers, 1994.

22. Konstantin Meskouris, Christoph Butenweg, Klaus-G. Hinzen, Rüdiger Höffer, Structural Dynamics with Applications in Earthquake and Wind Engineering, Berlin, Germany: Springer-Verlag GmbH, 2019.

APPENDIX: Program Codes

Matlab code with matrix and cell array argument support after training of neural networks

```
function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = Qx6 matrix, input #1
% and returns:
%   y = Qx6 matrix, output #1
% Q represents the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [1.5;6;920;2.4525;0.8165;1];
x1_step1.gain                                                    =
[0.307692307692308;0.133333333333333;0.025;0.815494393476045;10.8991825613079;10];
x1_step1.ymin = -1;

% Layer 1
b1=[-1.7896657399207576145;-
1.8023206102457316469;0.41868670649728967348;-
0.68565678192252710321;1.5751919166877987877;-
1.5846463723408688029];
IW1_1=[0.92255379556063221713         0.30724256602768901248
0.032446637490220912869               0.24872986945536620929
0.0028447002770737093817
0.00097482912516777390534;0.69775774530227574477
0.56863881125232451641                0.032883571510540700322
0.26877252777144861051       0.0012488244932904615878       -
0.00059111574324462829 93;-0.80162388417716290423            -
0.78567081870988064729       -0.022200764725394762922       -
0.13722499469353741097       -0.013339804109446370614       -
0.012334276163766266093;0.99518021700324865719
1.0846350844535113112        -0.061150072970339017231       -
1.9864690162124800654        0.014837024069178651395        -
1.1721460923552573519;1.4549368625426131185                  -
0.58664115223845691993                -0.23008008867641749351
2.5348627308370845768        0.067672961716805896115        -
0.074539733601078947411;0.22796569862515578264
0.86240480471372049642                0.032853390667752217 78
```

```
0.21918990270551214072      1.8769097748449711273e-05      -
0.0035783847029147073347];

% Layer 2
b2=[0.069455650772791879821;1.2398730967274502746;-
0.0049606332154047277916;1.1915602075329474463;1.3141281903
93294162;1.1097803537301476062];
LW2_1=[0.15044824801120243452    0.36359978893009509138    -
0.32005375935188273839               -0.15083886958639075582
0.15535930605409423166                                      -
0.35887813570186111178;0.097435300811945194099
2.8352099774098560303        -0.27052245867839153082        -
0.010730732127067571513       0.016663349494532395112       -
0.98057732747596715317;0.42299724240389907637
0.15308911732368102321               -0.24788157110112560289
0.038105270298213381186               0.13862333154453168604
0.077567141245433623986;0.038260224729991035297
2.866230414473942556        -0.28000439471754878795        -
0.01407558350986543895        0.016646614081402878044       -
1.0109015614688063867;1.0400229494828772125
2.0386378177091417641                -0.17108242366157994097
0.007325201418671789056      0.001343655928070877549 8      -
0.92952301762456657208;-1.8919698102008868368
4.6751661930206305229                -0.20953061480830806329
0.001301245882323767352 8     -0.004977339942748157213      -
0.87307533833790917921];

% Output 1
y1_step1.ymin = -1;
y1_step1.gain                                               =
[1.44243372022346;0.000168078394213618;0.00329955861211368;
0.000159931536122681;5.21126533369748e-05;9.2150754384918e-
06];
y1_step1.xoffset                                            =
[0.335145491149714;55.2483131923885;7.92961338157494;63.177
9265739634;37.1270828294583;337.590110283852];

% ===== SIMULATION ========

% Dimensions
Q = size(x1,1); % samples

% Input 1
x1 = x1';
xp1 = mapminmax_apply(x1,x1_step1);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;
```

```
% Output 1
y1 = mapminmax_reverse(a2,y1_step1);
y1 = y1';
end

% ===== MODULE FUNCTIONS ========

% Set the lowest and the highest value input processing
functions.
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Set minimal and maximal Output Reverse-Processing
Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end
```

Matlab code with matrix-only arguments(no cell array support) after neural network fitting used for predicting output layer elements using:- A = net(op(:,R))

Where A is the predicted value of the output layer

net is the neural network model

op is the file name of the imported numerical matrix(excel) file of input values

R is the number of row of the output element to be predicted

```
% Implement a neural network to deal with an input-output
fitting problem
%
% This code assumes these variables are defined:
%
%   data_1 - input data.
%   data_2 - target data.
```

```
x = data_1';
t = data_2';

% Choose a Training Function
% To view a list of all training functions type: help
nntrain
% 'trainlm' is usually fastest.
% trainbr' is more time-consuming but may be better for
complex tasks.
% 'trainscg' uses less memory. Suitable in low memory
situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt
backpropagation.

% Create a Fitting Network
hiddenLayerSize = 6;
net = fitnet(hidden layer size,trainFcn);

% Set up data divisions for training, validation, and
testing.
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment the following lines in order to enable numerous
plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```

Matlab code with matrix and cell array argument support after training of neural networks

```
function [y1] = myNeuralNetworkFunction(x1)
```

```
%MYNEURALNETWORKFUNCTION    neural    network    simulation
function.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = Qx6 matrix, input #1
% and returns:
%   y = Qx3 matrix, output #1
% Q represents the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [1.5;6;920;2.4525;0.8165;1];
x1_step1.gain                                                  =
[0.307692307692308;0.133333333333333;0.025;0.81549439347604
5;10.8991825613079;10];
x1_step1.ymin = -1;

% Layer 1
b1=[-
0.33698584614901783141;0.55980541073973477673;0.24711065596
46509826;0.39465333094823873417;-0.69087014233844779982;-
0.42052810487079117507];
IW1_1=[-1.0855210564067410139        -2.5630535557975839822
0.74963241669824309981               0.14337002323125297432
0.43107388319118022935              1.02308715470797251260;-
0.76894293169267113885       -0.87341070683078858572       -
0.06197407677953681304        0.48696617687277155007       -
0.30051001827057288729          -0.00090409550050873921379;-
0.83210741570889234020               0.44245890288035355509
0.09361481454544917646        1.2335652472411151592        -
1.1154270966498920359                                        -
0.46887562963238038627;0.78329587118769450349
0.56850702880087755631              -0.60293874402032265891
0.41499739716684613811              -0.15359708264803351163
0.76075002253787982198;-3.2843621504561890667                -
3.455860725317983384         0.93612099672302051534         -
0.91575870238418977110               1.17656095717678632620
1.37275843287152299330;-1.04643145838569706640               -
0.73502509769590140820        0.48397498495324081658         -
0.76438746155418757500               0.69741453963112409742
0.07539563612983651852];

% Layer 2
b2=[-0.46466766756278055084;-0.39372373644421176930;-
0.36613523654682822439];
LW2_1=[-0.04263875854747919690    0.14372655612555937770    -
0.01226181472379371528              0.09814773492455408643
0.23711628026559869453             -0.11009750462055946374;-
0.13521660703814192273              -0.38556391635421805031
```

```
0.18506174869926539372                0.18967388871975371378
0.048771684746402393928
0.15491294315438244866;0.050888744324826207244
0.23756404668880648501               -0.22415191660966632403
0.009030943281199688144 1      0.10487868682820267185      -
0.30148624671478718717];

% Output 1
y1_step1.ymin = -1;
y1_step1.gain    =    [0.0370307543118089;2.53943482844653e-
07;0.000376058095666393];
y1_step1.xoffset                                              =
[2.55204298723708;132790.962778586;230.326983684121];

% ===== SIMULATION ========

% Dimensions
Q = size(x1,1); % samples

% Input 1
x1 = x1';
xp1 = mapminmax_apply(x1,x1_step1);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
y1 = mapminmax_reverse(a2,y1_step1);
y1 = y1';
end

% ===== MODULE FUNCTIONS ========

% Set the lowest and the highest value input processing
functions.
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Set minimal and maximal Output Reverse-Processing
Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
```

```
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end
```

Matlab code with matrix-only arguments(no cell array support) after neural network fitting used for predicting output layer elements using:- A = net(op(:,R))

Where A is the predicted value of the output layer

net is the neural network model

op is the file name of the imported numerical matrix(excel) file of input values

R is the number of row of the output element to be predicted

```
% Implement a neural network to deal with an input-output
fitting problem
% This code assumes these variables are defined:
%
%   data - input data.
%   data_1 - target data.

x = data';
t = data_1';

% Choose a Training Function
% To view a list of all training functions type: help
nntrain
% 'trainlm' is usually fastest.
% trainbr' is more time-consuming but may be better for
complex tasks.
% 'trainscg' uses less memory. Suitable in low memory
situations.
trainFcn = 'trainlm'; %Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 6;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% To view a list of all processing functions type: help
nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns                                    =
{'removeconstantrows','mapminmax'};

% Set up data divisions for training, validation, and
testing.
```

```
% To view a list of all data division functions type: help
nndivision
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up each sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% To view a list of all performance functions type: help
nnperformance
net.performFcn = 'mse';  % Mean Squared Error

% Choose Plot Functions
% To view a list of all plot functions type: help nnplot
net.plotFcns                                                =
{'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)
%load outsample data
op = predictioninput1';
%to predict the outcome based on obseravtion
%with new data file name
a=net(op(:,1))

% Plots
% Uncomment the following lines in order to enable numerous
plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```

```
% Deployment
% Change the (false) values to (true) in order to enable
the following code blocks.
% See the help for each generation function for more
information.
if (false)
    % Develop a MATLAB function for neural networks for use
in an application
    % deployment in MATLAB scripts or with MATLAB Compiler
and Builder
    % tools, or simply to examine the calculations your
trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural
network code
    % generation with MATLAB Coder tools.

genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes
');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or
deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```