Fall 1-1-2020

Development of Artificial Neural Network Software And Models For Engineering Materials

Abdallah F. Bseiso
Cleveland State University

DEVELOPMENT OF ARTIFICIAL NEURAL NETWORK SOFTWARE AND MODELS FOR ENGINEERING MATERIALS

ABDALLAH F. BSEISO

Bachelor of Civil Engineering

Palestine Polytechnic University

June 2017

submitted in partial fulfillment of requirements for the degree

MASTER OF SCIENCE IN CIVIL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

December 2020

We hereby approve this thesis

For

ABDALLAH F. BSEISO

Candidate for the Master of Civil Engineering degree

For the Department of

Civil and Environmental Engineering

And

CLEVELAND STATE UNIVERSITY'S

College of Graduate Studies by

_____

Dr. Josiah Owusu-Danquah
Advisor, Chair

_____

Department & Date

_____

Dr. Srinivas Allena

_____

Department & Date

_____

Dr. Stephen F Duffy

_____

Department & Date

Student's Date of Defense: 11/24/2020

DEDICATION

To my family, who have given all the support through these years, my professors, and friends.

DEVELOPMENT OF ARTIFICIAL NEURAL NETWORK SOFTWARE AND MODELS FOR ENGINEERING MATERIALS.

ABDALLAH F. BSEISO

ABSTRACT

Artificial Neural Network (ANN), which is inspired by biological neural networks in the human brain, is one important tool of machine learning that creates artificial intelligence through computational systems. The creation of this intelligence is contingent on learning from available data regarding a specific subject. Although machine learning, in general, has profuse applications in most scientific disciplines, yet few have been developed in civil engineering due to the required time consuming and demanding programming. In order to minimize this, intelligible ANN software has been developed in this research capable of training networks with any number of hidden layers and nodes for each layer. Furthermore, two models have been created to demonstrate the robust applications of ANN. The first application involves a simulation of the strain-temperature behavior of a shape memory alloy (SMA) under thermal cycling. In the second case, the bond strength between the concrete and the steel-reinforced bars is predicted considering the effects of steel corrosion level, concrete compressive strength, and concrete cover. Java programming language was used in developing the ANN software and a simple graphical user interface (GUI) has been designed, allowing the user to control the inputs and the training progress, make predictions and save the outputs. In this study, the ANN models were developed with different structures and activation functions to prove the ANN eminent idiosyncrasy of modeling data from different fields. Comparison is made between these models as well as models created by statistical

regression and other models available in the literature. The developed software can efficiently train ANNs with any structure, as less time is needed to develop one ANN using the software than using programming methods. Moreover, the user will have the option to save the weights and the biases at any iteration and predict responses for the currently trained or previously trained ANN. The model predicted results can be saved or exported as an excel file. In terms of the created models, ANN can capture highly complicated relationships accurately and effectively compared to traditional modeling methods. Based on that, more accurate predictions are expected using ANN.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

ANN is a computational algorithm that consists of a number of neurons communicating with each other based on a predetermined system. This algorithm is capable of learning from available data regarding a specific subject, thus, forming an artificial intelligence which can make decisions and predictions, not just following a set of instructions made by the programmer. The power of ANN lies in accurately simulating the convoluted relationships and patterns in an enormous amount of training data. Therefore, a wide range of ANN applications has been developed in the last decade in various fields including medicine, economics, management, and engineering (Wu et al. 2018).

The structure of the ANN consists of parallel layers, one input layer, one output layer, and at least one hidden layer. Each layer has a specific number of neurons activated with a certain function (Wu et al. 2018). The performance, accuracy and required training time of the ANN are affected by the design of its structure, i.e., the number of hidden layers, the neurons in each layer, and the activation function. Whiles increasing the number of hidden layers and neurons generally increases the ANN's ability to capture the patterns and relationships between the different parameters in the training data, it

increases the complexity of the ANN structure (Kavzoglu 1999). A more complex structure goes hand in hand with more training time, which has more significant effects on training massive data. In addition, ANN with complex structure could cause overfitting in which the ANN gives accurate predictions only for the training data but contrary for any other datapoint outside the training data. Moreover, the activation function of the hidden layers also has huge effects on the performance of the ANN (in terms of the training process and the output). Thus, the ANN structure should be chosen judiciously to maintain the required accuracy of the ANN while having the least training time possible. It is worth knowing that after a certain optimized point, increasing the structural complexity may cause a negative or no effect on the ANN accuracy.

In order to demonstrate the wide range of applications of ANN in different fields, two ANN models have been developed in this study in different applications. The first ANN model developed in this study is the strain model for the cyclic behavior of a 55NiTi shape memory alloy material subjected to isobaric thermal cycles. The strain magnitude and the actuation character of the material are affected by the applied stress, cycle number, and temperature. The purpose of this model is to accurately predict the strain behavior of SMA under thermal cycling instead of the several micromechanical and phenomenological constitutive models that have been formulated previously in literature (Cisse et al. 2016; Khandelwal et al. 2009; Lagoudas et al. 2006; Saleeb et al. 2011; Owusu-Danquah et al. 2017). In addition, the ANN model aims to simulate the strain behavior with a small number of equations despite the convoluted patterns in the data while other models from literature consist of a significant and unwieldy number of equations. Furthermore, the ANN model has another advantage of predicting the strain

behavior of upper thermal cycles, which is very important as SMAs under real-life applications are subjected to heating and cooling for an enormous number of cycles. Finally, the ANN model aims to predict the SMA strain behavior under new stresses. The second model simulates the bond strength between steel rebars and concrete. Since conducting the pullout test under every possible condition is not feasible, this model aims to predict the bond strength at any values of corrosion level, concrete cover, and concrete compressive strength. Moreover, this ANN bond strength model is compared with nonlinear regression and previously developed models in literature to attest to the ANN superiority over other conventional modeling methods.

This study, which explores two different applications of ANN in engineering materials, has some limitations as a consequence of the available experimental data. Starting with the bond strength model, it only accounts for the effects of 3 factors (corrosion level, concrete cover, and concrete compressive strength), while other factors, such as the embedment length and the bar diameter, also affects the value of the bond strength. Moreover, the available experimental data is limited to 90 specimens which could affect the accuracy of the model as increasing the size of the training data set increases the accuracy of the ANN unless the extra data is doesn't have any significant effect (as might be experimentally observed). Regarding the SMA model, separate ANN is trained for each stress of the 4 stresses, that are available in the experimental data, to generate accurate models. Furthermore, the ANN is used to predict the strain behavior of upper cycles, from 101st to 200th, without consideration of fatigue failure which may occur during these cycles.

Currently, many obstacles hinder developing various ANN applications in many fields, especially civil engineering (Silva et al. 2017). To begin with, the process of designing the ANN and identifying the most adequate structure requires a huge amount of time and effort due to the lack of rules in this process and its reliance on trying different structures, then selecting the best one. For each trial, a new programming code must be built before training the ANN once more, then compare the results with other trials. Moreover, the paucity of required ANN programming skills by researchers and professionals in many disciplines impels them to use other methods. In order to overcome these challenges, ANN software has been developed in this research that can be used on any PC computer. The software consists of interactive windows allowing the user to train any ANN easily without writing any piece of code. In addition, changing the structure of the ANN can be done with few clicks which makes the process of identifying the best structure more facile.

CHAPTER II

ANN SOFTWARE

2.1 Methodology

The procedure of building any computer software starts with identifying the main purposes of that program and the tasks it is going to perform, then the most compatible environment and programming language are chosen. The ANN software designed in this research aims to build, train and predict different ANNs by ordinary computer users through simple and typical buttons, frames, windows, text fields, and files. Therefore, Windows operating system was selected to be the environment for the software as it is the most installed operating system on personal computers and the vast majority of computer users are familiar with it. The software was developed using Java programming language due to its convenience in creating a simple and flexible graphical user interface with Java Core Libraries. This language is also used by other operating systems which makes translating the software to other environments more feasible including other platforms such as android smartphones. The GUI of the software consists of three main sections which can be navigated between using the tab bar at the top. The first section is used for training new ANN while the second section is for predicting values using the

ANN under training in the first section. However, the third section is responsible for prediction in case there is already established or previously trained ANN.

The interface of the training section consists of 8 buttons, 2 text fields, 1 spinner, 3 checkboxes, 1 drop list, and 5 text labels. The spinner is responsible for identifying the number of hidden layers while the number of nodes in each layer is identified by clicking on "Enter number of nodes" button which opens a small dialog as many times as the total number of layers allowing the user to enter the number of nodes in a text field. The training data is divided into two groups, the input data, and the target data, each group is imported from a separate excel file using different buttons, the input data is stored in a two-dimensional array in the memory called "x" while the output data is stored in a one-dimensional array called "y". The 2 checkboxes and the 2 text fields determine when the training process will stop. The user can choose to end the training at one of the following 4 cases; (i) at a certain number of iterations, (ii) if the mean square error is less than a certain value, (iii)whatever comes first of the previous two conditions or (iv) it never stops automatically but rather decided by the user. The training process begins by clicking on the "Start" button constructing a new object of the Training class and passing the stored variables of the ANN structure and training data into this object. The training object creates a new thread which handles the training process in the background allowing the interface of the software to stay functional. Initially, the software generates random values between 0.005 and 0.035 for the weights and the biases storing them in a three- dimensional array for the weights and a two-dimensional array for the biases. The first index of these arrays indicates the node number while the second index indicates the layer number. The third index in the weight array indicates the node number in the

previous layer that the weight is multiplied with. The training process aims to minimize the square error function, which is the average square difference between the predicted and the target outputs of the training data, by calculating the derivative of this function in respect to each weight or bias and then update this weight or bias by subtracting the derivative value from it. Before starting this process, since calculating each one of these derivatives depends primarily on the ANN structure, a five-dimensional array is formed which indicates the number of terms in each derivative and which weights or nodes are multiplied in each term. The first three indexes in this array are used to identify which weight the derivative is for. The fourth index is for the term number in the derivative while the fifth index indicates the weights or nodes that are multiplied in each term to calculate the derivative. At the beginning of each iteration, all the derivatives are calculated based on the updated weights and biases from the previous iteration and then the weights and the biases are updated once again. At the end of each iteration, the mean square error is calculated and displayed with the iteration number on the interface using 2 text labels during the training process.

The second section of the software aims to make predictions using the ANN under training in the first section. For instance, the user can pause the training at any time and make predictions to evaluate the training progress. The ANN parameters, such as number of layers and last updated weights and biases, are obtained from the training section. The inputs, which the user wants to predict for, can either be one point or multiple points. If the first option was chosen, a small dialog will pop up asking the user to enter the inputs which the point consists of. The output, in this case, is just one value displayed by the text label on the GUI. However, predicting for multiple points requires

the user to store the inputs in an excel file then import it to the software. When the "Predict" button is clicked, the outputs for all the points are stored in a one-dimensional array inside the computer memory and can be exported as an excel file. The third section makes predictions the same way as the second section except the ANN parameters are defined by the user manually instead of obtaining them from the training section.

2.2 User Guide

The user guide explains how to properly use the software and achieve your goals of training new ANNs or making a prediction using trained ANNs. Therefore, this guide clarifies the function of every single component on the GUI, some of which are buttons, text fields, checkboxes, and labels, how they are properly used, in what order they should be used and what the properties are for the imported or exported files. To begin with, Figure 1 shows the tab bar at the top which allows the user to navigate between the software's three main sections. The first section is for training new ANN while the second predicts for the ANN under training in the first section. The third section makes predictions the same way as the second but for previously trained ANN. The rest of this chapter presents the guidelines for using each one of these sections.

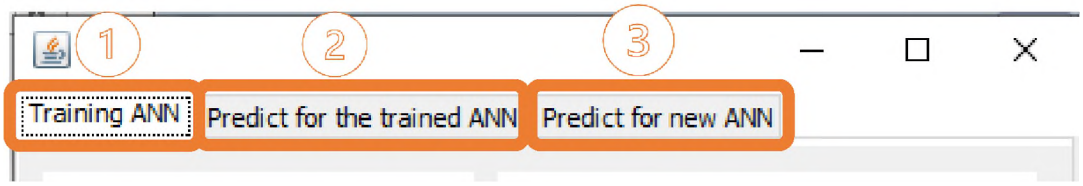


Fig. 1: Tab bar to navigate between the software's three main sections.

2.2.1 Training New ANN

Figure 2 displays the components of the training section and assigned numbers based on the order they should be used in. The ANN structure and parameters are defined

through the first four components. The number of hidden layers is defined using component (1), which is a spinner, and its value must be no less than 1. After clicking on component (2), a small dialog will pop up (see Figure 3) multiple times allowing the user to enter the number of nodes at each layer in this order, input layer, hidden layers, and lastly the output layer. In the current version, the software supports only one node for the output layer. Component (3) opens a new file explorer window (see Figure 4) allowing the user to choose the excel file for the input data. Component (4) serves the same function as component (3) but for the target data. Excel files should only contain numerical data with no headings or texts. Each row in the input file presents a separate point (sample) and correlates with the same row in the target file. Therefore, the number of rows in the input and target excel files must be the same. On the other hand, each column in the input file correlates with one input variable, so the number of columns must be the same as the number of nodes in the input layer.

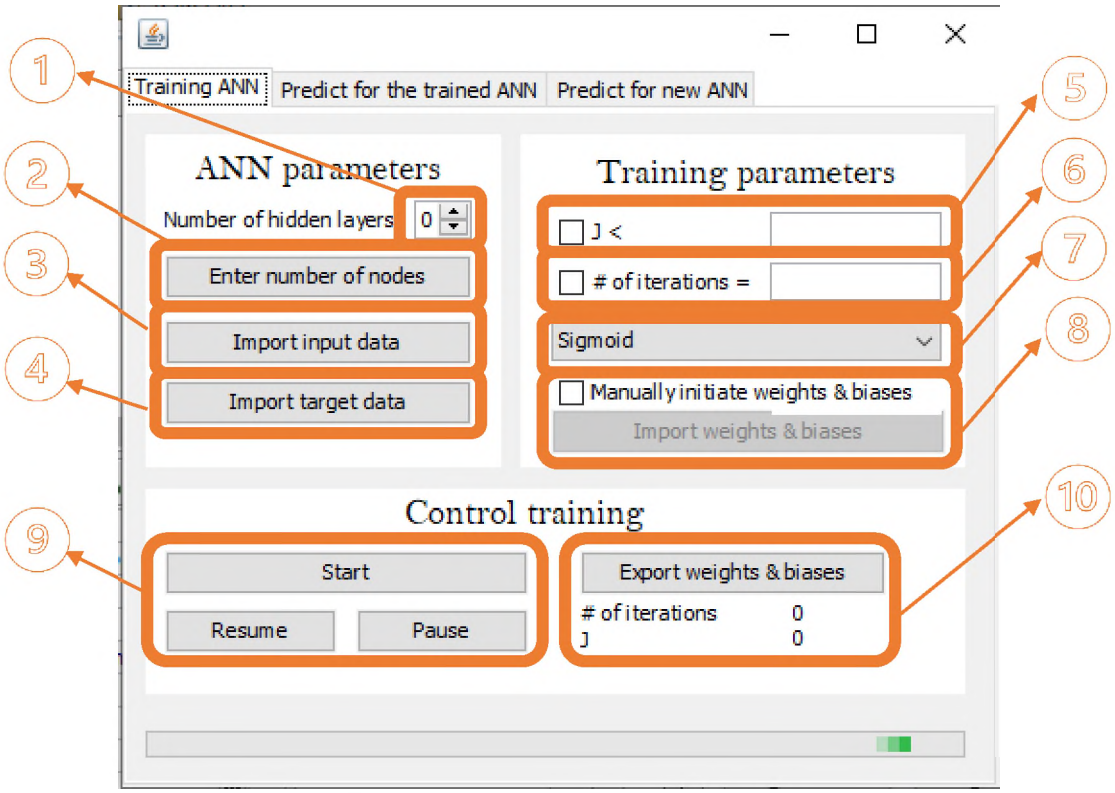


Fig. 2: The interface of the first section of the software for training new ANN.

Before the start of training the ANN, the user has the option to specify iteration conditions, and if any of these conditions are satisfied, the training process stops automatically. When component (5) is selected, the training process stops when J is less than the value in the corresponding text field. On the other hand, selecting component (6) stops the training process when the iteration number equals the value in the corresponding text field, which should be only an integer. If both components are selected, the training process stops whenever one of the two conditions is realized. The activation function for the hidden neurons can be chosen between Sigmoid, Relu, and Softplus through component (7) which is a drop list that contains these 3 functions. By default, the weights and biases are generated automatically and randomly between 0.005 and 0.035. However, the user can define the values of the weights and biases manually by

checking the box in component (8), enabling the button in the same component which opens a file explorer window to choose the excel files storing the values of the weights and biases. Each file contains the weights and biases of a certain layer while each row in this file contains the weights and biases of the corresponding node. All the values in the file are weights except the values in the last column, which are biases. The column number for the weights represents the node number in the previous layer which the weight is multiplied with. For example, the weight that exists in the file named layer 4, row 2 and column 5 is the weight for layer number 4, which is the third hidden layer, node number 2 and multiplied with the fifth node in layer number 3. However, the biases can be defined only using the layer number and the node number, which is the same as the row number, since they are not multiplied with any node.

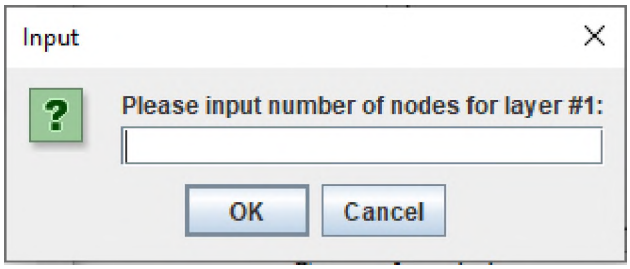


Fig. 3: Input dialog.

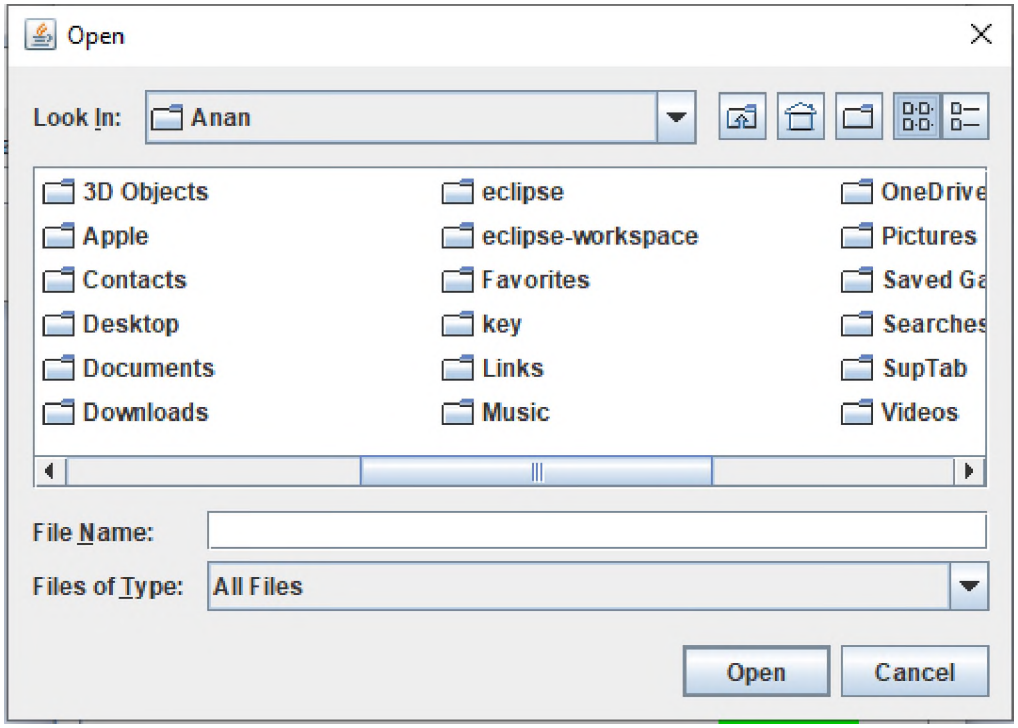


Fig. 4: File explorer window.

By clicking on the "Start" button, the software starts training the ANN and updates the iteration number and J on the text labels in component (10) informing the user of the progress of the process. The user will not be allowed to make modifications on the ANN parameters unless the training has been stopped or paused, if modifications are made, the "Start" button must be clicked once more to restart the training process with the new parameters. Whenever the training process is stopped or paused, clicking on the button in component (10) exports the weights and biases of the ANN into multiple excel files in the same structure that was discussed before for importing the weights and the biases.

2.2.2 Making Predictions

Predictions for the ANN under training are made using the second section of the software which is shown in Figure 5. The user can and must check only one box in both

components (1) and (2), either predicting for one point or multiple points. If the earlier option (1) is chosen, the inputs (or variables) for the point must be entered one at a time in a small dialog which will open as many times as the number of nodes in the input layer after clicking on "Enter point" button. However, predicting for multiple points requires importing the inputs as an excel file in which each row represents one point (sample), and each column represents one input (variable) in the input layer. This excel file can be chosen using file explorer which opens after clicking on "Import points" button. After identifying the inputs using either component (1) or (2), component (3) is clicked to perform the mathematical calculations in the background and display the result in the text label in component (4). If the predictions are made for multiple points, the outputs can be stored in an excel file that consists of one column, as the output layer has only one node and as many rows as the number of points (samples) in the input file.

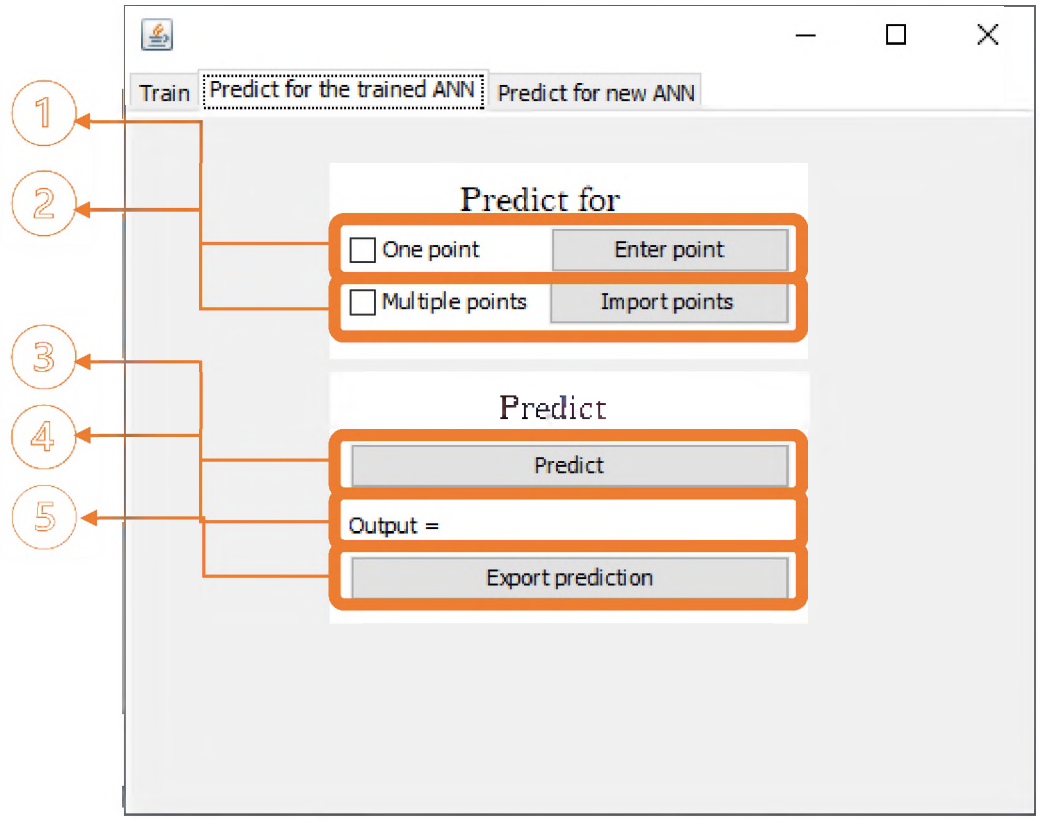


Fig. 5: The interface of the second section of the software for predicting using the ANN under training.

The third section serves the same function as the second but for new ANN (see Figure 6). Thus, the ANN parameters must be defined before making any prediction. Component (1) identifies the number of hidden layers in the ANN while component (2) allows the user to enter the number of nodes in each layer, including the input and output layer. Then, the weights and biases can be entered one by one in a small dialog box by clicking on "Enter weights and biases" button or they can be imported from excel files by clicking on "Import weights and biases" button. Component (4) is a drop list used to choose the activation function for the hidden layers of the ANN. Component (5) follows the same instructions, explained in the previous paragraphs for making predictions for ANN under training.

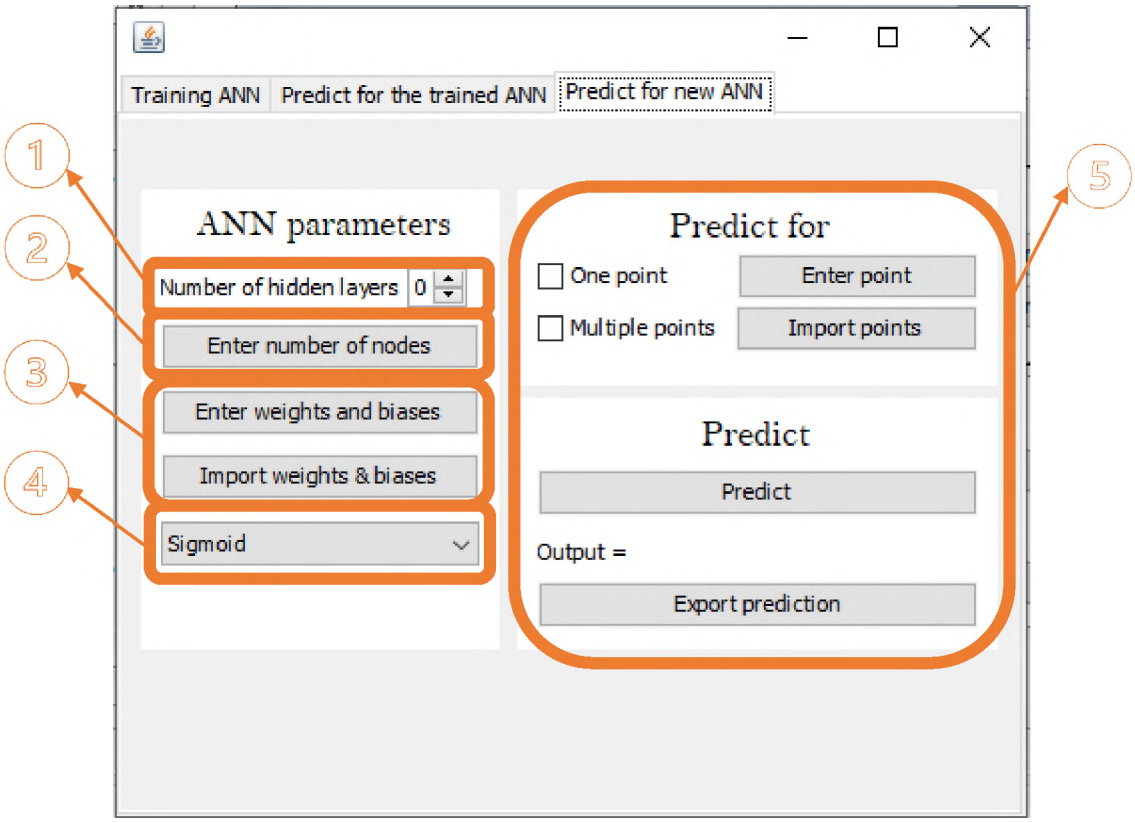


Fig. 6: The interface of the third section of the software for predicting using previously trained ANN.

CHAPTER III

SHAPE MEMORY ALLOY STRAIN MODEL

3.1 Introduction

Many studies have been conducted on 55NiTi shape memory alloy, composed of 55 wt% nickel. These studies investigate the application and control of their useful thermo-mechanical properties in the fields of energy and actuation. Shape memory effect and superelasticity are the two main distinctive characteristics which make the 55NiTi adequate to be used efficiently in various fields of engineering. NiTi, as all other shape memory alloys (SMAs), demonstrate the ability to recover huge strains when heated from the martensitic phase to the austenitic phase returning to its initial shape, this ability is called one-way shape memory effect (OWSME). Additionally, SMA can be trained to recover its martensitic shape when cooled from the austenitic phase to the martensitic phase having two-way shape memory effect (TWSME). Superelasticity is the material's ability to recover from relatively high strains spontaneously when the stress is removed isothermally. Due to these and other properties, NiTi has been used in a wide range of applications including thermal and electrical actuators, medical devices and orthopedic implants, intelligent reinforced concrete (IRC) with a self-rehabilitation ability of small cracks, heating and cooling devices, and many other applications (Tang et al. 2012;

Nemat-Nasser et al. 2005; Chang et al. 2001; Wada et al. 2008; Huang et al. 2010; Otsuka et al. 1999; Song et al. 2006).

Several SMA applications include heating and cooling the SMAs periodically or occasionally. For instance, an air conditioning device was developed by Kirsch et al. (2017) based on the electrocaloric cooling effect of SMAs. Recent studies have demonstrated that the SMAs exhibit thermal hysteresis when subjected to mechanical/thermal loads (Ortin et al. 2006). This effect can clearly be observed in the continuous change in the SMAs strain values with each cycle of heating and cooling under constant stress. Many experiments illustrate that increasing the number of heating and cooling cycles leads to a gradual increase in the strain values. This increase in strain values per cycle reduces gradually, thus, it is relatively significant for early cycles in comparison to upper cycles (Padula et al. 2012). Additionally, thermal cycling also affects the thermal transformation temperatures, i.e., martensite finish (Mf), martensite start (Ms), austenite start (As), and austenite finish (Af), from martensitic phase to austenitic phase and vice versa. Moreover, during the thermal cycling, SMAs stain behavior and thermal transformation temperatures are affected by the maximum temperature at each cycle and the applied stress (Padula et al. 2008). Data from literature for 55NiTi strain-temperature relationship for 100 heating and cooling cycles with 165C maximum temperature is used to build Artificial Neural Network (ANN) models.

Modeling the material behavior provides an approach to describe this behavior mathematically through one set of equation(s). Initially, experiments are conducted to study the material behavior under different conditions, and based on this experimental data, models are developed. The significance of modeling is to be able to predict the

material behavior under conditions out of the experimental data since performing experiments on all the possible conditions is unachievable and experiments are time consuming or expensive procedures. For instance, modeling 55NiTi strain behavior under thermal cycling is used to predict strains that were not experimentally recorded at corresponding temperatures. Therefore, these models are essential for various applications, e.g., being the core of developing computer software capable of simulating the characteristics of the materials (Gu et al. 2015). Due to the complexity of the relationship between the 55NiTi strains and the four independent variables (temperature, cycle number, cycle state, and applied stress), using traditional regression modeling methods is insufficient as a large number of equations will be needed in that case to generate the model (which will likely be inaccurate). On the other hand, ANN captures relatively more scrupulous relationships with feasible equations (Ghaboussi et al. 1991). Consequently, this study demonstrates an application of ANN to develop accurate models that captures the strain behavior of 55NiTi as a function of four major factors, i.e., temperature, cycle state (heating or cooling), cycle number, and stress.

3.2 Methodology

3.2.1 Experimental Observation

The available experimental data for a 55NiTi rod, which was subjected to 50, 80, 150, and 300 MPa stresses, was used for the model development, testing, and validation. The strains are measured at every second for 100 cycles of heating and cooling. Each cycle starts with heating the material from 30°C to 165°C, and then the cycle ends with cooling the material back to 30°C. For both states of heating and cooling at a certain temperature, as the cycle number increases the strain value increases as well. This

increase is not proportional; the strain difference between two following cycles (often called open-loop strains) reduces gradually as the number of cycles increase. Each heating cycle consists of 3 stages. In the first stage, the strain value is subjected to a very small decline as the temperature increases. In the second stage, a huge drop in the strain value occurs in a small period of time and a small interval of temperature compared to the two other stages. Finally, the strain behaves as almost constant with a small change in its value. Similarly, each cooling cycle consists of 3 stages. The strain change is very small in the first and third stages, while a relatively huge increase occurs in the second stage.

3.2.2 Artificial Neural Network Modeling Architecture

The artificial neural network (ANN) structure consists of one input layer, two hidden layers, and one output layer. Each layer has a different number of neurons to achieve its main role. For the input layer, the number of neurons depends on the number of factors affecting the strain value, i.e., temperature (t), cycle number (n), cycle state (heating or cooling), and the stress (s). Instead of using 4 neurons, one for each factor, the number of neurons is reduced to 3 by representing the cycle state using the temperature sign, positive for heating and negative for cooling. For the two hidden layers, the number of neurons should provide enough complexity to the ANN, so it could learn the patterns in the training data. After trying different numbers of neurons for the hidden layers, the best choice of using 6 neurons for the first one and 4 neurons for the second provided more accurate predictions than other ANN structures. The output layer consists of one neuron, which is the strain value.

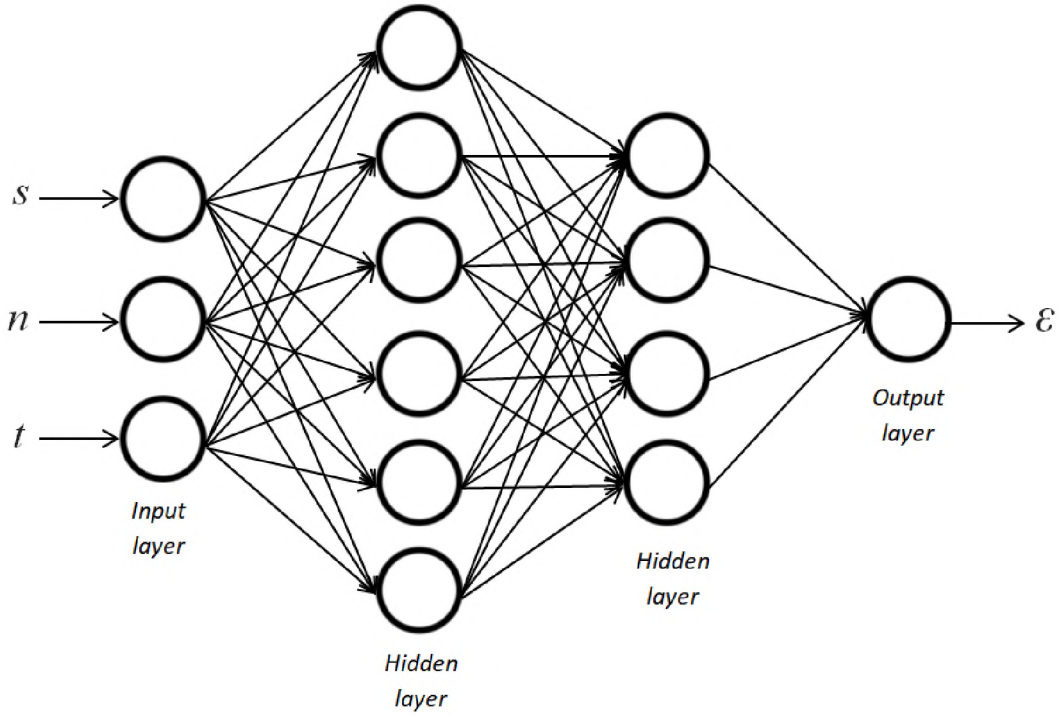


Fig. 7: The ANN architecture showing the input, hidden and output layers.

Part of the experimental data is used to train the ANN, while the other part is used to verify the ANN ability to predict values out of the training data. Since the behavior of the early cycles is different from the late cycles, cycles from the second to the 10^{th} cycle and cycles from the 80^{th} to the 90^{th} cycle were used in the training dataset. The last 10 cycles (cycles from 91^{st} to 100^{th}) were used as a test dataset to assess the ANN prediction for cycles out of the training range. Additionally, intermediate cycles were also used in the training dataset, including cycles from the 21^{st} to the 25^{th} cycle, the 30^{th} cycle, the 40^{th} cycle, cycles from the 45^{th} to the 50^{th} cycle, cycles from the 66^{th} to the 68^{th} cycle, and the 75^{th} cycle. Before training the ANN, input data scaling was required to accelerate the training process and improve the ANN stability by making the 3 inputs have a very close range. Therefore, the temperature input data was divided by 120, the cycle number was divided by 200 and the stress was divided by 100.

The ANN has a total of 48 weights (w) and 9 biases (b). Initially, random values in the range of 0.01 to 0.05 were assigned to these weights and biases. The mean square error function (J) is calculated using equation (1), which accounts for the number of points (N), the predicted strain (ε), and the experimental strain ($\tilde{\varepsilon}$). Then, the value of each weight and bias was updated by subtracting from it the product of a constant (α) and the first derivative of the mean square error function (J) with respect to that weight or bias (illustrated via Eq. (2) and Eq. (3)). After trying different values, the constant (α) was chosen to be 0.015. As choosing much larger values causes the training process to fail and the mean square error to increase, while choosing much smaller values slows the training process. The iteration of the process of updating the weights and biases values continues until the lowest possible mean square error value is obtained.

$$J = \frac{1}{N} \sum_{i=0}^{N} (\varepsilon_i - \tilde{\varepsilon}_i)^2 \qquad \text{Eq. 1}$$

$$w = w - \alpha \frac{dw}{dJ} \qquad \text{Eq. 2}$$

$$b = b - \alpha \frac{db}{dJ} \qquad \text{Eq. 3}$$

The sigmoid function (Eq. 4) shape is compatible with the strain behavior of a half cycle, either heating or cooling, as both have a relatively huge increase/decrease in a small domain while their values are almost constant out of that domain (Kciuk 2016). Therefore, the sigmoid activation function is used for the two hidden layers while the output layer does not have an activation function. The nodes of the first hidden layer (^1h) are calculated using Eq. 5, where x represents the 3 inputs (s, n, and t). The nodes of the

second hidden layer (^2h) are calculated using Eq. 6. Finally, the predicted strain (ε) is calculated using Eq. 7.

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad \text{Eq. 4}$$

$$^1h_i = \text{Sigmoid}\ (^1b_i + \sum_{j=1}^{3} {}^1w_{ij}\, x_j) \quad \text{Eq. 5}$$

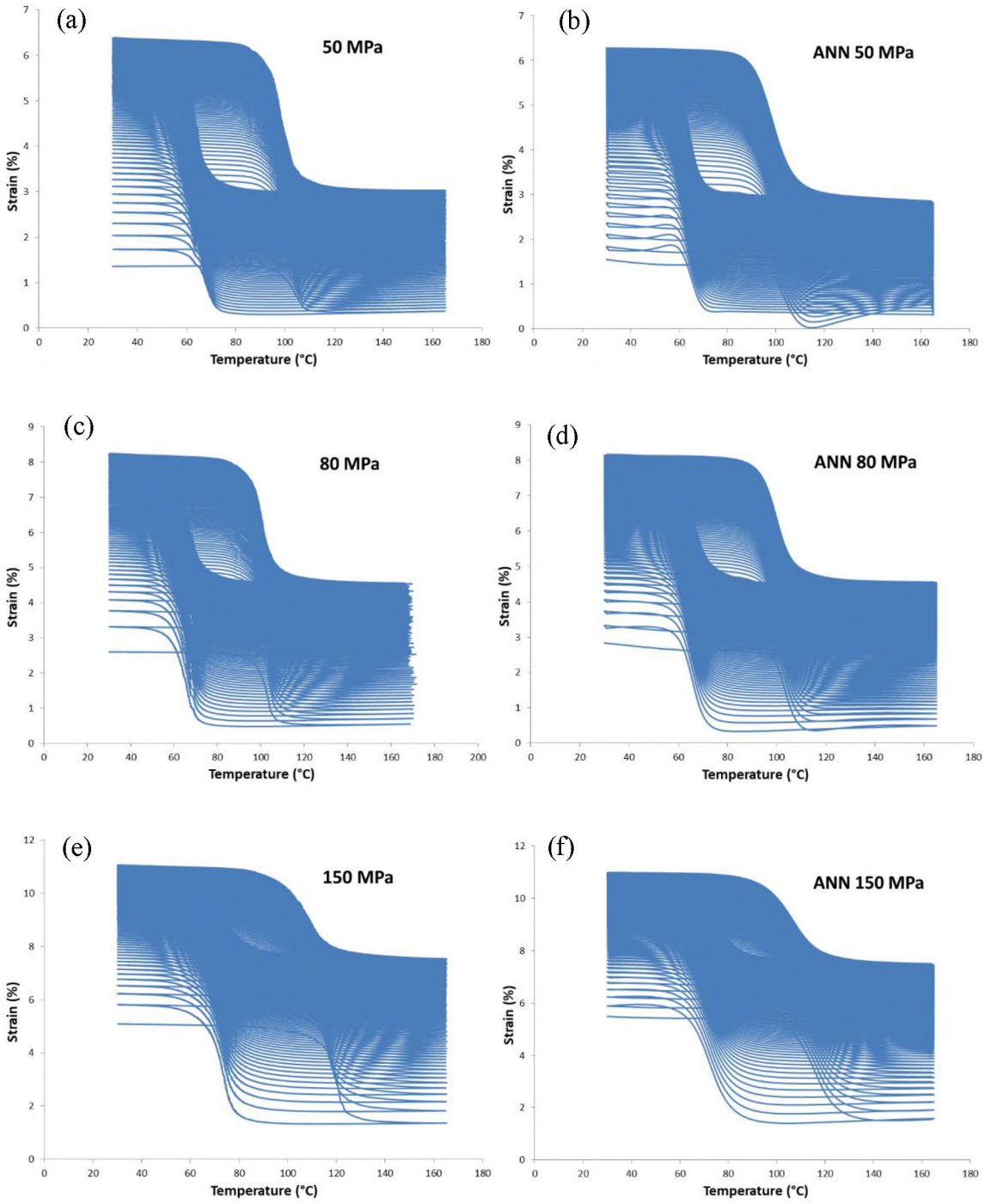
$$^2h_i = \text{Sigmoid}\ (^2b_i + \sum_{j=1}^{6} {}^2w_{ij}\, {}^1h_j) \quad \text{Eq. 6}$$

$$\varepsilon = (^3b + \sum_{i=1}^{4} {}^3w_i\, {}^2h_i) \quad \text{Eq. 7}$$

3.3 Results and Discussion

Two main approaches have been followed to model the available 55NiTi strain data. First, four ANNs were trained individually for each stress. Second, training one Artificial neural network (^{2}ANN) using the combined data of 50, 80, 150 MPa stresses. For the first approach, in order to assess the individual models' performance, a comparison between the experimental data and the models was created as shown in Figure 8. Although approximately only 37% of the available data for each stress were used to create the four ANNs, the models can predict the other 63% of the data sufficiently. Furthermore, all four models capture the temperature effects adequately, considering that the models adhere to the same three stages of temperature effects, discussed previously in the experimental observation section, for both heating and cooling. Additionally, they also comply with the cycle number effects, as it is seen that the open-loop strains reduce gradually per increasing cycling, alongside a parallel shift in the transformation temperatures between the martensitic phase and the austenitic phase (Mf, Ms, As, and Af). At higher stress levels, even for the same cycle number, there is an

increase in transformation temperature width, i.e, temperature range between (Mf) and (Ms) and between (Af) and (As). Subsequently, the ANN fits the experimental curves for cases at higher stresses more accurately than those at lower stresses as observed in Figure 8.



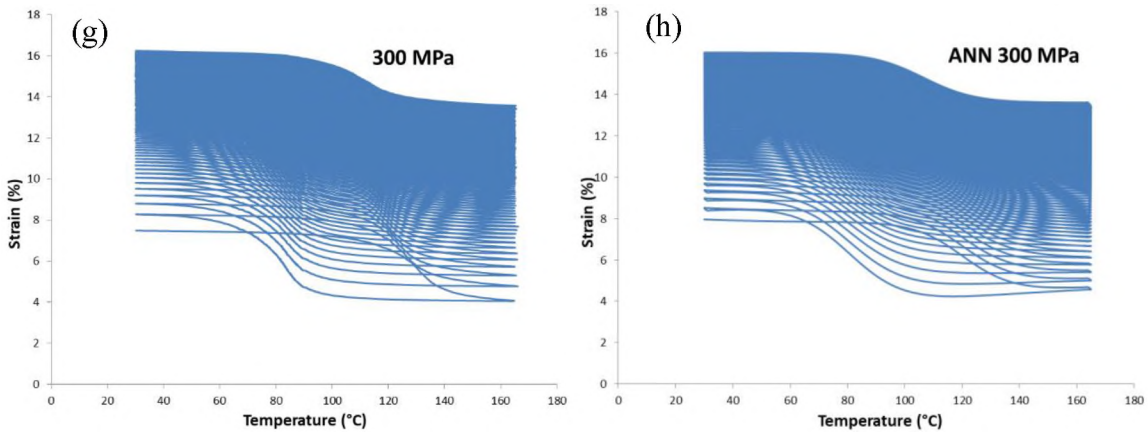


Fig. 8: Experiment versus ANN model for the temperature-strain relationship first 100 cycles. (a) 50 MPa experiment. (b) 50 MPa ANN. (c) 80 MPa experiment. (d) 80 MPa ANN. (e) 150 MPa experiment. (f) 150 MPa ANN. (g) 300 MPa experiment. (h) 300 MPa ANN.

Table 1. Final weights and biases for all ANN.

	50MPa ANN	**80MPa ANN**	**150MPa ANN**	**300MPa ANN**	**^{2}ANN**
$^{1}w_{11}$	4.46078	0.02989	0.15248	-1.19207	0.12517
$^{1}w_{12}$	-2.07098	14.11064	17.90138	24.98721	20.67097
$^{1}w_{13}$	-12.78059	-0.50160	-0.03633	-0.44269	-0.01490
$^{1}w_{21}$	0.22382	6.50447	-1.27463	4.03993	4.47035
$^{1}w_{22}$	11.41813	0.72592	9.72528	-4.10383	8.14998
$^{1}w_{23}$	-2.10885	-15.70924	-0.09277	-13.21946	-0.23621
$^{1}w_{31}$	-0.67973	-3.75158	-0.52341	-0.27932	0.14820
$^{1}w_{32}$	5.49686	-0.39564	9.56641	39.16377	-2.74369
$^{1}w_{33}$	1.78016	8.32478	2.03070	-1.69164	22.87021
$^{1}w_{41}$	-0.61832	0.15868	0.11190	-0.24865	5.51639
$^{1}w_{42}$	5.66682	13.49340	10.97897	36.76519	2.20805
$^{1}w_{43}$	-1.35072	0.37106	-2.24087	0.98767	3.28068
$^{1}w_{51}$	3.93916	4.06836	4.07459	2.47126	4.66500
$^{1}w_{52}$	-1.68896	1.37893	2.69810	0.87951	4.19276
$^{1}w_{53}$	17.01518	15.29609	14.82308	12.30748	-0.48399
$^{1}w_{61}$	0.64684	-1.56148	6.09215	-1.80411	1.91401
$^{1}w_{62}$	8.99062	6.18713	-0.65242	17.08351	-2.29609
$^{1}w_{63}$	2.13883	-0.11491	-13.27632	-0.02197	-15.59891
$^{2}w_{11}$	-1.23814	10.93917	3.43185	1.86099	7.72046
$^{2}w_{12}$	6.80439	-0.43851	4.88566	3.36375	3.49644
$^{2}w_{13}$	7.26360	1.60838	1.39802	2.78388	-0.47960
$^{2}w_{14}$	3.82898	8.62303	1.39261	3.47061	1.19878
$^{2}w_{15}$	-3.35026	-0.48237	0.53204	3.18513	4.95419
$^{2}w_{16}$	6.95576	4.56195	0.59102	2.98523	0.01103
$^{2}w_{21}$	-0.49627	1.28885	-1.72456	4.63214	6.30712
$^{2}w_{22}$	1.94989	1.47697	-0.51261	2.41612	-0.93152

${}^{2}w_{23}$	2.15678	-0.03297	-1.94983	-2.50866	0.72057
${}^{2}w_{24}$	3.05178	1.24222	-0.10513	0.33194	-0.72056
${}^{2}w_{25}$	0.34515	1.83317	8.21398	4.45276	0.56547
${}^{2}w_{26}$	1.39660	1.96141	8.12789	2.61818	3.94939
${}^{2}w_{31}$	8.40351	2.13192	8.54794	1.77017	1.45934
${}^{2}w_{32}$	-1.41622	-0.08574	3.13570	2.35603	5.41102
${}^{2}w_{33}$	1.86853	0.08554	3.60782	1.67790	0.76862
${}^{2}w_{34}$	3.56511	2.47275	3.48072	1.97095	1.55386
${}^{2}w_{35}$	1.37001	-0.09227	1.07028	2.82865	0.89959
${}^{2}w_{36}$	0.55551	5.24740	0.60549	6.72323	-0.80066
${}^{2}w_{41}$	0.31142	-2.19311	0.04385	5.07935	5.25910
${}^{2}w_{42}$	1.36442	9.85286	3.33441	1.17788	2.22510
${}^{2}w_{43}$	5.26412	-2.82527	1.86138	4.70952	5.77771
${}^{2}w_{44}$	0.67559	-0.55042	2.52765	4.82321	0.59066
${}^{2}w_{45}$	10.97443	12.32908	0.74921	2.38528	3.83009
${}^{2}w_{46}$	-3.02307	-2.55858	0.93826	2.77030	-3.73091
${}^{3}w_{1}$	7.10065	8.97381	3.02368	3.14147	2.97071
${}^{3}w_{2}$	2.53550	2.54267	3.40585	2.65801	5.50398
${}^{3}w_{3}$	3.41577	3.25841	6.97607	4.50101	3.69948
${}^{3}w_{4}$	3.08513	2.82804	2.73942	7.53351	4.25681
${}^{1}b_{1}$	8.94156	0.06486	0.13499	-0.36069	0.63191
${}^{1}b_{2}$	0.36763	8.08809	-0.86309	1.34998	-7.96668
${}^{1}b_{3}$	-1.35945	-4.66698	-0.30894	-0.04311	13.22931
${}^{1}b_{4}$	-1.35663	0.13085	0.04793	-0.08622	-4.53917
${}^{1}b_{5}$	7.86832	5.09420	2.73973	0.85542	-3.26913
${}^{1}b_{6}$	1.25369	-1.98060	4.04143	-0.61637	11.81353
${}^{2}b_{1}$	-6.73639	-8.98108	-6.51384	-13.48302	-12.32539
${}^{2}b_{2}$	-4.97765	-5.06764	-10.72059	-11.14069	-7.75558
${}^{2}b_{3}$	-7.45550	-4.58758	-9.18573	-6.53610	-8.07759
${}^{2}b_{4}$	-7.52931	-14.18184	-8.64630	-5.46621	-7.93253
${}^{3}b$	-9.56057	-8.70069	-3.37480	-0.80655	-3.90613

In order to emphasize the individual ANN model's capability of predicting the strain behavior, a comparison was made in Figure 9 between the experimental and the ANN predicted strain at martensite (ε_M), i.e., the strain at the beginning of the cycle) and at 165°C austenite (ε_A) for the first 100 cycles. For 50, 80, 150, and 300MPa stresses, the maximum differences between the experimental and the predicted ε_M are 0.22, 0.23, 0.4, and 0.48%, respectively, occurring in the 2nd cycle. Similarly, the maximum differences for ε_A also occur in the 2nd cycle having the values of 0.11, 0.07, 0.19, and 0.62%,

respectively. The average differences for the first 100 cycles between the experimental and the predicted data for ε_M are 0.048, 0.051, 0.083, and 0.11% respectively and for ε_A are 0.05, 0.024, 0.06, and 0.13%, respectively. Clearly, the ANNs are able to predict ε_M and ε_A adequately.

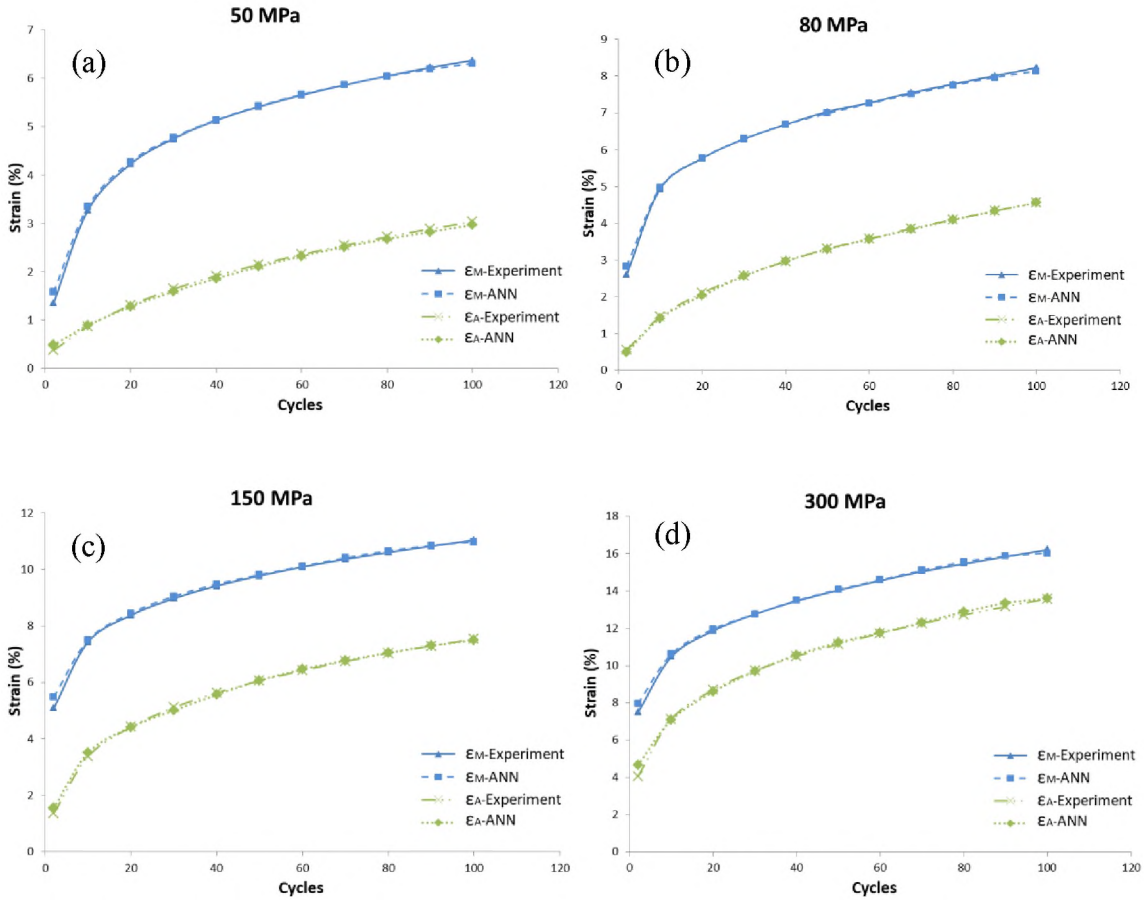
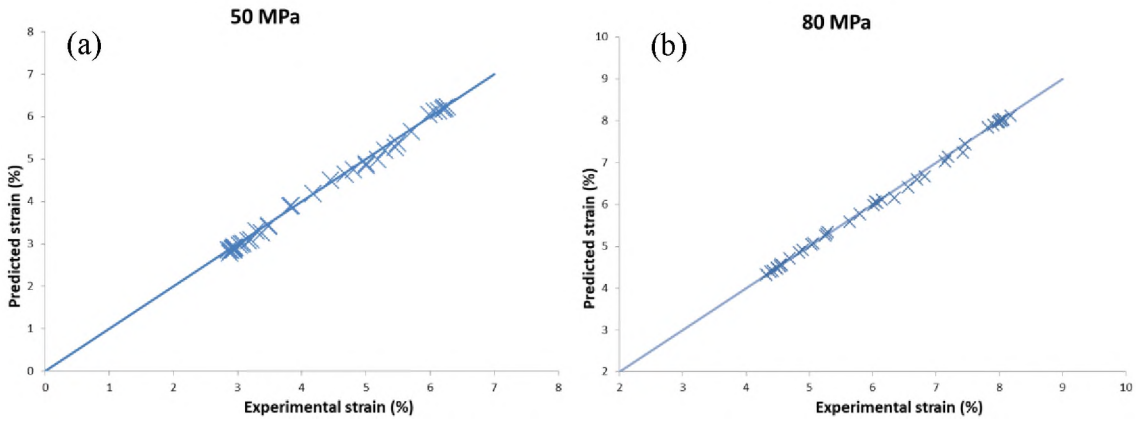


Fig. 9: Relationship of the strain at martensite (εM) and austenite (εA) with the first 100 cycles for the case of (a) 50 MPa, (b) 80 MPa, (c) 150 MPa and (d) 300 MPa.

A random test dataset from the 91st to the 100th cycles of each stress was used to evaluate the ANN capability of predicting the strain for upper cycles which are outside the training dataset. Figure 10 illustrates the relationship between the ANN predicted values and the experimental values for the 50, 80 150, 300 MPa stresses. The difference between the ANN predicted values and the experimental values is less than 0.1% for

82.5%, 85%, 92.5%, and 75% of the predicted data for the four stresses, respectively, while the maximum error for each stress was 0.198%, 0.191%, 0.156%, and 0.312%, respectively. These numbers reflect the probable accuracy of the ANNs prediction of upper cycles (cycles beyond the 100th). Figure 11 illustrates the predicted strain for cycles from 101st to 200th for the four stress cases. One can observe that upper cycles continue with the same strain behavior as the first 100 cycles for all stresses, since increasing the cycle number reduces the difference between two succeeding cycles and increases the transformation temperatures Mf, Ms, As, and Af. It is also observed that the difference between the maximum and the minimum strain of each cycle decreases with increasing the applied stress or increasing the cycle number.



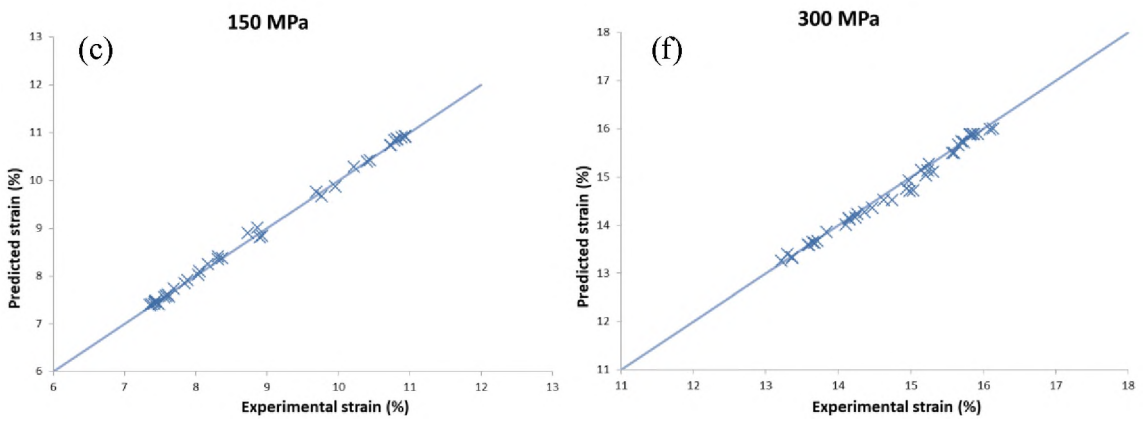


Fig. 10: Relationship between the experimental and the predicted values for random data in cycles from 91st to 100th for the case of (a) 50 MPa, (b) 80 MPa, (c) 150 MPa and (d) 300 MPa.

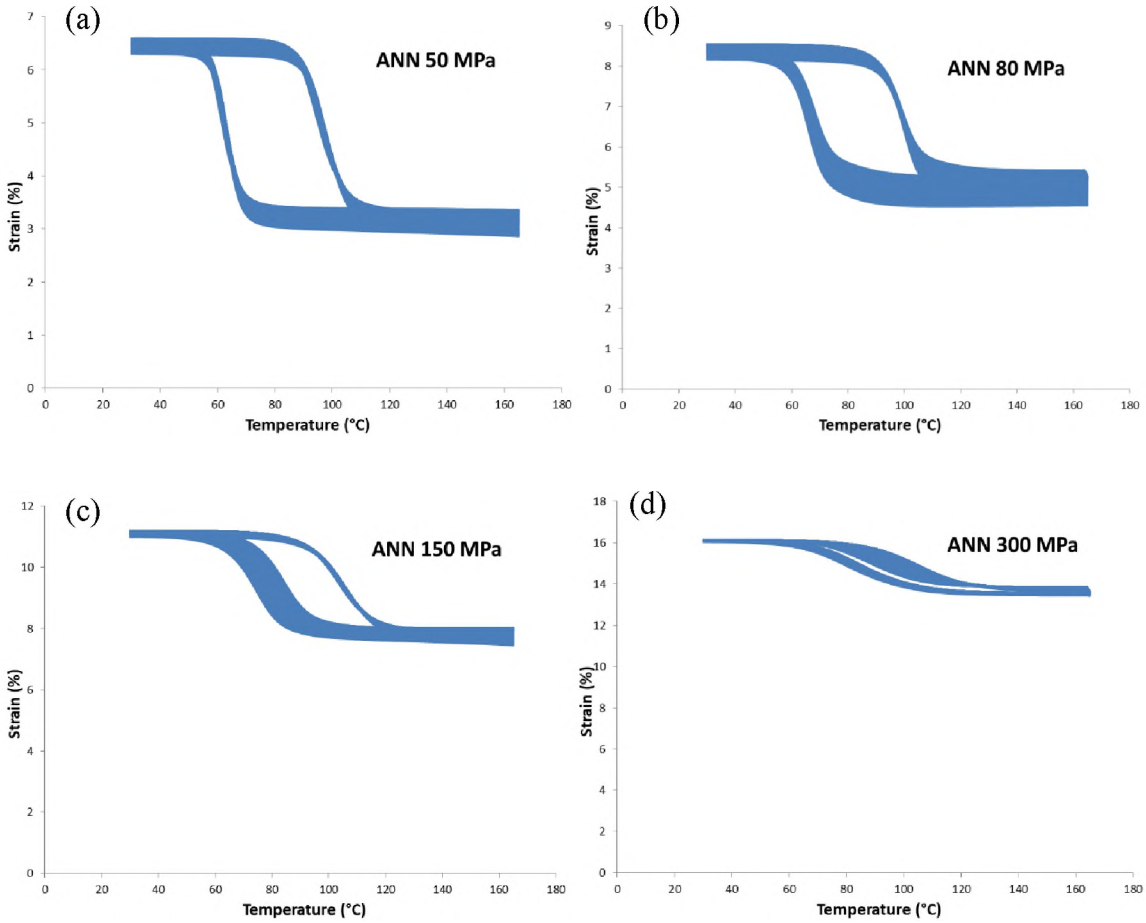
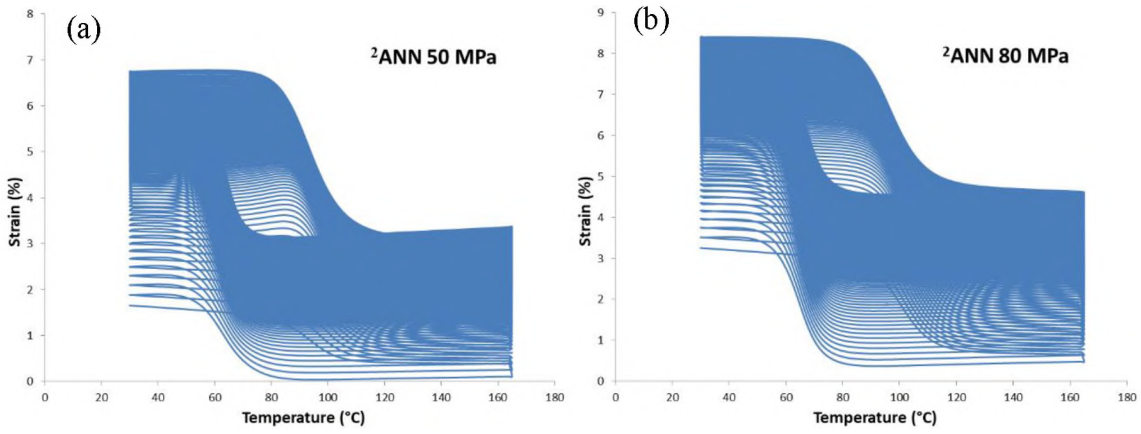


Fig. 11: ANN predicted temperature-strain relationship for cycles from 101st to 200th for the case of (a) 50 MPa, (b) 80 MPa, (c) 150 MPa and (d) 300 MPa.

Another approach was followed by training one ANN (^{2}ANN) using the available data of three different stresses (50, 80, and 150 MPa). Although the accuracy of this ^{2}ANN is less than the previous individual ANNs, as can be observed by comparing Figure 8 with Figure 12, it provided an advantage by predicting the strain behavior of 55NiTi under other stresses. For example, the ^{2}ANN was used to predict the strain under 70, 100, 120, and 135 MPa stresses (as illustrated in Figure 13). On one hand, the predicted models in this figure have some errors. For instance, the strain at the end of the heating cycle and the start of the cooling cycle for some cycles do not have the same value. On the other hand, they provide a very close estimate for the maximum and the minimum strain of each cycle and for the transformation temperatures Mf, Ms, As, and Af.



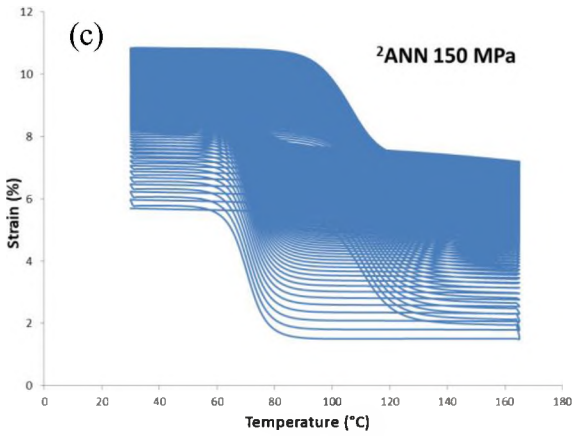


Fig. 12: [2]ANN model for the temperature-strain relationship first 100 cycles. (a) 50 MPa ANN. (b) 80 MPa ANN. (c) 150 MPa ANN.

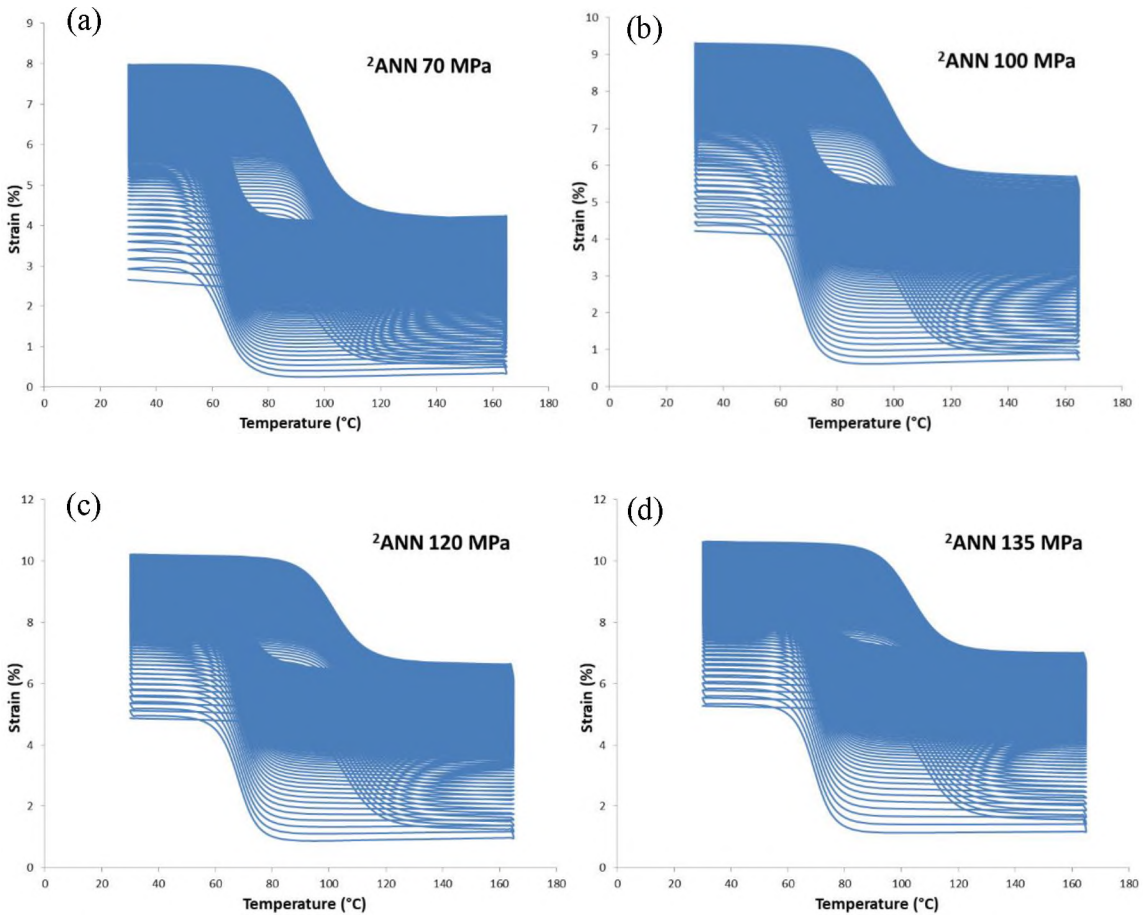


Fig. 13: [2]ANN model for the temperature-strain relationship first 100 cycles. (a) 70 MPa ANN. (b) 100 MPa ANN. (c) 120 MPa ANN. (d) 135 MPa ANN.

CHAPTER IV

REINFORCED CONCRETE BOND STRENGTH MODEL

4.1 Introduction

The bond strength of a reinforced concrete (RC) describes the ability to transfer the axial force from the reinforcement steel to the surrounding concrete effectively with no or very small slip. Friction and adhesion, which primarily depend on the materials' properties, are the two main factors that influence bond strength. Many experiments have been conducted in order to form an understanding of the different factors affecting the value of the bond strength and how this strength can be estimated (Juarez et al. 2011; Hong et al. 2012; Cheng et al. 2018; Dancygier et al. 2010; Lachemi et al. 2009; et al. Zhao 2013).

According to previous studies, the dominant factors are corrosion level, concrete compressive strength, and concrete cover. Fang et al. (2004) reported that the effects of corrosion level on bond strength differ can be presented in two main stages. During the first stage, which ends with a corrosion level between 2% and 4%, increasing the corrosion level comes hand in hand with a relatively small increase in the bond strength. On the other hand, a dramatic decrease in the bond strength occurs in the second stage when the corrosion level increases to be more than 4%. Lan Chung et al. (2008)

demonstrated that the bond strength was initially increased with the increase in corrosion up to a certain maximum value, which depends on other factors such as concrete cover and compressive strength, but gradually decreased as the corrosion level increases. The reason behind increasing the bond strength, in the early stage, lies in increasing the friction between the steel and the concrete by increasing the surface roughness (Toloei et al. 2013; Sajid et al. 2018). However, the second stage demonstrates a decrease in the contact surface area between the steel and the concrete which in turn decreases the bond strength. (Cheng et al. 2018; Tondolo 2015).

Abosrra et al. (2011); Dancygier et al. (2010); Albitar et al. (2017) showed that increasing the compressive strengths (f_c') of concrete increases the bond strength. Price (1951) reported that the bond strength and the compressive strength are increasingly proportional up to f_c' = 20 MPa and thereafter, the compressive strength of concrete showed little or no effect on the bond strength. On the other hand, effects of concrete cover, in the range of 0.5 to 7.5 of the bar diameters (d) were studied by Torre-Casanova et al. (2013). The results showed that increasing the cover up to 4.5d leads to an increase in the bond strength, whilst concrete cover greater than 4.5d records no effects.

Many models have been developed in previous studies that capture or predict the bond strength as a function of the corrosion level, the ratio between the concrete cover and the bar diameter, and the concrete compressive strength (Lin and Zhao et al. 2016; Chung et al. 2017; Feng et al. 2016; Lee et al. 2002; Darwin 2005; Albitar et al. 2017; Kivell 2012; Lin et al. 2016; Yafei et al. 2017). For instance, Yalciner et al. (2012) developed different models using statistical regression to predict the bond strength as a function of the cover-to-bar-diameter ratio, crack widths, corrosion levels, and concrete

compressive strength. These models were built on pullout test results conducted on 90 exclusive specimens, that there had no two identical specimens in the aspects of concrete compressive strength, concrete cover, and corrosion level. These test results from Yalciner et al. (2012) are also used in this research as a training dataset in building ANN models.

The ANN captures the patterns in the training dataset accurately and creates more adequate models than other methods such as regression; it also provides practical equations in terms of number and complexity (Basheer et al. 2000; Gonzalez-Fernandez et al. 2019). Therefore, ANN was developed as a function of three key factors, i.e., corrosion level (from 0% to 20%), compressive strength (from 23 to 51 MPa), and concrete cover (ranging between 15 and 45mm). Since the degree of accuracy of an ANN model is influenced by the choice of activation function, two non-linear activation functions: Rectified Linear unit (ReLu) and Sigmoid were compared. Moreover, statistical regression, non-linear and linear equations were also derived, and their results were compared with the ANN model.

4.2 Methodology

4.2.1 Experimental Observation

The formulation and capability of the generalized ANN equations are dependent on the accuracy of the set of training data points used. In this study, the ANN model is developed using the experimental results from the tensile pullout tests on 90 different specimens (Yalciner et al. 2012). These specimens had varying levels of corrosion, magnitudes of concrete compressive strengths, and cover values, but the same bond length of 50mm. In the model development, the training points were randomly selected to

constitute about 90 percent of the whole dataset whilst the remaining 10 percent of the data was used for model verification and validation.

All test points were placed under either corroded or non-corroded specimen groups. A third of the test sample had a 15 mm cover, while the second third had a cover of 30 mm. The remaining 30 cases belonged to a concrete cover of 45 mm (which was the largest cover considered in this study). Specimens belonged to either a compressive strength of f_c' = 23 MPa or f_c' = 51 MPa. It is worth mentioning that Concha and Oreta et al. (2019) earlier investigated the cases of concrete covers between 60 to 80 mm range and compressive strengths between 21 to 35 MPa. Details on the level of corrosion for specific samples were not reported in their study for the 108 specimens they used.

As seen in many tests, the effect of concrete cover or compressive strength on bond strength changes as a function of the corrosion level. Figure 14 shows the variation of the experimental bond strength with corrosion under different values of concrete cover and compressive strength. In the two scenarios, i.e., the case of 23 MPa with 30 mm cover (in Figure 14a) and the case of 51 MPa with 15 mm cover (in Figure 14b), there is a gradual reduction of the bond strength with the corrosion. The effect of corrosion on the bond strength is more prominent when the corrosion levels were between 1% and 5%. In particular, an approximate reduction of 11.34 and 12.2 MPa occurs between these two states of corrosion in Figure 14a and b, respectively. There is no significant change observed for cases where the corrosion level was less than 1% or greater than 7.5%. Figure 15a illustrates that at the low (2%) corrosion level, increasing the cover leads to slight growth in bond strength. However, at the high (4%) corrosion level, increasing the cover causes a drop in the bond strength value.

In this work, charts were made between bond strength and corrosion level under different cover and compressive strength conditions. These charts were compared with other existing data from some of the previous studies mentioned earlier in the introduction (section 4.1) to confirm the effects of corrosion. A similar procedure was applied when considering the effect of concrete cover. Determining the relationship between the compressive strength and the bond strength could not be done with the only two points, i.e., the 23 MPa and 51 MPa, available in Yalciner et al. (2012); therefore, additional data from the earlier work of Price (1951) on factors influencing concrete bond and shear strength was used.

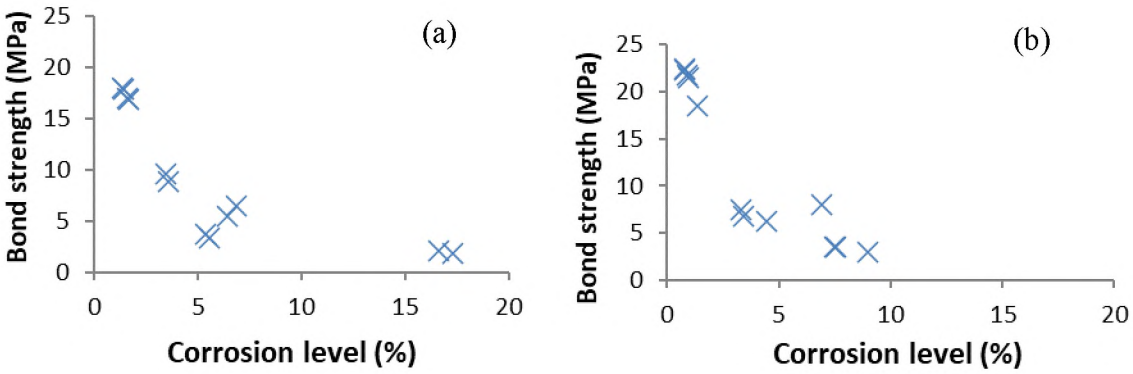


Fig. 14: Effect of corrosion level on bond strength. (a) 23 MPa compressive strength and 30 mm cover. (b) 51 MPa compressive strength and 15 mm cover.

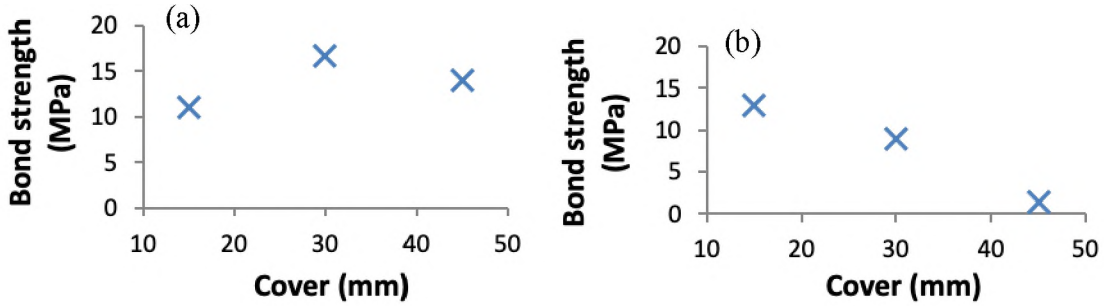


Fig. 15: Effect of concrete cover on bond strength. (a) 23 MPa compressive strength and 2% corrosion level. (b) 23 MPa compressive strength and 4% corrosion level.

4.2.2 Artificial Neural Network Modeling Architecture

The artificial neural network (ANN) application comprises of 3 interconnected layers, i.e., the input, hidden, and output layers (see Figure 16). Here, the input layer takes in three nodes which are the corrosion level (cl), the concrete cover (cc), and the compressive strength (cs). The number of neurons in the hidden layer was determined by trial-and-error, with 3 neurons (nodes) found to generate an accurate prediction. The Rectified linear unit (ReLU) (Eq. 8) and sigmoid (Eq. 9) activation functions were used for the hidden layer neurons. A comparison of the performance of these two model activation functions will be presented in the upcoming section 4.3. The output layer takes in one node since the bond strength (τ_u) is the only output needed. After testing linear and exponential activation functions for this layer, the exponential activation function was found to give a higher rate of convergence. To determine the weights and biases for the neural network, an error function was formed by squaring the difference between the predicted values and the experimental values. The error function must be as low as possible to give a more accurate prediction.

$$\text{ReLU}(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \quad \text{Eq. 8}$$

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad \text{Eq. 9}$$

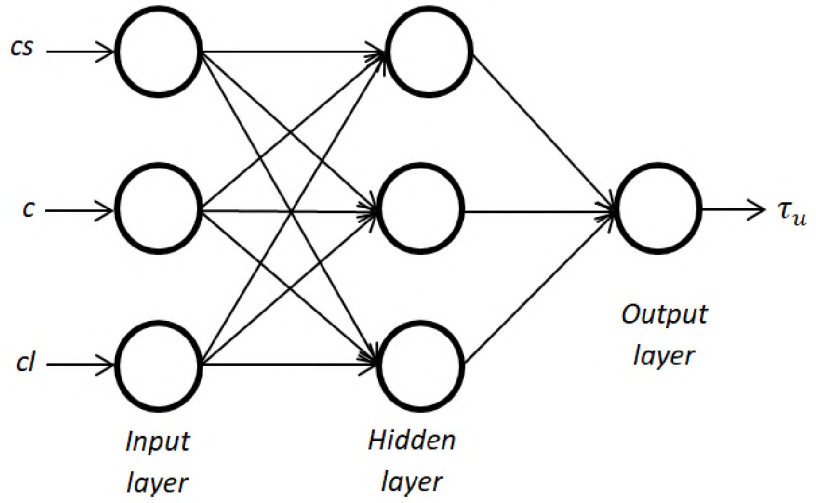


Fig. 16: The ANN architecture showing the input, hidden, and output layers.

Initial values for the weights and biases were generated randomly by the Matlab software. These values are automatically updated by reducing the previous quantities by the product of their first derivative and a constant obtained through trial-and-error. The iteration continues until the lowest possible error is obtained. The final equations for ANN(ReLU) (Eq. 10-15) are then generated for the hidden and output layers with weights (${}^1w$, ${}^2w$) and biases (${}^1b$, ${}^2b$) that minimize the error function. Using these equations, the prediction of the bond strength (τ_u) can be obtained at different corrosion levels (cl) in %, concrete cover (cc) in the units of mm, and the compressive strength (cs) in MPa units. A non-linear formula obtained from statistical regression is also stated in equation 16 to characterize the bond strength as a function of corrosion, cover, and compressive strength as well. Comparisons will be made between this equation and the ANN models in the upcoming section 4.

$$ {}^1w = \begin{bmatrix} 0.0223 & 0.0632 & 1.5199 \\ -0.0318 & 0.0598 & 0.2444 \\ 0.0559 & 0.0251 & 0.4758 \end{bmatrix} \quad \text{Eq. 10} $$

$$ {}^1b = [2.4334 \quad 1.1761 \quad 3.6675] \quad \text{Eq. 11} $$

$$^{2}w = [0.6355 \quad 1.5178 \quad 1.2695] \qquad \text{Eq. 12}$$

$$^{2}b = 1.7768 \qquad \text{Eq. 13}$$

$$h_i = \text{ReLU}\,(^{1}w_{i1}\,cs + {}^{1}w_{i2}\,cc + {}^{1}w_{i3}\,cl + {}^{1}b_i) \qquad \text{Eq. 14}$$

$$\tau_u = 0.66^{-0.7052\left(\sum_{i=1}^{3} 2w_i h_i + 2b\right) - 4.7343)} \qquad \text{Eq. 15}$$

The bond strength of the non-corroded samples depends on only two factors, i.e., the concrete cover and compressive strength of the concrete. Amongst the test data, there were only 18 non-corroded samples available. Changing one of these factors does not affect the relationship between the other factor and the bond strength. Therefore, statistical regression (which can be considered as a process of fitting data) could be used to generate a linear equation to predict the bond strength. In this study, equation 17 was formulated to predict the bond strength as the sum of the effects of the two independent parameters, i.e., the cover and the compressive strength. This equation has three constants which were determined to generate the most accurate prediction by minimizing the mean squared error function. It is worth mentioning that the predictions from the ANN model and this linear regression equation were the same in this case, thus, only the results from the linear regression will be shown.

$$\tau_u = 6.25414 - 0.00415cs - 0.11766cc + (27.04475)\,(0.98051)^{28.23021cl - 7.66377} \qquad \text{Eq. 16}$$

$$\tau_u = 0.325397\text{cs} + 0.21778\text{cc} - 1.58413. \qquad \text{Eq. 17}$$

4.3 Results and Discussion

Mean squared error (J) and the coefficient of determination (R^2) are the two main characteristics used to assess the performance and accuracy of the ANN activation functions used in this work. The higher the value of R^2, the better the model

predictability. Other important features such as training time and the number of iterations can also be used. As shown in Table 2, the values of J and R^2 from the two functions were very close, with a small difference in favor of the ANN(ReLU). The training dataset used consisted of 63 corroded samples, hence the training time for ANN(ReLU) and ANN(sigmoid) is relatively small.

Table 2. ANN(ReLU) and ANN(sigmoid) comparison.

ANN	J	R^2	training time (s)	# of iterations
ANN(ReLU)	2.4099	96.89%	42	642528
ANN(sigmoid)	2.8762	96.28%	17	353406

The experimental and model-predicted bond strength values for the corroded specimens are shown in Table 3. All these samples differed in the level of corrosion (which was the predominant factor affecting the bond strength values). It is seen that several of the ANN-predicted values closely matched that of the experiment. Figure 17 shows a relationship between the bond strength and corrosion for the model and counterpart experiment for the case of 23 MPa under different concrete covers. To put the predictive capability of the present ANN models in the right perspective, these values are also shown against the predictions from other analytical models in the literature as well as the nonlinear regression model (developed in the present study). It is seen that the new ANN models can characterize the non-monotonic variation of the bond strength with corrosion. For corrosion levels lesser than 3% in Figure 17a (when the concrete cover was 15 mm), the ANN and the nonlinear regression models predicted higher values than those observed in the experiment. Meanwhile, for the cases of 30 and 45 mm covers (in

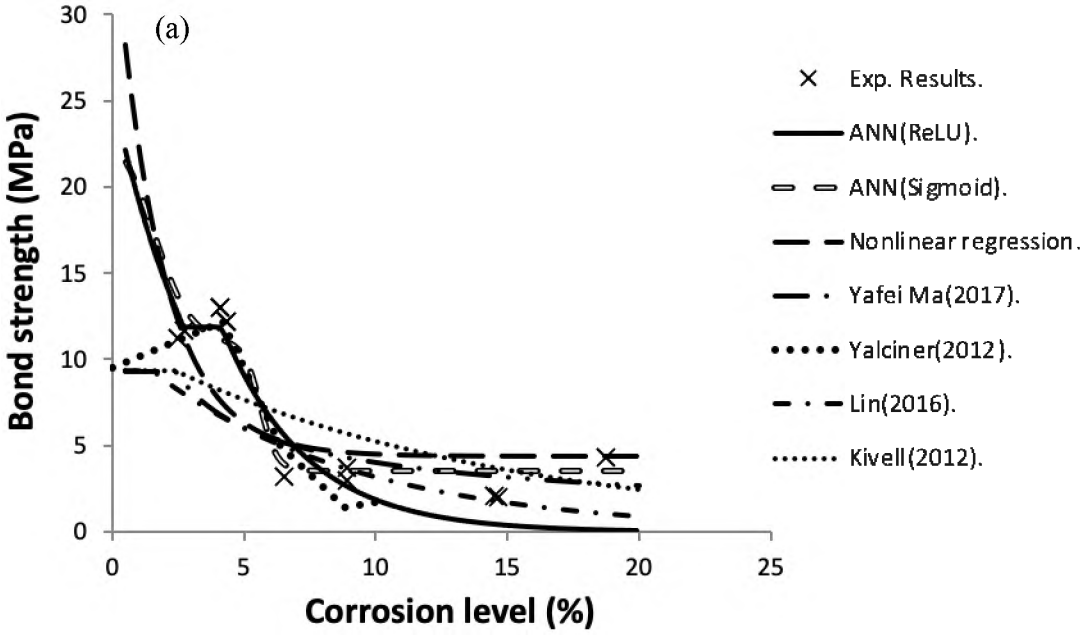
Figure 17b and c, respectively), the ANN(ReLU) results were very close to that of the test in the lesser levels of corrosion.

Table 3. Experimental and ANN(ReLU) predicted values of the bond strength for all specimens.

Sample #	Experimental bond	Model-predicted bond	Error	Sample #	Experimental bond	Predicted bond	Error
1	11.2	12.355	0.2427	37	22.3	22.45867	0.1587
2	11.7	12.0359	0.1295	38	22.4	22.26873	0.1313
3	13	12.0359	0.3295	39	21.7	21.64712	0.0529
4	13	12.0359	0.8071	40	21.5	21.30086	0.1991
5	12.2	11.56637	0.9567	41	18.5	17.80152	0.6985
6	12.2	11.53337	0.5481	42	7.5	7.887259	0.3873
7	3.2	6.185662	1.3075	43	6.8	7.735134	0.9351
8	3.7	3.124296	1.4293	44	6.3	6.411523	0.1115
9	3	3.124296	1.7800	45	8	4.169759	3.8302
10	2.1	0.626943	3.2476	46	3.5	3.762816	0.2628
11	2	0.602354	1.9012	47	3.5	3.709894	0.2099
12	4.3	0.187164	1.6363	48	3	2.556746	0.4433
13	18	18.2427	0.2427	49	23.8	25.23418	1.4342
14	17.9	18.02951	0.1295	50	3.9	24.88822	20.988
15	17	16.67051	0.3295	51	23.5	23.87853	0.3785
16	16.9	16.09286	0.8071	52	23.4	23.87853	0.4785
17	9.6	8.643265	0.9567	53	14	15.56505	1.5650
18	8.9	8.351876	0.5481	54	13.8	15.42246	1.6225
19	3.7	5.007476	1.3075	55	4.2	5.391433	1.1914
20	3.3	4.72929	1.4293	56	1.7	4.795316	3.0953
21	5.5	3.719972	1.7800	57	6.2	4.426547	1.7735
22	6.5	3.252414	3.2476	58	2.4	3.837141	1.4371
23	2.1	0.198776	1.9012	59	5.9	1.135742	4.7643
24	1.8	0.163669	1.6363	60	31.6	30.58958	1.0104
25	18.9	19.28386	0.3838	61	26.2	30.17019	3.9702
26	17.9	19.13333	1.2333	62	26.9	30.17019	3.2702
27	18	19.13333	1.1333	63	31.7	30.17019	1.5298
28	19.1	19.0585	0.0414	64	31	29.34859	1.6514
29	18.3	17.9706	0.3294	65	30.8	29.21385	1.5861
30	18.2	17.69113	0.5088	66	7.6	8.014906	0.4149
31	13.7	13.34227	0.3577	67	6.1	6.420102	0.3201
32	13.4	12.87994	0.5200	68	3.9	3.453329	0.4467
33	12.4	8.742726	3.6572	69	3.4	3.20548	0.1945
34	1.3	4.544842	3.2448	70	3.0	2.900367	0.0996
35	1.3	4.544842	3.2448				
36	3.2	2.249964	0.9500				

One observes that the degree of accuracy for the non-linear regression model changes from one test condition to the other. For instance, while the values obtained from the regression model were comparable to those in the experiment in the case of Figure 17b, the results produced for the cases of Figure 17a and c, are significantly different between that model and the tests for low corrosion levels. Also, it is seen that the bond strength predictions from the ANN(ReLu) and the ANN(Sigmoid) become widely different as the corrosion level rises (attesting to the superiority of the ReLU over the Sigmoid).

Figure 18 shows the case of the 51MPa for all concrete covers. Here also, the ANN(ReLU) model was able to capture several aspects of the bond strength under different corrosion levels and concrete covers. For instance, for the case of 45 mm cover, the prediction error ranged between 0.0996 and 3.97 (which is insignificant considering the limited data used here).



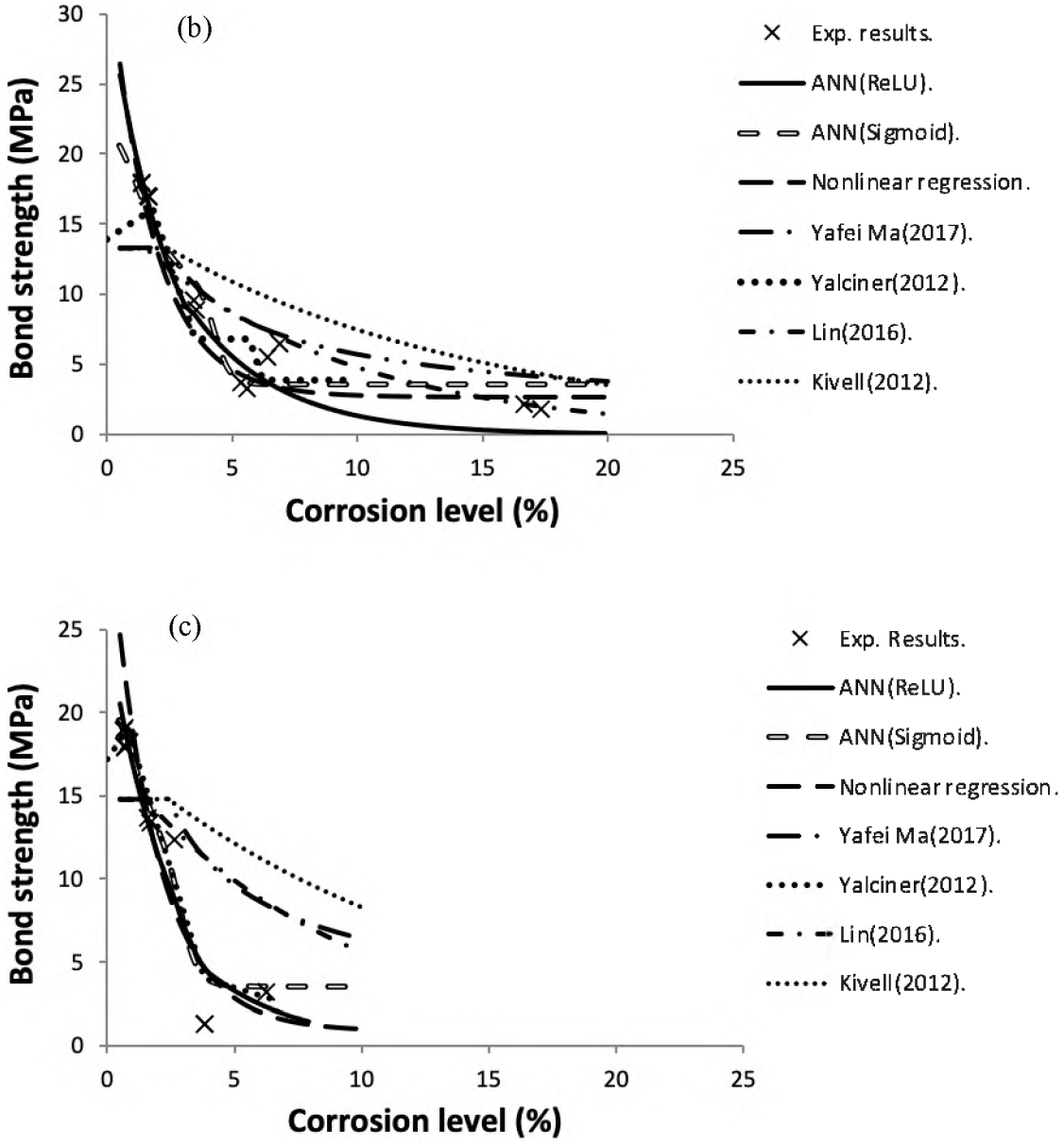
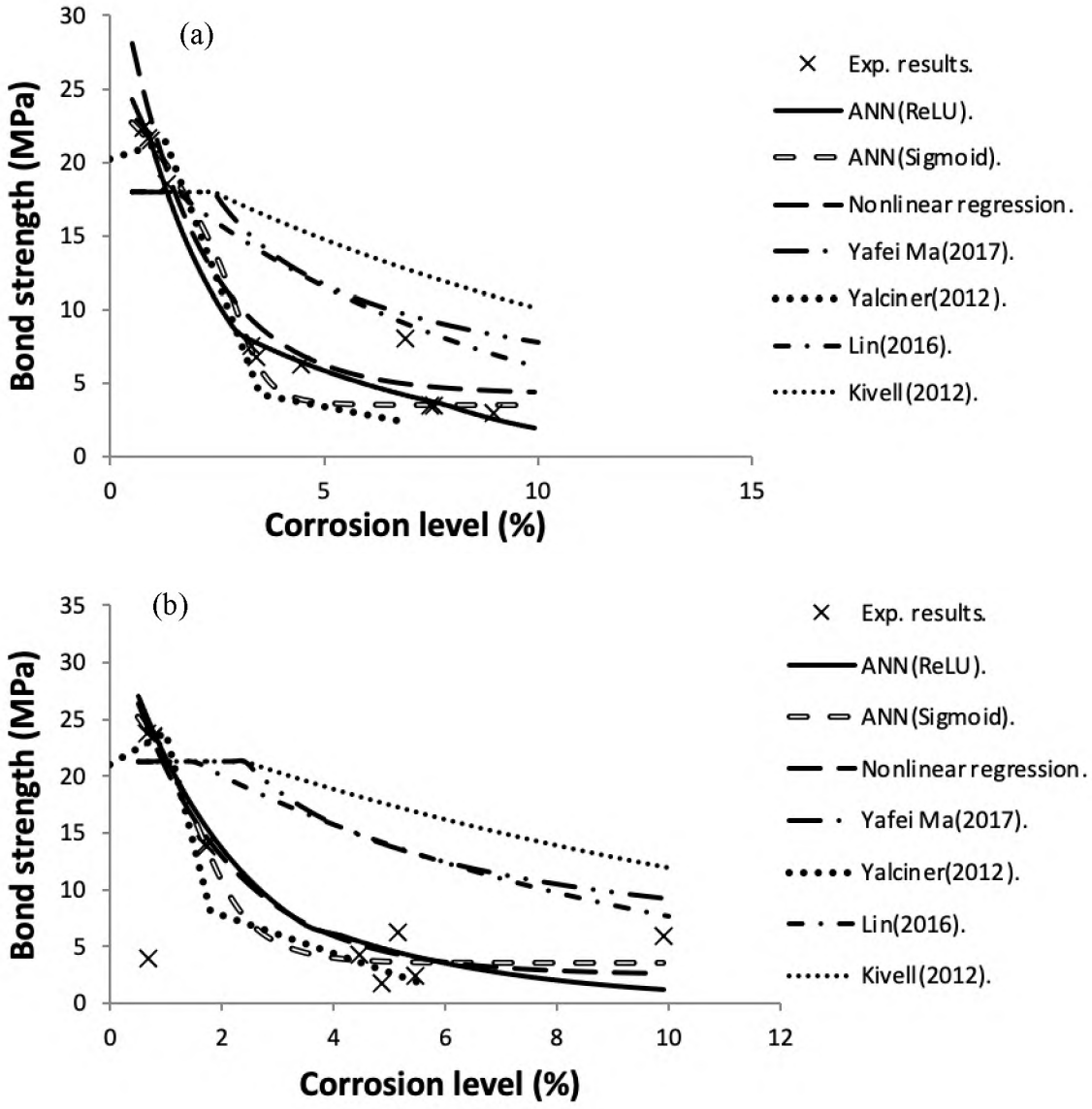


Fig. 17: Model-predicted versus experiment bond strength for the case of 23 MPa compressive strength for the case of (a) 15 mm cover, (b) 30 mm cover, (c) 45 mm cover.


(a)
Bond strength (MPa)
Corrosion level (%)
Exp. results.
ANN(ReLU).
ANN(Sigmoid).
Nonlinear regression.
Yafei Ma(2017).
Yalciner(2012).
Lin(2016).
Kivell(2012).
(b)
Bond strength (MPa)
Corrosion level (%)
Exp. results.
ANN(ReLU).
ANN(Sigmoid).
Nonlinear regression.
Yafei Ma(2017).
Yalciner(2012).
Lin(2016).
Kivell(2012).

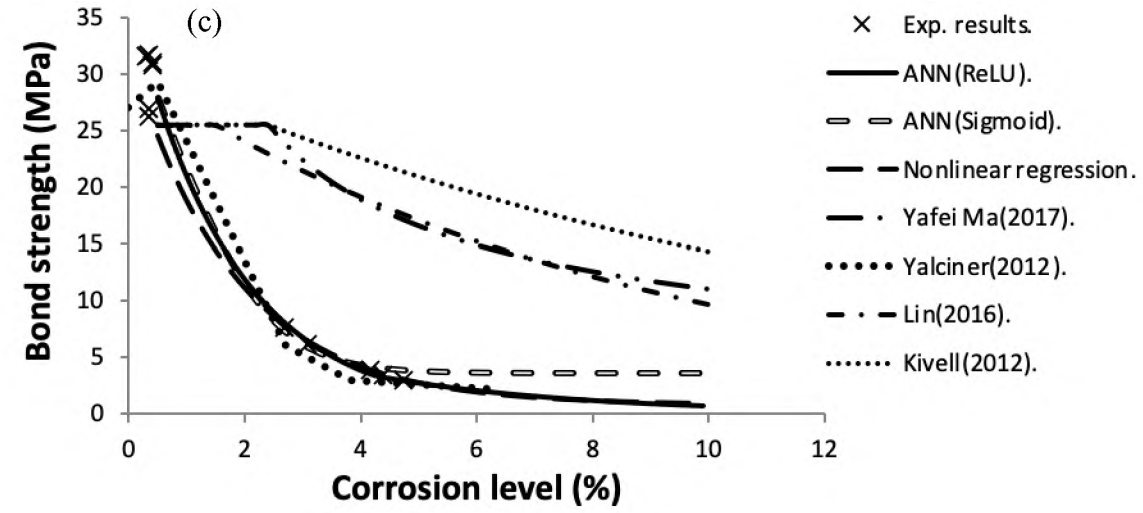


Fig. 18: Model-predicted versus experiment bond strength for the case of 51 MPa compressive strength for the case of (a) 15 mm cover, (b) 30 mm cover, (c) 45 mm cover.

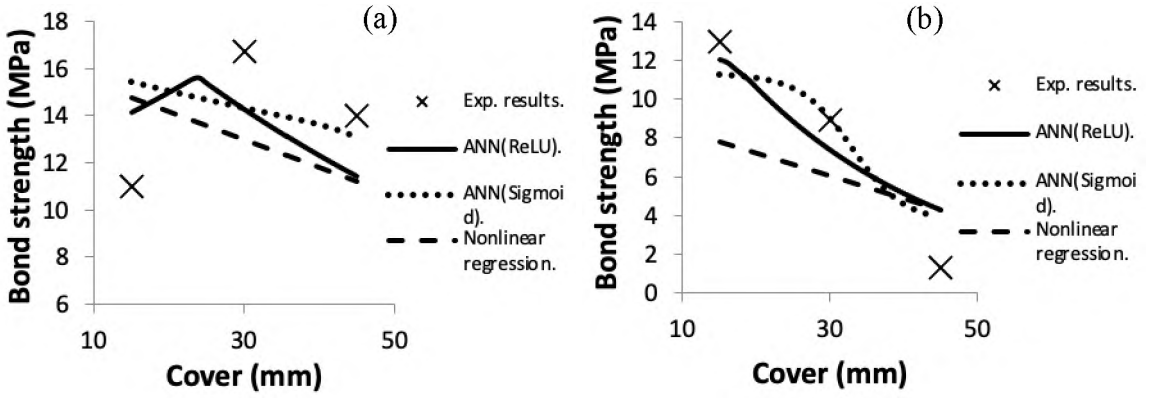


Fig. 19: Relationship between concrete cover and bond strength. (a) 23 MPa compressive strength and 2% corrosion level. (b) 23 MPa compressive strength and 4% corrosion level.

The theoretical bond strength as a function of the cover is shown in Figure 19a and b for the cases of 2% and 4% corrosion levels under the same 23MPa compressive strength, respectively. As seen in the earlier Figures 17 and 18, the Non-Linear regression model loses accuracy at the lower levels of corrosion state. There is a strong evidence in these cases that the ANN model with the ReLu activation function

outperforms the Non-Linear regression. The collective performance of the general ANN(ReLu) model against the experiment for the corroded samples is described in Figure 20. It shows that almost 93% of the predicted points are located within ±2.5 MPa margin of error. The coefficient of determination ($\mathbf{R}^2$) for the neural network is approximately 96%.

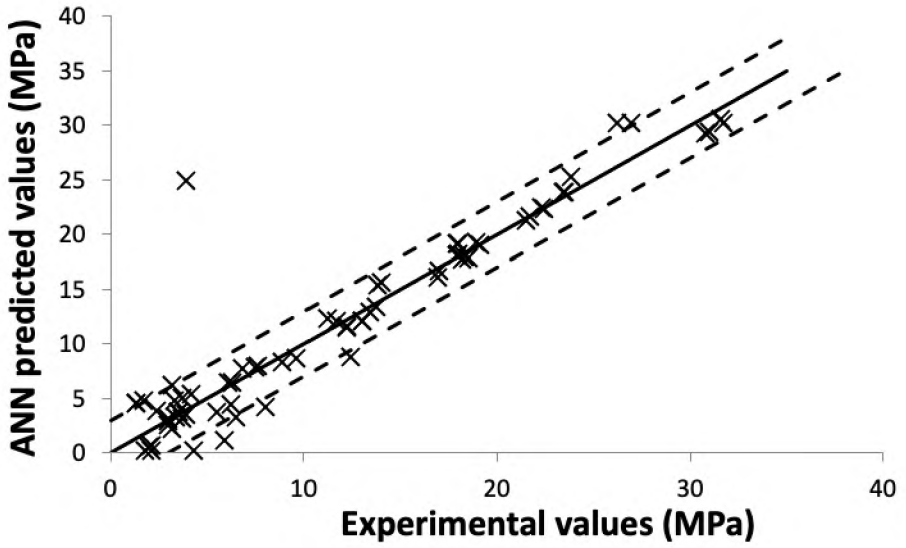


Fig. 20: Relationship between the experimental and the ANN(ReLu) predicted values.

For the non-corroded samples, Figure 21 compares the results from the model and the experiment to illustrate the effect of the cover on the bond strength at $f_c' = 23$ and 51 MPa. These predictions are based on the linear regression equation 10 only. Over the considered range, the bond strength is seen to increase with both the magnitude of cover and compressive strength of the concrete. For instance, at the same cover of 30 mm, a 57% increase in the bond strength measured between the cases of 23 and 51 MPa compressive strength. One observes that the overall trends of the predicted results are in agreement with the test data. This indicates the adequacy of the linear regression model

in characterizing the behavior of the bond stress for cases with a reduced number of independent variables.

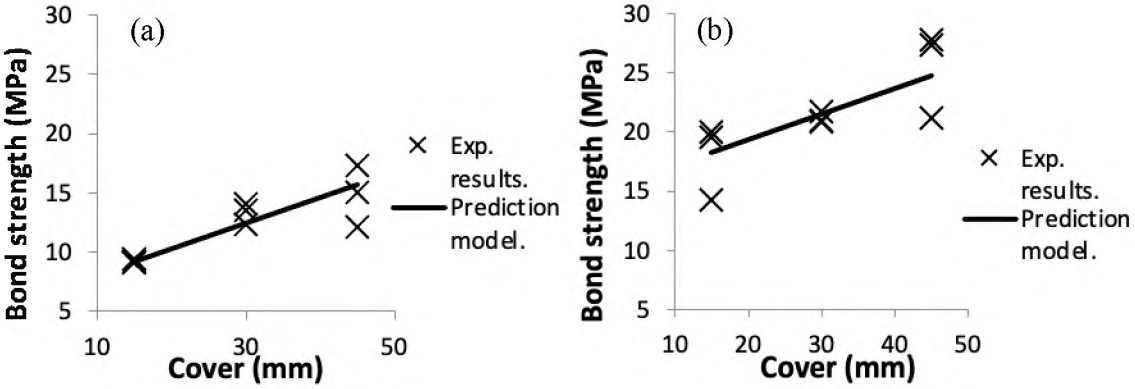


Fig. 21: Relationship between concrete cover and bond strength for non-corroded samples. (a) 23 MPa compressive strength. (b) 51 MPa compressive strength.

CHAPTER V

CONCLUSION

The ANN software utilizes a predetermined algorithm to create ANN that simulates the relationships between the input and target data. This algorithm dwindles the difference between the target data and the ANN outputs to the least possible value. However, in order to create the optimum ANN, many processes and decisions, which this algorithm cannot handle, must be made by the user. For instance, the most suitable activation function, which provides a more accurate model is contingent on the nature of the training data. In the present study, the ReLU activation function was associated with more accurate predictions than the Sigmoid function for the bond strength model, while the opposite is true for the SMA model.

Second, the percentage of the data used in the training process, which should include all the patterns that the ANN aims to capture, depends on how illustrative this percentage is. For example, for the bond strength model, reducing the training percentage below 90% would affect the ANN model negatively, while for the SMA model, only 37% of the data was used in the training process, and increasing this percentage does not improve the model.

The bond strength models created in chapter 4 reveal the superiority of ANN over conventional methods (linear and non-linear regression) utilized previously in literature by several researchers. In the non-linear regression model, the effect of each factor on the bond strength is assumed to be independent, which is the opposite of what the experiment suggests, so increasing the concrete cover would decrease the bond strength regardless of the corrosion level value. On the other hand, the ANN model takes the effect of each factor to be dependent so the cover effect at low corrosion level is completely different from its effect at high corrosion level.

The ANN was applied on two completely different areas (reinforced-concrete and SMA) and generated accurate models since the ANN bond strength model and SMA model have proved their superiority to nonlinear regression and other models created in literature. This confirms the capabilities of ANN as a powerful tool leading to relatively simple yet accurate models in various fields.

BIBLIOGRAPHY

Abosrra, L., Ashour, A.F. and Youseffi, M., 2011. Corrosion of steel reinforcement in concrete of different compressive strengths. Construction and Building Materials, 25(10), pp.3915-3925.

Albitar, M., Visintin, P., Ali, M.M., Lavigne, O. and Gamboa, E., 2017. Bond slip models for uncorroded and corroded steel reinforcement in class-F fly ash geopolymer concrete. Journal of Materials in Civil Engineering, 29(1), p.04016186.

Basheer, I.A. and Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods, 43(1), pp.3-31.

Chang, C.Y., Vokoun, D. and Hu, C.T., 2001. Two-way shape memory effect of NiTi alloy induced by constraint aging treatment at room temperature. Metallurgical and materials Transactions A, 32(7), pp.1629-1634.

Chung, L., Kim, J.H.J. and Yi, S.T., 2008. Bond strength prediction for reinforced concrete members with highly corroded reinforcing bars. Cement and concrete composites, 30(7), pp.603-611.

Cisse, C., Zaki, W. and Zineb, T.B., 2016. A review of constitutive models and modeling techniques for shape memory alloys. International Journal of Plasticity, 76, pp.244-284.

Concha, N.C. and Oreta, A.W.C., 2019. Bond strength prediction model of corroded reinforcement in concrete using neural network. International Journal, 16(54), pp.55-61.

Da Silva, I.N., Spatti, D.H., Flauzino, R.A., Liboni, L.H.B. and dos Reis Alves, S.F., 2017. Artificial neural network architectures and training processes. In Artificial neural networks (pp. 21-28). Springer, Cham.

Dancygier, A.N., Katz, A. and Wexler, U., 2010. Bond between deformed reinforcement and normal and high-strength concrete with and without fibers. Materials and Structures, 43(6), pp.839-856.

Darwin, D., 2005. Tension development length and lap splice design for reinforced concrete members. Progress in Structural Engineering and Materials, 7(4), pp.210-225.

Fang, C., Lundgren, K., Chen, L. and Zhu, C., 2004. Corrosion influence on bond in reinforced concrete. Cement and concrete research, 34(11), pp.2159-2167.

Feng, Q., Visintin, P. and Oehlers, D.J., 2016. Deterioration of bond–slip due to corrosion of steel reinforcement in reinforced concrete. Magazine of Concrete Research, 68(15), pp.768-781.

Ghaboussi, J., Garrett Jr, J.H. and Wu, X., 1991. Knowledge-based modeling of material behavior with neural networks. Journal of engineering mechanics, 117(1), pp.132-153.

Gonzalez-Fernandez, I., Iglesias-Otero, M.A., Esteki, M., Moldes, O.A., Mejuto, J.C. and Simal-Gandara, J., 2019. A critical review on the use of artificial neural networks in olive oil production, characterization and authentication. Critical reviews in food science and nutrition, 59(12), pp.1913-1926.

Gu, X., Zaki, W., Morin, C., Moumni, Z. and Zhang, W., 2015. Time integration and assessment of a model for shape memory alloys considering multiaxial nonproportional loading cases. International Journal of Solids and Structures, 54, pp.82-99.

Hong, S. and Park, S.K., 2012. Uniaxial bond stress-slip relationship of reinforcing bars in concrete. Advances in Materials Science and Engineering, 2012.

Huang, W.M., Ding, Z., Wang, C.C., Wei, J., Zhao, Y. and Purnawali, H., 2010. Shape memory materials. Materials today, 13(7-8), pp.54-61.

Jiang, C., Wu, Y.F. and Dai, M.J., 2018. Degradation of steel-to-concrete bond due to corrosion. Construction and Building Materials, 158, pp.1073-1080.

Juarez, C.A., Guevara, B., Fajardo, G. and Castro-Borges, P., 2011. Ultimate and nominal shear strength in reinforced concrete beams deteriorated by corrosion. Engineering Structures, 33(12), pp.3189-3196.

Kavzoglu, T., 1999, September. Determining optimum structure for artificial neural networks. In Proceedings of the 25th Annual Technical Conference and Exhibition of the Remote Sensing Society (pp. 675-682). Remote Sensing Society Nottingham, UK Cardiff, UK.

Kciuk, M., Chwastek, K., Kluszczyński, K. and Szczygłowski, J., 2016. A study on hysteresis behaviour of SMA linear actuators based on unipolar sigmoid and hyperbolic tangent functions. Sensors and Actuators A: Physical, 243, pp.52-58.

Khandelwal, A. and Buravalla, V., 2009. Models for shape memory alloy behavior: an overview of modeling approaches. The International Journal of Structural Changes in Solids, 1(1), pp.111-148.

Kirsch, S.M., Schmidt, M., Welsch, F., Michaelis, N., Schütze, A. and Seelecke, S., 2017. Development of a shape memory based air conditioning system. In 59th Ilmenau scientific colloquium (Vol. 59, pp. 11-15).

Kivell, A.R.L., 2012. Effects of bond deterioration due to corrosion on seismic performance of reinforced concrete structures.

Lachemi, M., Bae, S., Hossain, K.M.A. and Sahmaran, M., 2009. Steel–concrete bond strength of lightweight self-consolidating concrete. Materials and Structures, 42(7), pp.1015-1023.

Lagoudas, D.C., Entchev, P.B., Popov, P., Patoor, E., Brinson, L.C. and Gao, X., 2006. Shape memory alloys, Part II: Modeling of polycrystals. Mechanics of Materials, 38(5-6), pp.430-462.

Lee, H.S., Noguchi, T. and Tomosawa, F., 2002. Evaluation of the bond properties between concrete and reinforcement as a function of the degree of reinforcement corrosion. Cement and Concrete research, 32(8), pp.1313-1318.

Lin, H. and Zhao, Y., 2016. Effects of confinements on the bond strength between concrete and corroded steel bars. Construction and Building Materials, 118, pp.127-138.

Ma, Y., Guo, Z., Wang, L. and Zhang, J., 2017. Experimental investigation of corrosion effect on bond behavior between reinforcing bar and concrete. Construction and Building Materials, 152, pp.240-249.

Nemat-Nasser, S. and Guo, W.G., 2006. Superelastic and cyclic response of NiTi SMA at various strain rates and temperatures. Mechanics of materials, 38(5-6), pp.463-474.

Ortin, J., Planes, A. and Delaey, L., 2006. Hysteresis in shape-memory materials. The Science of Hysteresis, pp.467-553.

Otsuka, K. and Wayman, C.M. eds., 1999. Shape memory materials. Cambridge university press.

Owusu-Danquah, J.S. and Saleeb, A.F., 2017. On the cyclic stability of the thermomechanical behavior of NiTi shape memory cylindrical actuators. European Journal of Mechanics-A/Solids, 64, pp.143-159.

Padula II, S.A., Gaydosh, D.J., Noebe, R.D., Bigelow, G.S., Garg, A., Lagoudas, D., Karaman, I. and Atli, K.C., 2008, April. Influence of test procedures on the thermomechanical properties of a 55NiTi shape memory alloy. In Behavior and Mechanics of Multifunctional and Composite Materials 2008 (Vol. 6929, p. 692912). International Society for Optics and Photonics.

Padula, S., Qiu, S., Gaydosh, D., Noebe, R., Bigelow, G., Garg, A. and Vaidyanathan, R., 2012. Effect of upper-cycle temperature on the load-biased, strain-temperature response of NiTi. Metallurgical and Materials Transactions A, 43(12), pp.4610-4621.

Price, W.H., 1951, February. Factors influencing concrete strength. In Journal Proceedings (Vol. 47, No. 2, pp. 417-432).

Sajid, H.U. and Kiran, R., 2018. Influence of corrosion and surface roughness on wettability of ASTM A36 steels. Journal of Constructional Steel Research, 144, pp.310-326.

Saleeb, A.F., Padula Ii, S.A. and Kumar, A., 2011. A multi-axial, multimechanism based constitutive model for the comprehensive representation of the evolutionary response of SMAs under general thermomechanical loading conditions. International Journal of Plasticity, 27(5), pp.655-687.

Song, G., Ma, N. and Li, H.N., 2006. Applications of shape memory alloys in civil structures. Engineering structures, 28(9), pp.1266-1274.

Tang, C., Huang, W.M., Wang, C.C. and Purnawali, H., 2012. The triple-shape memory effect in NiTi shape memory alloys. Smart materials and structures, 21(8), p.085022.

Toloei, A., Stoilov, V. and Northwood, D., 2013, November. The relationship between surface roughness and corrosion. In ASME International Mechanical Engineering Congress and Exposition (Vol. 56192, p. V02BT02A054). American Society of Mechanical Engineers.

Tondolo, F., 2015. Bond behaviour with reinforcement corrosion. Construction and Building Materials, 93, pp.926-932.

Torre-Casanova, A., Jason, L., Davenne, L. and Pinelli, X., 2013. Confinement effects on the steel–concrete bond strength and pull-out failure. Engineering Fracture Mechanics, 97, pp.92-104.

Wada, K. and Liu, Y., 2008. On the two-way shape memory behavior in NiTi alloy—An experimental analysis. Acta Materialia, 56(13), pp.3266-3277.

Wu, Y.C. and Feng, J.W., 2018. Development and application of artificial neural network. Wireless Personal Communications, 102(2), pp.1645-1656.

Yalciner, H., Eren, O. and Sensoy, S., 2012. An experimental study on the bond strength between reinforcement bars and concrete as a function of concrete cover, strength and corrosion level. Cement and Concrete Research, 42(5), pp.643-655.

Zhao, Y., Lin, H., Wu, K. and Jin, W., 2013. Bond behaviour of normal/recycled concrete and corroded steel bars. Construction and building materials, 48, pp.348-359.