University of Southern Queensland

FACULTY OF ENGINEERING AND SURVEYING

DEVELOPMENT OF AN ARTIFICIAL NEURAL NETWORK (ANN) FOR PREDICTING TRIBOLOGICAL PROPERTIES OF KENAF FIBRE REINFORCED EPOXY COMPOSITES (KFRE).

A dissertation submitted by

Mr. Tyler John Griinke

in fulfilment of the requirements of

CourseENG8411 and 8412 Research Project

towards the degree of

Bachelor of Engineering (Mechanical)

October 2013

Abstract

Study in the field of tribology has developed over time within the mechanical engineering discipline and is an important aspect of material selection for new component design. Most of these components experience failure due to this form of loading. It has been well established that there are several conditions or parameters that may influence the tribological performance of a material. Good correlations with experimental results are not clearly obtained or achieved from mathematical models.

Artificial neural network (ANN) technology is recognised as an effective tool to accurately predict material tribological performance in relation to these influencing parameters. The benefit and importance is the ANN models capability to predict solutions by being trained with experimental data. They essentially catalogue the performance characteristics eliminating the need to refer to tables and the requirement for additional time consuming testing. This will aid in continuing research, development and implementation of fibre composites.

The aim of the project was to investigate artificial neural network (ANN) modelling for the accurate prediction of friction coefficient and surface temperature of a kenaf fibre reinforced epoxy composite for specific tribological loading conditions.

This study has verified the ability of an artificial neural network to make closely accurate generalised predictions within the given domain of the supplied training data. Improvements to the generalised predictability of the neural network was realised through the selection of an optimal network configuration and training method suited to the supplied training data set.

Hence, the trained network model can be utilised to catalogue the friction coefficient and surface temperature variables in relation to the sliding distance, speed and load parameters. This is limited to the domain of the training data. This will ultimately save time and money otherwise used in conducting further testing.

Faculty of Engineering and Surveying

ENG4111 Research Project Part 1 &

ENG4112 Research Project Part 2

Limitations of Use

Certification

I certify that the ideas, designs and experimental work, results analysis and conclusions set out in this dissertation are entirely my own efforts, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Tyler John Griinke

Student Number: 0061004639

Signature

Date

Acknowledgements

I extend thanks to my supervisor, Dr Belal Yousif for his guidance and support when it was needed throughout the project.

I would like to thank my parents, Pieter and Trish and my grandfather, Vern for their support and patience.

Table of Contents

List of Figure

Nomenclature

ANN –Artificial Neural Network

NN – Neural Network

R – Correlation Coefficient

MSE –Mean Square Error

SSE – Sum Square Error

KFRE – Kenaf Fibre Reinforce Epoxy

MSW – Mean Square Weights

1 Introduction

The outline and the research objectives of the project are established within this chapter. The main intention of the project is to investigate and develop an artificial neural network (ANN) that effectively predicts tribological characteristics of kenaf fibre reinforced epoxy composite (KFRE).

1.1 Project Topic

Development of an Artificial Neural Network (ANN) for predicting tribological properties of kenaf fibre reinforced epoxy composites.

1.2 Project Background

Societies increasing focus toward awareness for the environment has driven the development within the fibre composite industry. Sustainable, environmentally friendly, materials have subsequently emerged in popularity. There is also recent concern for the sustainability and limited nature of resources used in traditional petro-chemical based composites. These synthetic composites typically use synthetic fibres with petro-chemically based resins. It's also recognised that as resources are reduced there is a realistic concern for increased costs. The implementation of natural fibre composites is thus becoming increasingly favourable as sustainable replacements within industry.

The growing interest in implementing natural fibres for polymeric composite reinforcement is also driven by the recognition of their desirable properties. Such properties include their low density, non-abrasiveness, non-toxicity, biodegradability, renewability and low costs (Chin and Yousif 2009). Their higher specific properties like modulus, flexibility, strength, and impact resistance also make them attractive. Some fibre composites are successfully employed as component materials in various sectors. Many of these industrial components are placed under tribological loading.

Study in the field of tribology has developed over time within the mechanical engineering discipline and is an important aspect of material selection for new component design. Essentially the topic covers the science of wear, friction and lubrication (Yousif. B 2013). Most of these components experience failure due to this form of loading. It has been well established that there are several conditions or parameters that may influence the tribological performance of a material.

Artificial neural network (ANN) technology is recognised as an effective tool to accurately predict material tribological performance in relation to these influencing parameters (Nasir et al. 2009, Zhang et al. 2002, Rashed and Mahmoud 2009, Hayajneh et al. 2009). The recent increased application of ANN technology to model and characterise the tribological behaviour of the natural fibre materials is assisting in their further research, development and implementation.

1.3 Research Aim and Objectives

The project aim is to investigate artificial neural network (ANN) modelling for the accurate prediction of friction coefficient and surface temperature of a kenaf fibre reinforced epoxy composite for specific tribological loading conditions.

The primary project objectives are characterised below:

- Understand the process and benefit of developing neural networks used for prediction applications.
- Process sufficient previously collected tribology data and implement this data to establish an optimal ANN model through testing various neural, layer and function configurations.
- Train developed optimal ANN model and compare results with data to confirm accuracy of model. Consider implementing methods to improve network generalisation.
- Simulate ANN model and assess its ability to make predictions beyond trained domain.

1.4 Justification

More recently the recognition of the superior properties and demand for kenaf as polymer reinforcing fibres has become evident. There is history of this fibre being cultivated in areas like Malaysia, India, Thailand, Bangladesh, parts of Africa and south east Europe. Twine, paper, cloth and rope are some examples of where the fibre has been implemented. Many recent studies have established the superior properties that the kenaf fibre exhibit over other commonly used natural fibres such as jute, sugar cane, and oil palm (Chin &Yousif 2009). The kenaf fibres have also been shown to demonstrate strong interfacial adhesion between the fibres and the matrix (1 and 2).

The stronger interfacial adhesion has been recognised to promote improved wear performance (Chin &Yousif 2009). The usage of this natural fibre as polymeric composite reinforcements for tribology applications has had little conducted research. Subsequently, Chin and Yousif (2009) have conducted work assessing the potential of kenaf fibres for reinforcement in polymer based tribo-composites. In their work they have assessed various related tribological conditions and parameters. These include sliding distance, applied loads, sliding velocity and fibre orientation with respect to sliding direction. The tribological characteristics assessed where the coefficient of friction, contact surface temperature and the specific wear rate.

There are many operating parameters and contact conditions that can have a strong influence on the tribological properties of a polymeric composite (Yousif and El-Tayeb 2007). Thus establishing models that characterises and predicts the performance based on the tribological conditions are useful tools. They essentially catalogue the performance characteristics eliminating the need to refer to tables and the requirement for additional time consuming testing. This will aid in continuing research, development and implementation of the fibre composite.

Good correlations with experimental results are not clearly obtained or achieved from mathematical models. It is also recognised to be a time consuming process to develop a pure mathematical model to estimate these properties (Nasir et al. 2009). Artificial Neural Network (ANN) modelling is more readily implemented as a successful alternative tool to closely estimate tribological properties (Zhang et al. 2002, Jiang et al. 2007). Today many complex engineering and scientific problems are being solved by utilising this ANN technology. The benefit and importance is the ANN models capability to predict solutions by being trained with experimental data.

1.5 Scope

To develop an optimal ANN model, data from the previous works will be processed to be implemented in the training of the network. . Initial trial and error training will be conducted for various network setups. The process of establishing the optimal ANN setup will involve a simple series of attempts with various layer, neural and function configurations. By comparing the performance of these various setups or developed sample models, an optimal ANN model will be derived.

The models will be developed and trained within the ANN toolbox. The optimal layer configurations, available transfer functions and training functions will be assessed by comparing the performances of the sum squared error (SSE). The model setup developed based on this selection process will undergo further training to try and achieve higher accuracy and finally produce an ANN model based on the training data set.

1.6 Conclusion

The project strives toward investigating and developing an optimal ANN tool that accurately predicts some tribological performance characteristics of a KFRE composite through comparing a series of attempted configurations. A review of sufficient and relevant literature will be conducted to establish an understanding for methods implemented to develop an optimal neural network. A basis of limitations and expected outcomes for the project may be established from this.

2 Literature Review

2.1 Introduction

Within the following chapter current literature and previous research studies will be reviewed. Most of the information in the various subject matters looked at are obtained from mostly published information sources and from communication with supervisors. A general background of neural networks pertaining to their operation, development and applications will be presented. Additional background will be provided regarding the tribology, its importance and current testing procedures used to generate relevant characterising data. A review on natural fibre composites and their growing position in engineering along will be given. An assessment will also be conducted on the consequential effects of the project.

2.2 Neural Networks (NNs)

The use of artificial neural networks (ANNs) has grown exponentially in recent decades. Its current applications encompass a vast range of subjects as diverse as image processing; signal processing, robotics, optics, manufacturing systems medical engineering, and credit scores (*Lisboa 1992*). In 1943 McCulloch and Pitts, using simple threshold logic elements, represented individual neuron activity and showed how many units interconnected could perform logic operations. This was based on the realisation that the brain performed information processing in a particular way. The understanding of biological neurons is that their basic activity involves the transmission of information via electrical impulses propagating along the axon to activate the synapses (Refer to Figure 1)(*Lisboa 1992*, *Fauset 1994*). This excitation at the synapses junction travels to the next neurone by its connected dendrites. The hillock zone is recognised as the region of the neurons which dictates their firing rate. (Lisboa 1992).

2.2.1 Biological Neurons

Brain Function

ANNs draw much of their inspiration from the biological nervous system. Therefore some knowledge of the way this system is organised is very useful. A controlling unit which is able to learn is required by most living creatures, providing them with the ability to adapt to changes within their environment. To perform such tasks complex networks of highly specialized neurons are used by higher developed animals and humans. The brain is the control unit that is connected by nerves to the sensors and actors in the whole body (*www.teco.edu*). It is divided into different anatomic and functional sub-units, each having specific tasks like hearing, vision, motor control and sensor control. The brains complexity can be contributed to the considerably large number of neurons, approximately 10^{11} on average that it consists of (*www.teco.edu*). These are recognised as the building blocks of the central nervous system (CNS). The CNS has around 10^{10} neurons conducting the neural signalling elements (*Groff and Neelakanta 1994*).

Biological Neuron structure

There is enormous complexity to the structure and processes within simple neuron cell. Most sophisticated neuron models in artificial neural networks seem toy-like. The neurons are interconnected at points called synapses. Structurally the neuron can be divided in three major parts: the cell body (soma), the dendrites, and the axon (*Fausett 1994*). These features of the neuron are indicated in Figure 1.

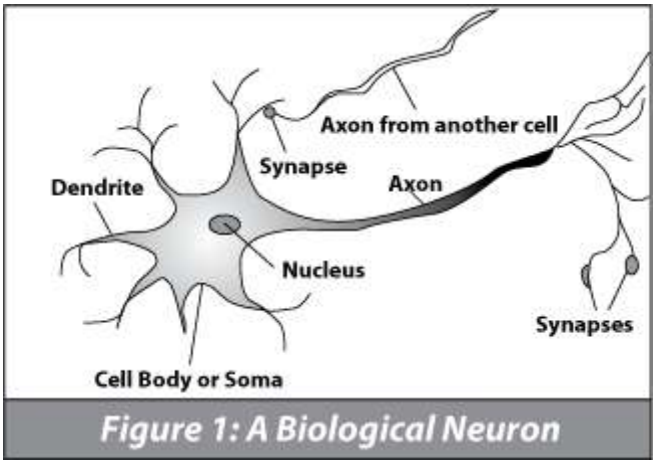

Figure 1- Biological Neuron (www.neuralpower.com)

Lisboa (*1992*), Fausett (*1994*), along with Groff and Neelakanta (*1994*) all recognise neurons as the building blocks of signalling unit in the nervous system. Excitability, development of an action potential and synaptic linkage are considered as general characteristics of all nerve cells (*Groff and Neelakanta 1994*). These are key neural properties mathematical models of neurons base their construction on.

The dendrites make connections to a larger number of cells within the cluster. They are referred to as a hair liked branched fibres emanating from the top of the cell (*Groff and Neelakanta 1994*). Most input signals enter the cell via the dendrites (*www.teco.edu*). Input connections are made from the axons of other cells to the dendrites or directly to the body of the cell. Each neuron consists of a single axon, a fine long fibre leading from the neuron body and eventually arborizing into strands and sub strands as nerve fibres. From 1-100m/s, it transports the output signal of the cell as electrical impulses (action potential) along its length to its terminal branches (*Lisboa 1992, Groff and Neelakanta 1994*). Synapse refers to the connection of a neurons axon nerve fibre to the soma (cell) or dendrite of another neuron (*Lisboa 1992*). Justifying the complexity of a biological neuron, there is typically between 1000 to 10000 synapses present on each neuron (*www.neuralpower.com, Groff and Neelakanta 1994*).

Biological Neuron Operation

Dendrites work as input receptors for the incoming signals from other neurons by channelling the postsynaptic potentials to the neurons soma, which performs as an accumulator /amplifier. The neuron's output channel is provided by the axon as it conveys the neural cell's action potential (along nerve fibres) to synaptic connections with other neurons (*www.teco.edu, Groff and Neelakanta 1994*). This transfer of impulse and neuron connection is illustrated by Figure 2.

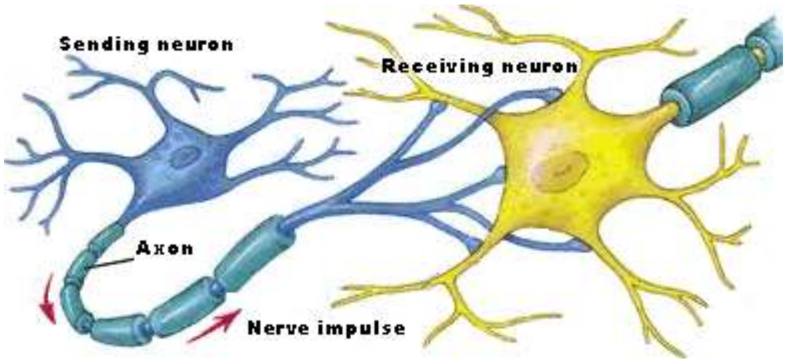

Figure 2– Connection and impulse transfer of two biological neurons (www.optimaltrader.net).

Electrical signals that encode information by the duration and frequency of their transmission are action potentials. The transmission of the action potential down the axon involves a large movement of ions cross the axon's membrane (*Groff and Neelakanta 1994, Barnes 2012*). As a collective process across the neuronal assembly, neural transmission is physically a bio chemical activated flow of electric signals (*Barnes 2012*).

A flow of chemicals across the synaptic junctions, from the axons leading from other neurons, cause the activation of the receiving neuron. The electrical synapse effects will either be excitatory or inhibitory. This is based on whether the hillock potential is raised or lowered by the postsynaptic potentials, enhancing or reducing likeliness of triggering an impulse, respectively. (*Lisboa 1992, Groff and Neelakanta 1994*). The neuron fires by the propagation of an action potential down the output axon if all the gathered synaptic potentials exceed a threshold value in a short period of time. This time period is referred to as the period of latent summation. A cell cannot refire for a short period of several milliseconds, known as the refractory period (*Barnes 2012*). Neural activation is a chain like process, where a neuron that activates other neurons was itself activated by other activated neurons.

There are many different types of neuron cells found in the nervous system. The differences are due to their location and function. The neurons perform the summation of the inputs, which may vary by the strength of the connection or the frequency of the incoming signal (*www.teco.edu*). The input sum must exceed certain signal strength or activation threshold for an impulse to be sent past the hillock zone and along the axon. The hillock zone is recognised as the region of the neurons which dictates their firing rate (*Lisboa 1992*).

2.2.2 Artificial Neural Networks (ANN)

Most describe ANN as a biologically inspired mathematical model used to solve complex scientific and engineering problems. Artificial neurones implement weightings or multiplication factors to simulate synaptic junction strength of biological neurones. Summations of signals received from every link models the action of the hillock zone (*Lisboa, 1992, www.teco.edu*). Numerous literatures on ANNs have been presented in recent years. Gyurova and Friedrich (*2010*) described the neural networks as being similar to the brain, containing a massive parallel collection of small and simple processing units. Models typically compose of numerous non-linear computational elements that operate in parallel, organised into reminiscent patterns of biological neural nets (*Lippmann 1987*). Figure 3 illustrates this concept with a typical structure of an ANN setup.

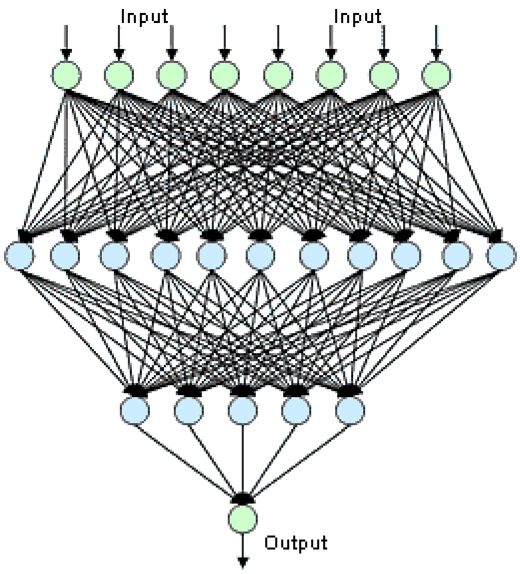

Figure 3– ANN structure representing interconnected organised parallel operating nature of numerous individual neurons (www.optimaltrader.net).

It is also identified that an ANN acts like "black-box" as the modelling process is relatively not clear and any physical relationships within the data set are difficult to obtain from the Network (*Gyurova. & Friedrich 2010*). Figure 4 simply depicts this perception of the ANN. Lippman (*1987*) suggests the non-linear nature enables NNs to perform signal filter operations and functional approximations which are beyond optimal linear techniques. Thus they are capable of performing pattern recognition/classification by defining non-linear regions in feature space. The NNs are also recognised to perform at higher computational rates than Voneuman single processor computers due to the parallel nature of the networks (*Fausett 1994*).

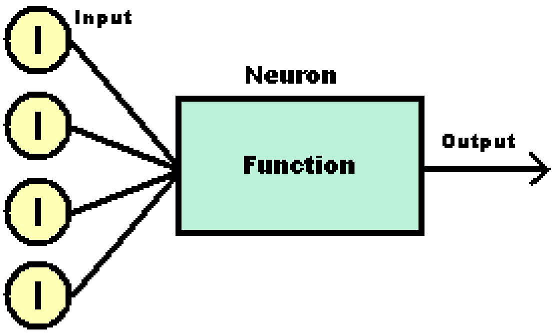

Figure 4 – ANN summarised as black-box that computes outputs from various input parameters. (www.optimaltrader.net).

The ANN learns and models itself on experience by detecting trends or patterns within the data it is presented with (*Gyurova & Friedrich 2010*). This is achieved by the computational elements or nodes being connected by weights and bias factors. These are adapted during use and training of the network to improve performance. This adaptive nature enables the NN to learn characteristics of the input signals and to adjust to changes in data (*Lippmann R. 1987*).

Subsequently, no defining physical relationships and observational theory is necessary in the ANNs construction. This aspect clearly has an advantage over regression analysis and is therefore accommodates problem modelling where input and output relationships are unclear or significant formulation time is required. Gyurova & Friedrich (*2010*), Tchaban et al. (*1998*), Velten K et al. (*2000*), Myshkin et al. (*1997*) and Schooling et al. (*1999*) all recognise and validate the previously defined aspect. Buttsworth et al. (*2009*) and Yusaf et al. (*2009*) also recognised implementing ANNs as an investigative tool, to model and predict data, greatly reduces the amount of expensive and time consuming testing required.

In the engineering tribological field the several applications such as wear, erosion, friction, temperature sensitivity and surface roughness have employed the ANN method of prediction. All works carried out implementing the ANNs report that their models were capable of output predictions to variable accuracy levels. This is depicted by the graph (Figures 5 & 6) presented by Yusaf et al. (*2009*), showing the predicted vs. experimental values of a derived ANN model for motor performance parameters.

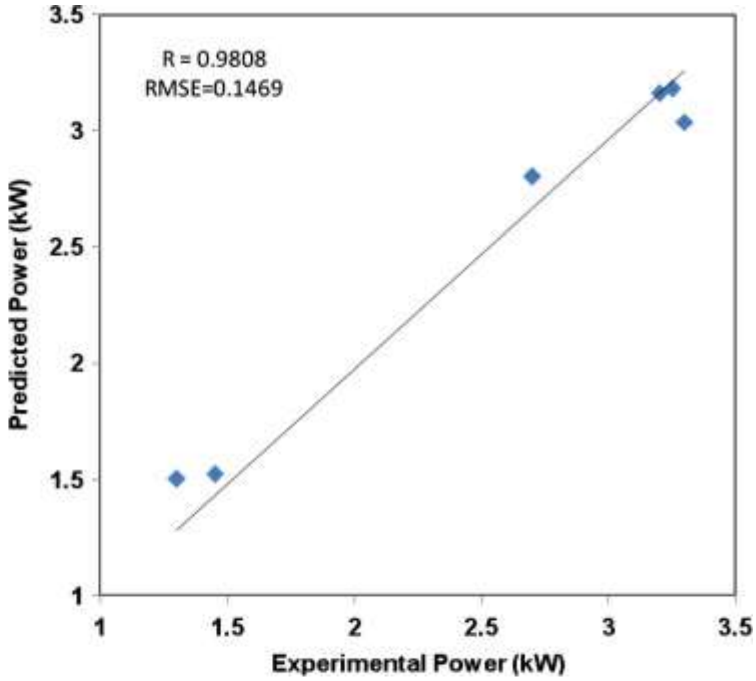

Figure 5 - The predicted vs. experimental values for experimental motor performance parameters (Yusaf et al. 2009).

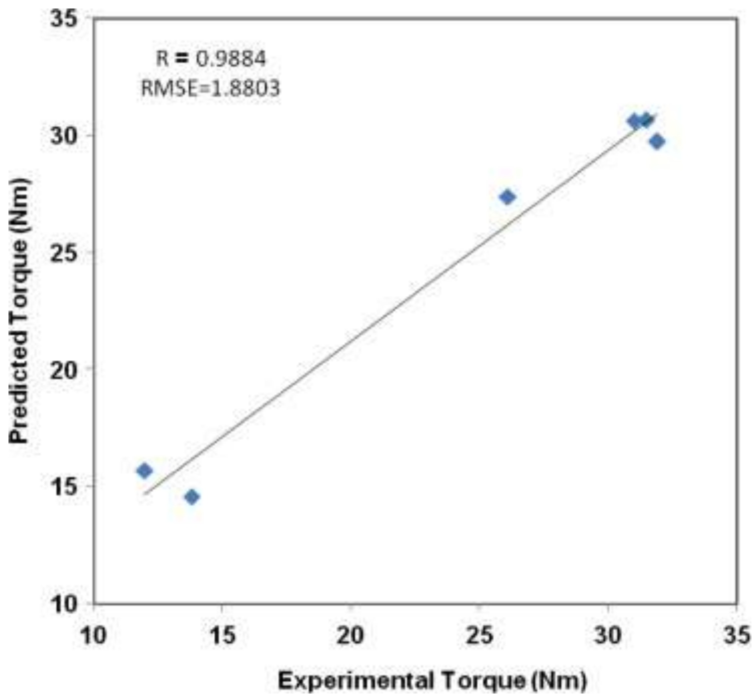

Figure 6 - The predicted vs. experimental values for experimental motor performance parameters (*Yusaf et al. 2009*).

These levels of accuracy or performance are recognised to be controlled by a few elements (*Nasir et al. 2009*). The NN structure, input data, and the training functions have been recognised as influential factors by most recent literature (Zhang *et al. 2002, Jiang et al. 2007, Pai et al. 2008, Aleksendric & Duboka 2006, Jie et al. 2007*). Lippmann (*1987*) also identifies NN model performance with respect to a dataset is specified by the node characteristics, network topology, and training or learning rules. Subsequently, both network design rules and training rules are the topic of much current research.

2.2.3 Node/Neuron Operational Structure

McCulloch and Pitts developed the first mathematical (logic) neuron model. The sum unit multiplies each input x_i^{in} by a weight W before summing them. If a predetermined threshold is exceeded by the sum, the output will be one or else it will be zero. Thus in this models case the neuron is either excited or inhibited by its inputs giving an output when its threshold is exceed. This neuron model is considered a binary device since it exists as either as active or inactive. This is presented in the arithmetic notation of 1 and 0, respectively (*Groff and Neelakanta 1994*).

The first ANN containing single layer artificial neurons connected by weights to a set of inputs were first seen around the in 1950's and 1960's. Rosenblatt conceived that this simplified model of the biological mechanisms of processing of sensory info refers to perceptron (*Groff and Neelakanta 1994*).

Nodes or neuron setup as computing elements is characterised by the summation of inputs multiplied weight and/or bias multiplication factors and passed through a specific transfer function to produce a node output. The function and operation of the neuron is perceived the same by Haykin (*1999*), Fausett (*1994*), Zeng (*1998*) and many other literatures. Essentially they all believe an artificial neuron may be regarded as a simple calculator.

Mathematical Expression

Hillock zone is in essence modelled by the summation of the signals received from every link. The neuron's firing rate in response to this summative incoming signal is then portrayed by a mathematical function. The resulting value represents the frequency of emission of electrical impulses along the axon (*Lisboa 1992*). These are essential in the behaviour of the neural networks. Thus making an exact mathematical treatment difficult, yet essential if artificial networks are to do anything useful.

Neuronal network connections are mathematically presented as a basis function U(**W, x**) where **W** is the Weight matrix and **p** is the input matrix. U is a linear basis function in hyper-plane, given by:

$$U_i(W,p) = \sum_{j=1}^{R} W_{ij} p_j \qquad 1$$

The net value expressed by the basis function is generally added to a bias factor. This is then transformed by a nonlinear function or activation function to portray the nonlinear activity of the neuron (*Groff and Neelakanta 1994*). Figure 7 illustrates this with an elementary neuron model with R inputs.

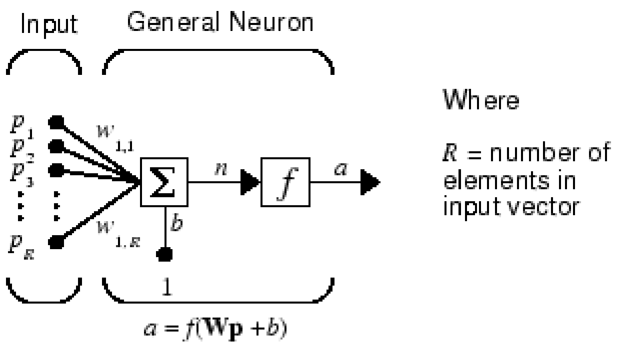

Figure 7 - Elementary Neuron Model (*Demuth and Beale 2013*)

Each input p in to the neuron is multiplied by appropriately assigned weights *w*, which characterise the fitting parameters of the model. The weighted inputs are summed as defined by the linear basis function (equation1). The sum is added to a bias factor to form the input to the transfer function *f*. Neurons may implement any differentiable transfer function *f* to generate their output (*Demuth and Beale 2013, www.teco.edu*). This may be summarised in the associated formula presented as:

$$a = f(Wp + b) \qquad 2$$

This presented mathematical treatment of neuron calculative process is the general consensus of most of the related literature viewed.

Weights

Groff and Neelakanta (*1994*) perceived the mathematical degree of influence that a single neuron has on another is accomplished by a weight associated with their interconnection. The synapses are in essence the biological counterpart of this interconnection. Lisboa (*1992*) identifies that mathematically the strength of each synaptic junction is represented by a multiplication factor or weight. A positive weight is used for excitatory responses and negative weights for an inhibitory effect. When the NN learns something in response to new input the, weights are modified. Hence, training the network involves alteration of the weights in order to more accurately fit the models parameters.

Bias

As previously indicated a bias can be included by adding a component to the input vector **p** or to the sum of the dot product of the weight and input vectors (**Wp)**.The bias is therefore treated exactly like any other weight. It performs like a connection weight from a unit whose activation is always 1 (*Fausett 1994*). The term determines the spontaneous activity of a neuron, i.e. in absence of any incoming signal. This can also be viewed as setting the threshold values for the sudden onset of a high firing rate, thus the term non linear threshold element (*Lisboa 1992*). Some authors implement a fixed threshold for the activation function instead. However, this is demonstrated be essentially be equivalent to using an adjustable bias (*Fausett 1994*).

Transfer Function

As previously established the summation of the weighted input products must be put through an activation function to ensure that the neuron output doesn’t exceed its minimum or maximum activation value. Lsiboa (*1992*) identifies that real neurons have a limited dynamic range from nil response to the full firing rate. Subsequently, the function is typically non-linear, levelling off at 0 and1. The common and most useful activation functions are step, ramp, sigmoid, and gaussian functions (*Groff and Neelakanta 1994*).

The output is typically transferred forward to the neurons in the next connected neural layer. This perception of an artificial neuron recognises that it is a non linear function of its inputs (*Lisboa 1992*). The function is commonly a sigmoid function that compresses the combined neuron input to the required range of the activation value, between 0 and 1 (*Lippmann 1987*).

Most multilayer networks often implement the log-sigmoid transfer function. As the a neurons net input goes from negative to positive infinity the log-sig function generates outputs between 0 and 1. This function is illustrated by Figure 8.

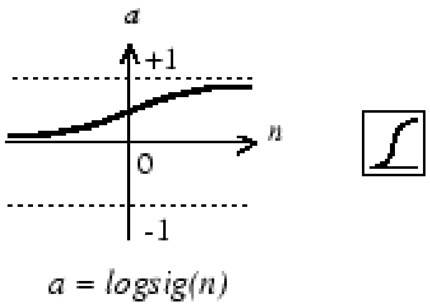

Figure 8 – Log-Sigmoid Transfer Function (*Demuth and Beale 2013*)

The tan-sigmoid function is considered as a common alternative in multilayer networks. This function generates outputs between -1 and 1, as the neurons net input goes from negative to positive infinity. The function is illustrated in Figure 9.

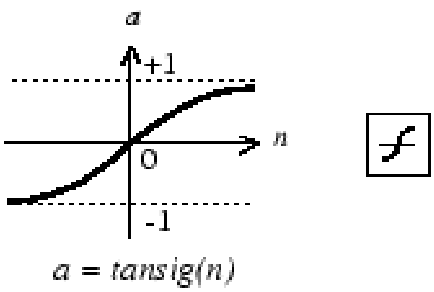

Figure 9 – Tan-Sigmoid Transfer Function (*Demuth and Beale 2013*)

The neurons that implement the sigmoid output functions are often used for pattern recognition problems. Linear output neurons are used for function fitting problems. The pure linear transfer function is depicted in Figure 10.

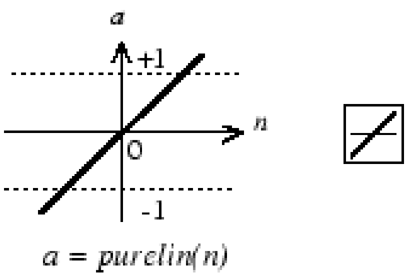

Figure 10 – Linear Transfer Function (*Demuth and Beale 2013*)

The three transfer functions presented are the most commonly employed in multilayer networks. There various other differentiable transfer functions like the step, ramp and Gaussian that may be implemented (*Groff and Neelakanta 1994, Demuth and Beale 2013*).

2.2.4 Layout

As previously discussed it has been established by numerous works that the accuracy of the NN capability of predicting data is dependent on the network structure or layout. The structure is ultimately defined by the setup of the nodes or neurons and the network topology.

Network Structure (Topology)

The structure of an ANN involves the organisation of network neurons into layers. The three primary layer types are the input layer, hidden layer/s and the output layer (*Gyurova L. & Friedrich K. 2010*). This is the general consensus of mostly all viewed literature regarding NN structuring. The input layer is the initial layer where the data is presented into the network while the output layer is the final layer dictating the outcome of the system (*Demuth and Beale 2013*). The layer in between is the referred to as the hidden layer/s which represents the calculative brain (*Nasir et al. 2009*). Signals from the input layer are spread through the hidden layer/s where the neurons and the inter connections manipulate the input data at each layer then finally sum to produce an output (*Lisboa 1992, Nasir et al. 2009*).

The number of neurons in the input and output layers typically reflect the number of input and output variables. More than one layer may make up the hidden layer and the volume of neurons in each layer is flexible. Nasir et al. (*2009*) identifies that the complexity of the system will influence the number of hidden layers and their associated neuron volume required to ascertain higher levels of performance. The systems complexity is in respect to the number of input parameter, irregularities and fluctuations in the data. Therefore layer configuration involving the number of layers and the number of neurons within each layer is dependent on the nature of the input data. This has been validated by various previous works conducted in the related field of tribology.

In the work of Zhang et al. (*2002*) the ANN generated to predict tribological properties of short fibre composites consisted of 9 input parameters and required 3 hidden layers. The ANN developed by Nasier et al. (*2009*) to predict tribological properties of polymeric composites performed best with a single hidden layer for its 4 input parameters. A single hidden layer was also required in the ANNs with two input parameters for the works of Jie et al. (*2007*) and Cetinel et al. (*2006*). These works related to the study of tribological behaviour for 30 wt.% carbon-fibre-reinforced polyetherketone composite (PEEK-CF30) and Mo coating wear loss, respectively. The work conducted by Aleksendric and Duboka (*2006*) in using ANNs to predict automotive friction material characteristic established that the use of larger databases provided a greater degree of accuracy.

Feed Forward Network

Feed forward networks typically consist of one or more hidden layers of sigmoid neurons, followed by an output layer of linear neurons. A detailed model of single-layer network containing *S* neurons with *R* inputs and log-sigmoid transfer functions is presented on the left in Figure 11. A layer diagram of the neurons is also presented on the right.

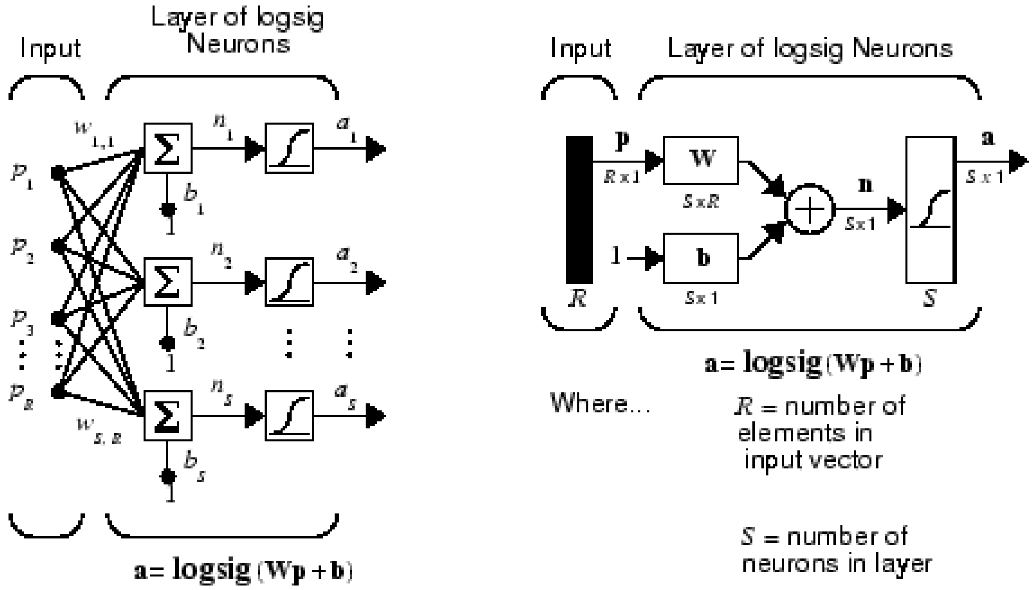

Figure 11 – General Feed forward network (*Demuth and Beale 2013*)

Nonlinear relationships between input and output vectors are able to be learned by multiple neuron layers that implement nonlinear transfer functions. Function fitting problems often use a linear output layer. If however the network outputs are desired to be constrained, a sigmoid transfer function should be employed. An example of this would relate to pattern recognition problems, where the network is required to make decisions (*Demuth and Beale 2013).*

Figure 12 that follows is a two-layer tan-sigmoid/pure-linear network. It may generally be implemented to approximate functions. Given sufficient hidden layer neurons, can approximate any function with a finite number of discontinuities subjectively well

(*Demuth and Beale 2013*). As gathered from the diagram the subscript on the weight matrix is determined by the associated layer number.

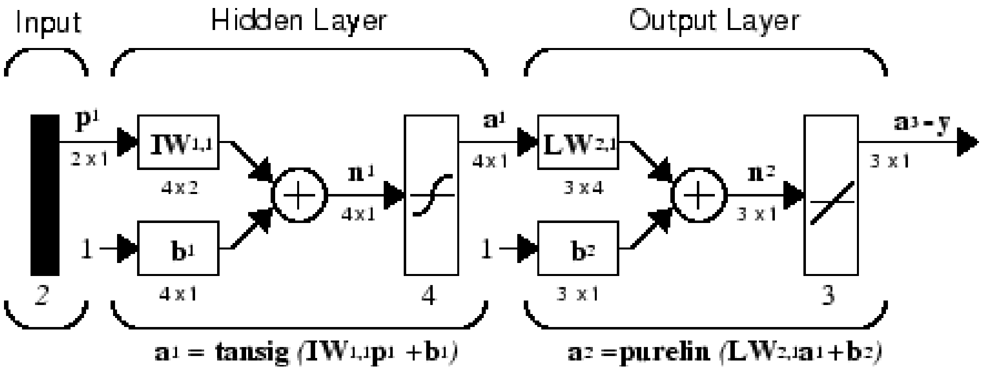

Figure 12 - Two-layer tan-sigmoid/pure-linear network (*Demuth and Beale 2013*)

2.2.4 Training and Training Functions

A response pattern or a distribution of memory within interconnecting neurons is clearly evident by the spatial propagation of their linked sequential responses. Relevant writing and reading phase exist for this memory phase unit. Writing refers to the storage of the set of info data to be remembered, whilst the reading phase is involves the retrieval of this data. The storage of the data specifies the gained training and learning experiences of the network (*Lisboa 1992*). A dilemma with developing an ANN is establishing weight or coefficient values that best fit the network and the known experimental data. Adaption or learning is a major focal point for NN research.

To characterise the connection strength the neural network adaptively updates the synaptic weights. This process follows a set of informational training rules (*Lisboa 1992*). Most NN algorithms adapt connection weights in time to improve performance based on current results. The learning rules specify an initial set of weights and indicate how weights should be adapted during use to improve performance (*Lippmann 1987*). Typically, the actual output values are compared to the teacher values and if a difference exists it is minimised on a basis of least-squares error. This is therefore achieved by optimising the synaptic weights by reducing the associated energy function (*Lisboa 1992*).

Mean Squared Error (MSE)

The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance. The common performance function is mean square error. The average squared error between the networks outputs a and the target outputs t (*Demuth and Beale 2013,Nirmal 2010*). It is defined as follows:

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \qquad 2$$

Any standard numerical optimization algorithm can be used to optimize the performance function. There are a few key standouts that have demonstrated excellent ANN training performance. These optimization methods commonly use the gradient or the Jacobian of the network errors with respect to the network weights. The gradients are calculated using a technique called the *back-propagation* algorithm, which involves performing computations backward through the network (*Demuth and Beale 2013, Nirmal 2010*).

Supervised and Unsupervised

Unsupervised and supervised learning are the two primary learning techniques. The unsupervised strategy the network is trained via a training set containing input training patterns only. Without teacher aid the network Adapts itself upon the experiences collected through the previous training set. The method is also referred to as Hebbian learning, where neuron units I and J are simultaneously excited and their connection strength is increased in proportion to their activation product. Many pairs of input and output training patterns within the training data are required for supervised learning. Fixed weight networks are those that have pre-stored synaptic weights and don’t implement training (*Lisboa 1992, Fausett 1994*). A single layer of input and a single layer of output neurons exist within such networks.

Training Algorithms

There are various types of available training algorithms. The gradient descent optimisation algorithm is considered the simplest and is used to demonstrate the general training operation. The network weights and biases are updated in a way that promotes the most rapid decrease in the performance function, the greatest negative gradient (Nasir etal. 2009, *Nirmal 2010*). An iteration of this algorithm may be expressed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k \tag{3}$$

where $\boldsymbol{x}_k$ is a vector of current weights and biases, $\boldsymbol{g}_k$ is the current gradient, and α_k is the learning rate. Iteration of this equation is continued until the networks performance function converges (*Demuth and Beale 2013, Nirmal 2010*). In essence the gradient $\boldsymbol{g}_k$ approaches zero. Often the "backpropagation" term refers specifically to this gradient descent algorithm. However, the process of computing the gradient and Jacobian by performing calculations backward through the network is applied in all of the training functions listed above. Therefore, specifying the optimisation algorithm used rather than just back propagation alone is recommended for clarity.

Back propagation Training Algorithm

The back-propagation computation is derived using the chain rule of calculus. The training involves repetitive steps of evaluating and optimizing the weights until the performance ceases improving. Lippmann (*1987*) defines it as an iterative gradient algorithm developed to reduce the MSE between the actual output and the desired output of a multilayer feed-forward perceptron, requiring continuous differentiable non-linearities.

The following is a step by step algorithm of the back propagation training phase presented by Fausett (*1994*):

1. Initialise weights (set to small random values)
   Complete following steps for each training pair while stopping condition is false.

2. *Feedforward*:
   Involves input units (X_i, $i=1, \ldots, n$) broadcasting it signal to all units in the above layer (hidden units). Each hidden unit (Z_j, $j=1, \ldots, p$) sums the weighted input signals and applies its activation function to compute an output signal, which is sent to all the output units in the above layer. The output units (Y_k, $k=1, \ldots, m$) also sum the weighted input signals and applies it activation function to produce its output signal.

3. *Backpropagation of error:*
   Each output unit (Y_k, $k=1, \ldots, m$) receives a target pattern corresponding to the input training pattern, computes its error information term,
   $\delta_k = (t_k - y_k)f'(y_{in_k})$,
   Calculates the weight correction term (to later update w_{jk}),
   $\Delta w_{jk} = \alpha \delta_k z_j$,
   Calculates its bias correction term (to later update w_{0k}),
   $\Delta w_{0k} = \alpha \delta_k$,
   And sends δ_k to units in the layer below.

4. Each hidden unit (Z_j, $j=1, \ldots, p$) sums its delta inputs (from units in layer above),
   $\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk}$,
   Multiplies by derivative of its activation function to calculate its error information term,
   $\delta_j = \delta\_in_j f'(z\_in_j)$,
   respectively calculates its weight and bias correction terms (to update them later),
   $\Delta v_{ij} = \alpha \delta_j x_i$,
   $\Delta v_{0j} = \alpha \delta_j$,
   The δ's are repeatedly calculated for each additional layer.

5. *Update weights and biases:*
   Each output unit (Y_k, $k=1, \ldots, m$) updates its weights and bias ($j=0, \ldots, p$):
   $w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$.
   Each hidden unit (Z_j, $j=1, \ldots, p$) updates its weights and bias ($i=0, \ldots, n$):
   $v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$.

6. Test stopping condition.

Epoch is the term used to define one cycle through the entire set of training vectors (*Fausett 1992, Nasir et al. 2009*). Many are typically required for the complete backpropigation training of the neural network. The algorithm updates the weights after each training pattern is presented.

Generalisation

Reasonable answers or predictions are capable of being made by properly trained multi layer networks when presented with unseen inputs. If the new inputs are similar to inputs used in the training data set, an accurate output is typically produced (*Demuth and Beale 2013*). ANNs may be thought of as a group of generic filters which store information in a dispersed form. The sample data form is changed into a new form depending on the training algorithm and architecture of the network used. This stored information may consist of pattern classifications samples, data regularities, or temporal behaviour predictions of a dynamical system. Implementing the same data in combination with different networks could accomplish any of these storage cases (*Lisboa 1992*).

The inherent nonlinearities and the collective action of the numerous individual elements give rise to this generalising property of the system. This enables a pattern completion capability, making it possible to train a network with only a representative set of input/target pairs and get good results (*Demuth and Beale 2013*). Therefore, example data presented with missing or corrupted info leads the network to recall the completed stored pattern, with the corrupted information filled in or corrected. This is referred to as an associative memory capacity. New related patterns will activate the network to recall or interpolate a response which is intermediate between the most appropriate responses related to the stored patterns (*Lisboa 1992*).

Often during the training process a problem referred to as over fitting may occur if the network is not trained correctly. This evidently occurs when the training data set predictions have been driven to very small error values. In this case the network essentially memorizes the training set, and has not learned to generalize to new conditions (*Demuth and Beale 2013*). Hence there will typically be large errors when unseen data is presented to the network. Therefore the trained network will be ineffective at interpolating new data points.

There are alternative measures employed to ensure that over fitting is avoided and a network is trained effectively so that it is capable of generalising new data points well are trained properly. One clear method for improving network generalization is utilising a sufficiently large NN to give an adequate fit (*Demuth and Beale 2013*). It is evident within the range of reviewed literature that more complexity in the networks computing functions are introduced as the networks size is increased (*Lippmann 1987, Demuth and Beale 2013*). Thus, a small enough network structure will not have enough power or complexity to overfit the data. However, difficulty arises in knowing and establishing the sufficient size of a network for its given application.

It was noted by Demuth and Beale *(2013)* that there is a considerably reduced chance of over fitting if the quantity of network parameters is significantly less than the amount of points within the training set. Hence, providing additional training data for the network is also more likely to produce a network that generalizes well to new data. This is quite evident in the work of Nasir and Yousif (*2009*), where they used a training data set consisting of greater than 7000 points of data. This has also been clearly noted and expressed numerous related works. It is clearly distinguished in the works conducted by Zhang et al and Jiang et al. Within their work they make comparisons with the amount of a given data set required to achieve specific levels of correlation coefficient, also referred to as R values.

However, in relatively large data sets or additional data may not be available and supplied data may be limited. Such cases call for alternative methods that make effective use of the supply of limited data. Demuth and Beale *(2013)* recognise two alternative generalisation techniques commonly implemented as regularization and early stopping. These are two features that are incorporated in the Neural Network Toolbox software to aid in improving network generalization.

Data

A set of examples of proper network behaviour including inputs p and target outputs t is required for the training process. For MATLAB use, the data is generally divided into three subsets (*Demuth and Beale 2013, Nirmal 2010, Nasir et al. 2009*). The training set is the first subset, which is implemented to compute the gradient and to update the weights and biases. The second subset is the validation set and is used to monitor the error throughout training. This error along with the training set error typically decreases in the initial phase of the training. Error on the validation set will tend to rise as the network starts to overfit the data. Training cycles are therefore discontinued as network weights and biases are stored or saved at the minimum error of the validation set.

2.3 Tribology & ANN Applications

2.3.1 Tribology

Tribology is a topic that has developed over time within the mechanical engineering discipline and is an important aspect of material selection for new component design. Essentially the topic covers the science of wear, friction and lubrication (*Yousif 2013*). As stated understanding the tribological performance or properties of material has become important for material selection in some component design situations. An example would be the consideration of wear and friction in the design of a light weight composite bearing. Asperity interaction in contact controls these tribological behaviours. Topography and other modifications on the surfaces of the interacting materials are influenced by the frictional heat and shear force in the interface region during the sliding or rubbing (*www.tribology-abc*).

Many industrial components are placed under tribological loading. Most of these components experience failure due to this form of loading. It has been well established that there are several conditions or parameters that may influence the tribological performance of a material. Some of these influential factors include the sliding distance, velocity, normal load force, contact conditions, contact mechanisms, material structure. Conditions of contact may refer to wet or dry contact. Point, line or areas are referred to as mechanisms of contact. Material micro structure is also recognised to be of significant importance particularly with the increasing development and applications of new polymers and composite materials (*Yousif 2012*).

2.3.2 Tribology Testing

Materials with different microstructures under various contact mechanisms, contact conditions and operating parameters have had much attention in investigations into wear behaviour. Investigations conducted by Bansal et al. (*2011*) and Narish et al. (*2011*) highlight sliding distance, sliding velocity and applied load as some common operating parameters. Work done by Yousif and El-Tayeb (*2010*) identifies considerations to conditions of dry verses wet contact. Line, point and area mechanisms of contact have also been investigated (*Yousif and El-Tayeb 2008*). The effect of material micro structure has also been investigated in the works of Jawaid et al. (*2011*) and Siddhartha et al. (*2011*).

Numerous designed and standardised tribological apparatus have been employed to study the material behaviour in relation to the identified influential factors. Most of the laboratory machines have been designed and fabricated to conduct investigations based on individual techniques. These include block-on-disk (BOD), block-on-ring (BOR), wet sand rubber wheel (WSRW), dry sand rubber wheel (DSRW), and sand/steel wheel (SSW) test in wet/dry conditions (*Yousif 2012*). The key difference between the test techniques is primarily the tested material's method of contact with the counter-face. This is clearly evident in the depiction of each of these common techniques in the following figures.

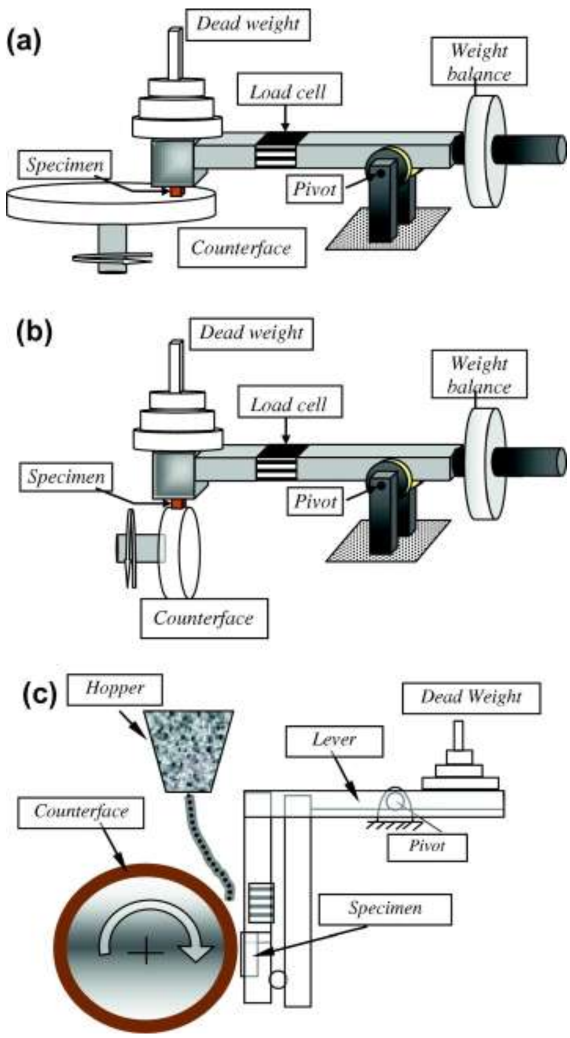

Figure 13 - Schematic drawing showing the most common configurations of tribological machine for adhesive and abrasive testing. (a) block on disc (BOD), (b) block on ring (BOR) and (c) dry sand rubber wheel (DSRW) (*Yousif 2012*).

Figure 13a depicts the standard BOD test set up according to ASTM G99-05. The standard BOR technique as defined by ASTM G77-98 is illustrated in Figure 13b. The technique setup for DSRW, WSRW and SSW tests in line with ASTM G105 and ASTM B611 is shown in Figure 13b. Figure 14 depicts a newly developed testing apparatus that is currently in use within the testing laboratories of the University of Southern Queensland. The machine is essentially able to perform each of the outlined testing mechanisms. It is also capable of conducting both BOD and BOR testing mechanism simultaneously, reducing considerable additional testing time. The apparatus has load cells (Accutec H3-50 and B6 N-50) equipped on the BOR and BOD load levers to measure the contact frictional forces. Infrared thermometers (Extech 42580) are also equipped to the on the rig frame and directed toward the contact areas in order to record interface temperature (*Yousif 2012*).

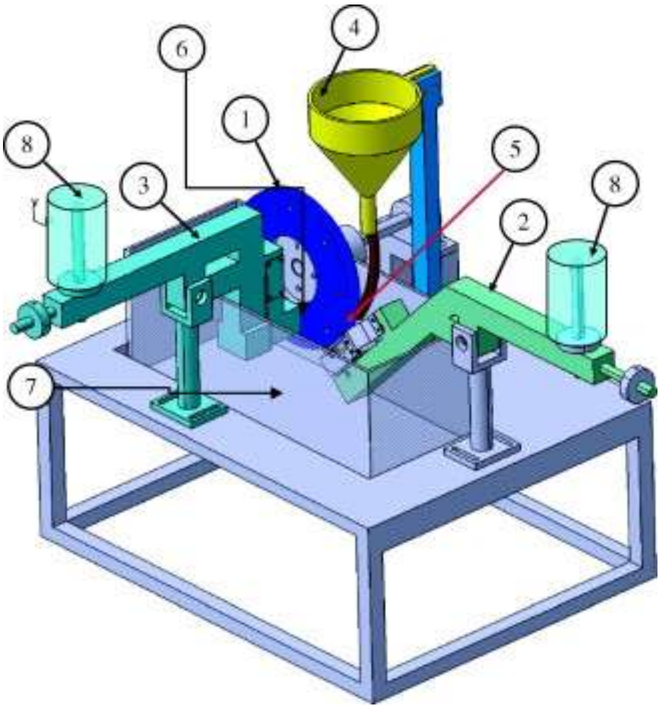

Figure 14 - A three dimensional drawing of the new tribo-test machine. 1-Counterface, 2-BOR load lever, 3-BOD load lever-, 4-third body hopper, 5-BOD-Specimens, 6-BOR-Speceimen, 7-Lubricant Container, 8- Dead weights (*Yousif 2012*)

2.3.3 Materials

Societies increasing focus toward awareness for the environment has driven the development within the fibre composite industry Sustainable, environmentally friendly, materials have subsequently emerged in popularity. There is also recent concern for the sustainability and limited nature of resources used in traditional petro-chemical based composites (*Yousif 2009b*). This has lead to recent and growing interest in implementing natural fibres for polymeric composite reinforcement. Properties like their low density, non-abrasiveness, non-toxicity, biodegradability, renewability and low costs have also driven this interest (*Chin and Yousif 2009*). Their higher specific properties like modulus, flexibility, strength, and impact resistance also make them attractive.

The study of tribology has thus developed as an important aspect of material selection for new component design (*Yousif 2013*). Many industrial components are placed under tribological loading and experience failure due to this form of loading. Numerous recent studies have thus been conducted and are still yet to be completed on the tribological behaviour of these newly emerging natural fibres. These studies will aid the employment of such materials within industrial component applications.

Fibre Composites

A composite is generally a material made from two or more different phase types, each with varying material properties. Constitutes of the material are selected to achieve desired specific material properties (*Mano1991*). One component (fibre) will reinforce the other component (matrix) structurally. The polymer matrix or secondary phase provides a means of load dispersion and ensures the primary phase or reinforcing fibres remain in position by adhesion (*Kaw 1997, www.mdacomposites.org*). Fibres vary from fillers or particular reinforcements by their much greater display a length to cross section ratio (*Matthews & Rawlings 1999*).

<u>**Resins**</u>

Polymer resins are typically used as the matrix for many modern commercial fibre composites. Polymer resins are primarily categorised as thermoplastics and thermosets (*www.mdacomposites.org*).

<u>Fibres</u>

Bunsell and Renard (2005) categorise fibres as synthetic, regenerated and natural. Plant, mineral and animal fibres are used to subcategorise the natural fibres. Typical synthetic fibres include nylon, glass and carbon. Hemp and flax from plants, wool from animals and asbestos minerals are some recognised natural fibres. Long filaments processed from a plants molecular structure represent regenerated fibres (*Bunsell & Renard 2005*).

Composite properties are intimately associated with the properties and content of the reinforcing fibres. Most research and testing characterise the fibre content of a composite in terms of either a weight or volume fractions, relevant to fabrication or property calculations, respectively (*Matthews & Rawlings 1999*).

Literature reports have identified that the degree of adhesion or the matrix bond quality has a significant influence on the composite properties (*Chin and Yousif 2009*). Flexural strength, compression strength, traverse tensile strength, fracture toughness, in-plane shear strength and wear performance are all influenced by adhesion. Matthews and Rawlings (*1999*) note that the fractions of weight and volume can modify the matrix to fibre bond quality to some degree.

<u>Natural Fibres</u>

This project focuses on ANN development to characterise the tribological characteristics of a kenaf fibre reinforce epoxy composite. Various advantages of natural fibres are their lower expense with higher specific properties, ease of processing; recyclability and renewable supply with a reduced carbon foot print (*Chin and Yousif 2009)*. Table 2.1 presents a comparison of the mechanical properties of some common natural fibres and traditional fibres.

Table 2.1 - Some common natural fibre and traditional fibre mechanical properties

Fibre	Specific gravity (g/cm^3)	Tensile Strength (GPa)	Tensile modulus (GPa)	Specific strength (g/cm^3)	Specific Modulus ($GPa/g.cm^3$)	Cost ratio
Sisal	1.20	0.08-0.5	3-98	0.07-0.42	3-82	1
Flax	1.20	2.00	85	1.60	71	1.5
E-Glass	2.60	3.50	72	1.35	28	3
Aramid	1.44	3.90	131	2.71	91	18
Carbon	1.75	3.00	235	1.71	134	30

Kenaf is a plant based fibre, the structure of a plant fibre can be seen below in Figure 15. A plant fibril is basically structured with a primary cell wall surrounding a secondary wall (*www.ccrc.uga.edu)*. Growth rate, structural support and cell interactions are the responsibility of the primary cell wall. Bulk mechanical strength is given by the three layers of the secondary wall. The middle lamella, referring to the fibres outer layer provides stability by fixing together adjoining cells. The fibres themselves may be perceived as a composite, with mainly cellulose fibres secured in a matrix of lignin and hemi-cellulose. Thus, the reinforcing cellulose content is in direct relation modulus and tensile strength (*www.ccrc.uga.edu*).

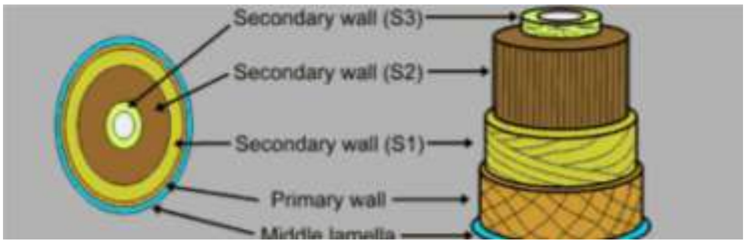

Figure 15 – Plant fibre structure (*www.ccrc.uga.edu*)

2.3.4 Kenaf Fibre Reinforced Epoxy Composite (KFRE)

Many recent studies have established the superior properties exhibited by kenaf fibres over other commonly used natural fibres such as jute, sugar cane, and oil palm. The kenaf fibres have also been shown to demonstrate strong interfacial adhesion between the fibres and the matrix (*Chin & Yousif 2009*).

Little research has been conducted regarding the usage of natural fibres as polymeric composite reinforcements for tribology applications. Subsequently, Chin and Yousif (2009) have conducted work assessing the potential of kenaf fibres for reinforcement in polymer based tribo-composites. Their work has assessed the composite's specific wear rate, contact surface friction coefficient and contact interface temperature. The assessment was made in relation to sliding distance, applied load, sliding velocity and fibre orientation with respect to sliding direction as the controlled parameters.

The previous experimental work was conducted using 10mm x 10mm x 20mm test specimens of the composite prepared by closed moulding and machining. The resin used was widely used liquid epoxy (DER 331). JOINTMINE 905-3S was utilised as the curing agent, uniformly mixed in a 2:1 ratio of epoxy to hardener. About 48% volume fraction of fibre were used within the matrix. Fibre diameters range between 0.25 – 0.4mm. Table 2.2 lists some of the properties of the neat epoxy and the KFRE composite.

Table 2.2 – Neat poxy and KFRE composite Specifications (*Chin & Yousif 2009*)

Property	NE	KFRE
Fibre volume fraction	0%	48%
Density, kg/m^3	1100 ± 2	850 ± 2
Modulus of elasticity, GPa	20 ± 2	14.5 ± 2
Tensile strength, MPa	130 ± 5	135 ± 2
Elongation, %	3.5 ± 0.2	6.5 ± 2
Thermal conductivity, W/(m K)	0.17	0.11

A BOD machine, as depicted in Figure 16, was used to conduct the tests on the specimens against AISI 304 stainless steel. Before each test, strict procedures were followed to prepare both the steel and specimen counter-face to ensure high intimate contact. Tests were conducted at different sliding velocities (1.1-3.9 m/s), sliding distances (0-5 km) and applied loads (30-100 N) at a 28˚C room temperature. This was done for the parallel (P-O), anti-parallel (AP-O) and normal (N-O) fibre orientations (Figure 17).

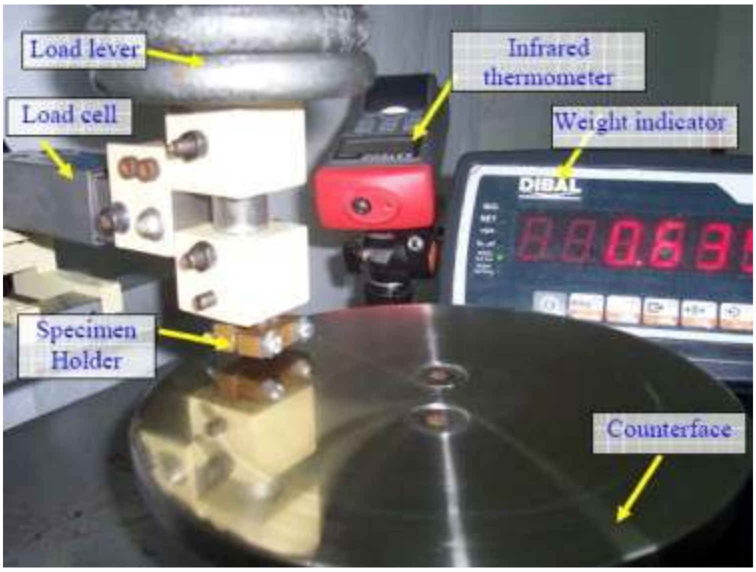

Figure 16– Pin-on-Disc machine (*Chin & Yousif 2009*)

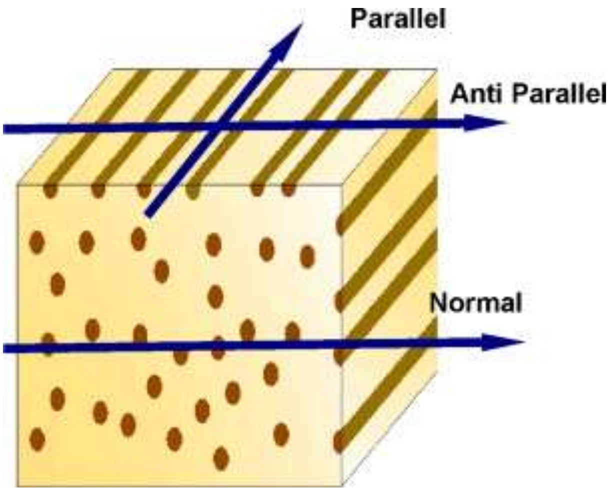

Figure 17– Orientation of fibres with respect to sliding direction (*Chin & Yousif 2009*).

Each test was repeated three times and the average measurements were derived. Friction force was measured by load cell on the load lever and interface temperatures were recorded by an infrared thermometer. Specimen weights were tested before and after tests to calculate weight loss and subsequently the specific wear rate (mm^3/Nm) at each operating condition. The graphs presented within the report of the work depict and compares some of the resulting data. These figures are presented in appendix (?).

2.4 Risk Management

2.4.1 Introduction

A risk assessment is involved in the consequential effects of this project. Safe guards and associated risks need to be documented. Throughout and outside the execution of the project certain risks are likely to be encountered. Subsequently, it is imperative to establish a level of continuing responsibility.

2.4.2 Identification of Risks

Since the course of the project is primarily computer based, there are no identifiable direct risks associated with the project work. However there are several risks that are identifiable for the related works from which necessary computational data has be gathered. The primary risks associated with this related outside work can be summarised as sample preparation, testing, maintenance, and project sustainability.

Sample preparation involves risks related to the handling synthetic and/or natural fibres, hardeners, resins and other fibre treatment chemicals. Greater risk is presented during the shaping of the composites by means of cutting and polishing to size. Operators involved in this process typically exposed to elevated equipment noise, airborne particles and spinning disks and/or blades. There is the potential for both long term and short term operator injury for these identified hazards. Lose of limbs, hearing or vision impairment, skin irritations and impaired breathing highlight the range of possible injuries.

Operator error and inflicted injury by released airborne testing fragments are the form of risk considered in the testing stage. Depressing the wrong machine buttons may result in limbs being crushed. This represents the occurrence of injuries due to insufficient operator confidence and training. Maintenance also reflects in tidiness and general areas of risk include slippery surfaces from spills, correct labelling of chemicals and equipment, work area cleanliness and trip hazards. Risks relevant to the sustainability of the project work involve the environment and direct future users. Disposal of no longer required or used materials presents environmental risks. Considerations to particle emissions and power use are also required.

2.4.3 Evaluation of Risks

Low levels of risk are associated with most of the risks identified in the previous subsection. If materials are handled correctly by the operator they are harmless and this preparation phase of the sample presents a low risk to the operator. However, potential for injury still exists if incorrect handling occurs. Encountering injury during shaping preparation of the sample has the higher risk probability.

Minor to moderate levels of risk may be associated with the mechanical cutting and polishing devices. Permanent injury possibilities arise if these machines are utilised incorrectly. Examples include cuts or amputations of limbs due to blade or disc breakage. Minor to moderate risk to eye injury is perceived in relation to projectile debris from polishing or cutting. The level of polishing or cutting dust also presents moderate risk of lung damage. Due to the machines situated distance from the operator and the clear protective coverings/shields, the testing stage has only a minor probability of operator. As the machine is mostly remotely controlled, the associated injuries caused by twisting and crushing are not likely. The presence of clear viewing shields around the machine should make the potential of any injuries inflicted by projectile debris an unlikely event.

Regular scheduled maintenance and cleaning of the labs along with immediate cleaning of equipment after use indicate maintenance risks as unlikely events. Materials used are mostly natural and may be reused. The non-recyclable materials such as the epoxy resin are used in significantly small, none threatening, amounts. Risks to the environment are therefore considered low.

2.4.4 Risk Control

The following action plan should be implemented to minimise risks, before undertaking tasks.

1. Understand the task

If further testing and data collection is required for further or other related work it is essential that all tasks are explained by supervisors and technicians before conducting tasks.

2. Complete relevant training

Safety inductions relating to handling materials, machine operation and safety actions need to be incorporated with demonstrations during operator training.

3. Identify risks

Informal job safety assessments (JSA) should be carried out identifying any risks before commencing any operations.

4. Reduce or control the risks

Additionally, any risks should be minimised by employing protective controls. This includes utilising personal protective equipment (PPE) or immediately cleaning spill are examples.

3 Research Design and Methodology

3.1 Introduction

3.1.1 ANN Development Process

The following chapter is separated to address the typical steps that were conducted in developing the most optimal ANN prediction tool the tribological characteristics of a KFRE composite. The general consensus from the literature regarding the systematic optimal ANN development process is presented in the flowchart in Figure 18. Initially previous experimental data is to be collected and processed for use to train and test the network. Following this is the generation of an optimal network model derived through a series of attempts. The resulting optimal model is further trained to hopefully achieve greater accuracy before it is finally tested to simulate predictions and compared to with experimental data. Within the continued training process it is also recognised that there needs to be point of termination, such that the model will not over fit the training data and will capable of effective generalisation. Further investigation into generalisation of the network will be carried out using previously established techniques.

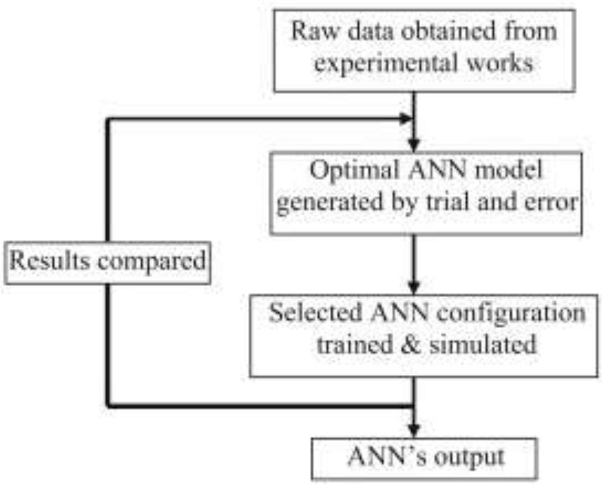

Figure 18 - Flowchart illustrating steps in developing the ANN model (*Nirmal 2010*).

3.1.2 Implementing MATLAB

MATLAB has been recognised as an effective neural network modelling tool and is subsequently used to carry out the project. The Neural Network Toolbox function provides varied levels of complexity in which the user is able produce ANNs. The first level is represented by the Guided User Interfaces (GUIs) that provide a quick way to use the toolbox function for many problems of function fitting, pattern recognition, clustering and time series analysis. The toolbox may used through basic command-line operations that use simple argument lists with intelligent settings for function parameters. Customization of the toolbox is an advanced capability that permits the creation of custom neural networks still with full functionality of the toolbox. Every computational component is written in MATLAB code and is fully accessible (*Demuth and Beale 2013*).

The toolbox will enable the user to setup custom networks and essentially trains the networks by means of specified training algorithms. To train the networks the toolbox divided supplied training data into training, validation and test subsets. These are used to evaluate the networks performance after each training epoch. Once the training is stopped by a specified method or condition the toolbox can present summarised training information.

The general MATLAB code setup for creating the various network configurations and assigning specific training algorithms and termination parameters is presented in Appendix C. The presented code was implemented and manipulated throughout the project to derive and train various networks with three hidden layers. Two other modified versions of this code were utilised in the same manner for the single and double layer systems. The setup of this code was guide by the literature presented by Demuth and Beale *(2013)* and the MATLAB Toolbox.

During training a training window will appear, like the one presented as Figure 19. The window displays to the user the data division function, training method and performance function used to train the network. The progress of the training is constantly updated in this window. Also presented is the performance, the magnitude of the gradient of performance and the number of validation checks. Various methods such as minimum gradient magnitude, training time, number of training cycles and the number of validation checks are used to terminate the training. As the training reaches a minimum performance value the gradient will become very small. The number of successive training iterations that don't yield lower performance values is represented by the number of validation checks. If the default or nominated values for either the gradient magnitude or validation checks are reached the training is stopped.

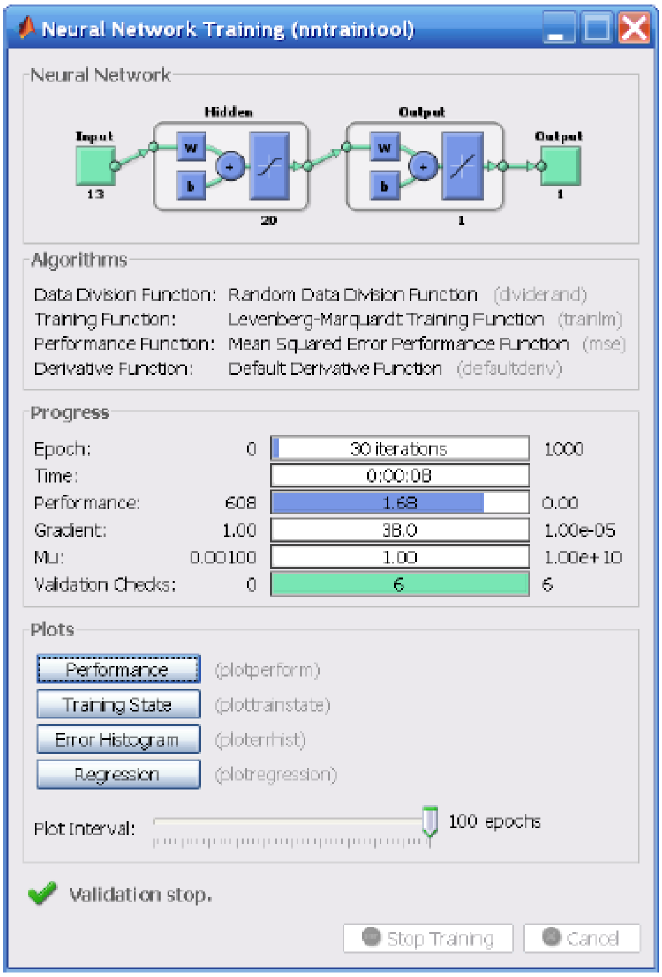

Figure 19 – Example MATLAB training window

The performance, training state, error histogram and regression plots can be accessed from the training window. The value of the performance function for the training, validation and test subsets are plotted against the iteration number in the performance plot. The other training variable like the gradient magnitude and number of validation checks have their progress plotted in the training state plot. The plot of error histogram depicts the network error distribution. The regression plots may be used to validate the performance of the network as it shows a regression between network outputs and network targets for each of the data subsets.

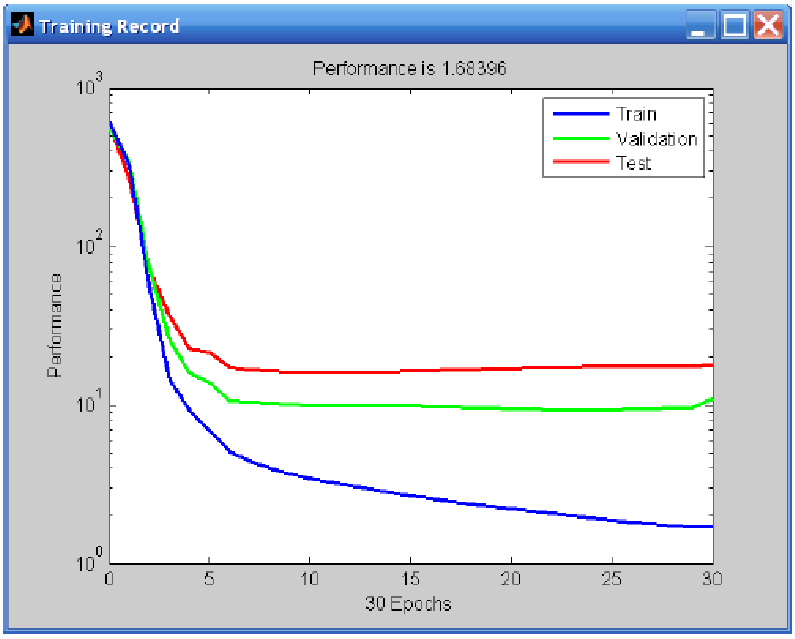

Figure 20 – Example of MATLAB performance plot

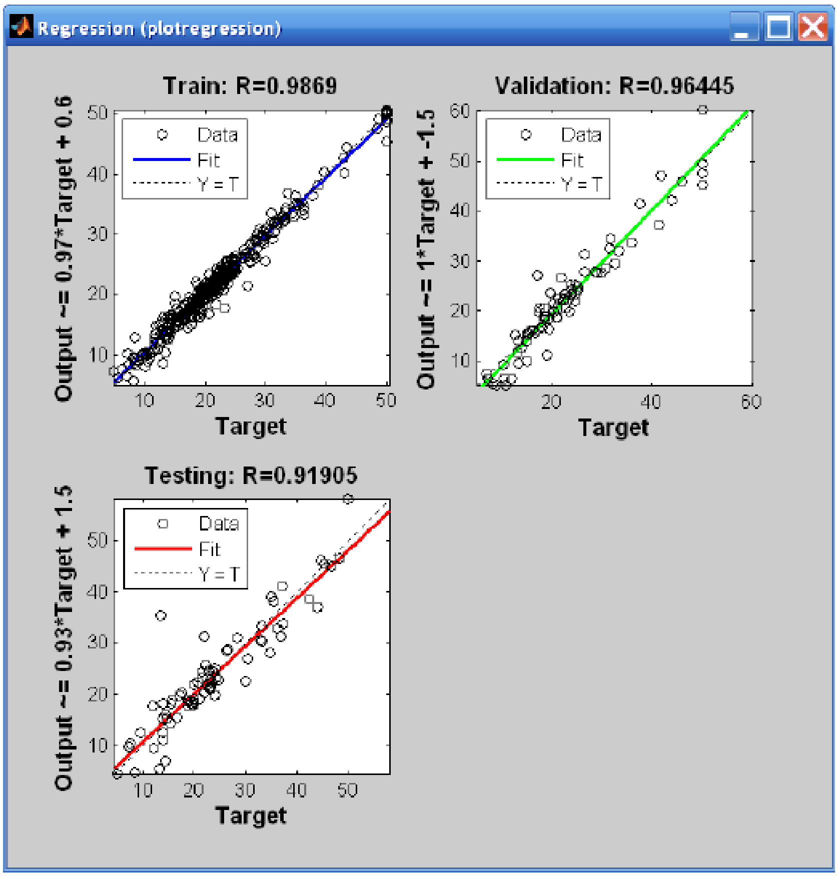

Figure 21 – Example of MATLAB regression plot

3.2 Collect & Process Data

The experimental data on the KFRE composites was a small portion of the data supplied by the project facilitator that was collected from the unpublished work currently being conducted. The based on this collected and presented data the neural network will be capable of being trained to calculate two tribological characteristic outputs based on three tested parameter inputs. The outputs are the contact surface temperature, and the coefficient of friction. The parameter inputs characterising these outputs are the fibre sliding distance, sliding velocity and applied load.

The portion of data provided contained a total of 305 data points. It was desirable to utilise the remaining data set for the later simulation and predictions. However this work was still yet unpublished and incomplete. Thus, the remaining data would not be used in the validation and assessment steps. This was to respect privacy and confidentiality and to avoid dependency on the unpublished work. The alternative was to extract 10 percent of the data points from the given data and utilise this to assess the networks overall predictability.

This method of dividing the data is recognised and employed as an effective method for assessing predictability for unforseen data in various other related works (*Nasir et al 2009, Aleksendrik & Duboka 2007*). This testing subset of the data is extracted from the network by utilising a randomly dividing algorithm commonly employed MATLAB before the NN is trained. Thus, the values extracted for the test subset will vary for every training session or reinitialized training. Having several different subsets during repetitive training on various networks and configurations will ultimately verify the robustness of the performance of the networks.

This provided training data set has been included as. It's clear that this is not a complete set of data based on the completed sets described in other related tribology works (*Nasir et al 2009, Chin & Yousif 2009*). The data is subdivided into major categories. The major categories define the speed at which the test specimens were tested. There are data points presented for only two speeds, 2.8 meters per second and 1.1 meters per second. Under each of these are two sub categories for the outputs, which are temperature and the friction coefficient. There are successive data points for each of these outputs exists under four sub-categories specifying the load used to apply the normal force to the test specimen. These loads are 30 newtons, 50 newtons, 70 newtons and 100 newtons.

The successive data points are related to a specified sliding distance at which it was recorded during the test. The sliding distances are spaced evenly at 0.084 kilometres, starting at zero and ending at 5.04 kilometres, for a total of sixty one recording intervals. As mentioned the data set is incomplete, only presenting data points for the 50 Newton load category for the test speed of 1.1 meters per second. Previous related tribology studies also indicate datasets containing similar categories, with other test speed settings of 1.5, 3.1 and 3.5 metres per second (Nasir et al 2009, Chin & Yousif 2009).

For the data to be implemented in training the networks in MATLAB they were converted into MATLAB matrix files. In this case the number of columns in the input and output matrices represent the number of network input and output parameters, respectively. The dataset was re-organised to make it more presentable or readable as inputs and outputs for training the NNs. This was simply completed by making three input columns and the two output columns. All the temperature and friction coefficient data points are arranged into these two columns. The three input columns represent the sliding distance, the force and the velocity respectively. The data inputs were arranged into these columns so they are associated with the corresponding output data. The inputs and the corresponding outputs were then imported as two separate input files, which are presented as Appendix A.

From this data, a neural network will be trained and developed to closely approximate friction coefficient and temperature based on sliding speed, sliding distance and load. This dataset also highlights the need for developing ANNs as predictive tools. As they can be trained with a portion of data to hopefully provide accurate predictions and to interpolate unseen data points not provided within a given data set.

3.3 Generate Optimal Model

Generating the optimal ANN model involved the selective series of attempts with different neural, function and layer configurations. Selection of the most optimal configurations for the three separate considerations was completed by comparing the network performances based on the relative mean squared error (MSE).

The process of developing an optimal ANN model was completed through a systematic trial and error approach. This involved setting up various network structures within Matlab that were trained with a specific set of collected data. Evaluations and comparisons of the performance of each of the network structures were conducted primarily based on their MSE. During the training of the networks the performance and performance curves will generally converge towards zero. This means that the MSE is approaching zero and a higher performance of the system is being achieved (*Demuth, H. and Beale, M. 2013*).

The alternative sum square error (SSE) is also employed for the same purpose as it gives similar representation of the performance. However the MSE gives a better idea of relative error at each data point. Both are recognised by many texts to be effective comparative values in terms of function fitting (*Demuth, H. and Beale, M. 2013*). It is also employed in various related works and is set as the default calculated value for measuring performance within the NN Toolbox in MATLAB (*Jiang et al 2007, Zhang et al 2002*). Hence, judging the performance based on the models error clearly established the ability of the networks to accurately predict the desired outputs based on specific inputs.

In obtaining these performance values it was recognised that the results for a particular setup would vary each time the network weights and biases were initialised. Different initial weights and biases would be selected marginally influencing the performance after the final epoch of each training session. Thus each network setup was trained and the performance recorded three times each so that an average would be obtained. This average was expected to provide a general sense of the training performance for each network setup. Testing several varied initial conditions validates a robust performance of the network. Subsequently, the performance averages were used as the primary figure in comparing the networks. All training performance results and calculated averages were logged in tables within excel. These logs are included as Appendix D.

The systematic steps taken in developing the model included initially establishing the desired combination of transfer functions between each of the network neural layers. Next the various training methods provided within the NN Toolbox were employed and assessed on their influence to the performance of each of the setup networks.

Finally, the numbers of layers were established along with the numbers of neurons within those layers. The reason for testing the networks in this sequence was to try covering as many of the endless potential network setups as possible with as few network configurations tested. Zhang et al (*2002*), Jiang et al (*2007*) and Jie et al (*2007*) are some of the related works from current literature that have adopted a similar process to the optimization of the neural network configuration.

3.3.1 Select Transfer Function

As previously established in subsection 2.2.3 of Chapter 2, the three most common transfer functions used in ANNs are log-sigmoid, tan-sigmoid and pure-linear. Each function is equipped to the ANN MATLAB toolbox. It's recognised from the literature that the transfer function for the output layer would typically be pure linear since the outputs results are required to be any numerical value (*Demuth and Beale 2013, Nasir et al 2009*). Therefore, testing for the optimal transfer function for the hidden layer/s was completed by training a few random networks with respect to each of the functions and comparing their performance values.

The literature reviewed identified that there is a clear relationship between the effective number of layers and the complexity of the input to output relationships. This complexity of the network is also influenced by the number of input parameters that have an influence on the outputs. Generally, the more complex the relationships the more hidden neural layers may be employed within the network structure. The typical range for the number of hidden layers employed in most other works is between one and three. This is because the complexity of relationships that most networks are being designed to make predictions for can generally be accomplished with these lower hidden layer networks. It should also be noted that implementing too many hidden layers for relatively smaller data sets may result in potential over fitting. Subsequently, in conducting the systematic search for the optimal ANN, multiple networks containing between one and three hidden layers were trained and assessed.

The purpose for assessing transfer functions for each hidden layer setup and with different layer volumes was to provide a greater scope of all network possibilities. Therefore, any preferences or influence to preference in transfer functions in relation to the number of hidden layers and their volume could be identified. The network neuron configurations trained and assessed were randomly selected for the purpose of establishing general relationships and preferences of transfer functions.

3.3.2 Select Training Function

Training has already been established as the process in which the weights and biases are modified to achieve greater performance. There are various training functions employed to carry this out and Table 3.1 presents a list of the available algorithms in MATLAB. These training methods dictate the means of adaptation. With the optimal transfer function selected for the hidden layers the various training functions were used to train the same general networks. The comparison of the performance values of each function revealed the optimal function for data set and transfer function.

Table 3.1 – MATLAB training functions and associated algorithms.

Algorithm	**MATLAB Function**
Levenberg-Marquardt	trainlm
Bayesian Regularization	trainbr
BFGS Quasi-Newton	trainbfg
Resilient Backpropagation	trainrp
Scaled Conjugate Gradient	trainscg
Conjugate Gradient with Powell/Beale Restarts	traincgb
Fletcher-Powell Conjugate Gradient	traincgf
Polak-Ribiére Conjugate Gradient	traincgp
One Step Secant	trainoss
Variable Learning Rate Gradient Descent	traingdx
Gradient Descent with Momentum	traingdm
Gradient Descent	traingd

3.3.3 Select Layer Configuration

Different network layer configurations were created and tested to identify the most optimal network structure. This involved setting up numerous single, double and triple layer networks with varying neuron volumes. Each model was trained to 300 epochs and their performance values compared. After the comparison, the optimal number of layers for the ANN was established. Further models were constructed and tested to finally ascertain the optimal neuron volume for the preferred number of hidden layers.

3.3.4 Train and Test Generalisation of Selected Model

Further training of the derived optimal network is carried out by repetitively applying the training process to the ANN. This has been recognised to improve the models performance (*Nasir et al. 2009*) as the model accumulates improvement from the previous training session and adjusts for greater accuracy. As the training cycles continue to increase the network the error percentage drops and the performance of the system gradually converges until no further improvements are observable. It was also recognised that a more generalised network would be desirable since the training data represents only a portion of the full data set. The network weight and bias setups after certain amounts of training were stored so that these models may be used to assess their ability to generalise and predict the unforseen data.

3.4 Improved Generalisation

It is obviously critical to ensure the selected optimal network configuration is appropriately trained in such a way that will avoid over fitting and emphasise the networks ability to predict in a generalised sense. The literature reviewed indicated some alternative methods commonly employed in the training processes that are recognized to produce improved generalisation of a network.

3.4.1 Generalisation Technique

Early Stopping

The early stopping process requires the division of the data into validation and training sets. Computation of the gradient and updating of the weights and biases of the NN is performed with the training set. The error on the validation set during training is monitored. This error initially improves along with the training error before it converges to a minimum and then starts to rise as the training is continued. This is illustrated by the MSE plots for all three data subsets in Figure 22. This rise indicates that the network is beginning to overfit the training data set and its ability to generalise starts reducing (*Demuth and Beale 2013, Jie et al. 2007*).

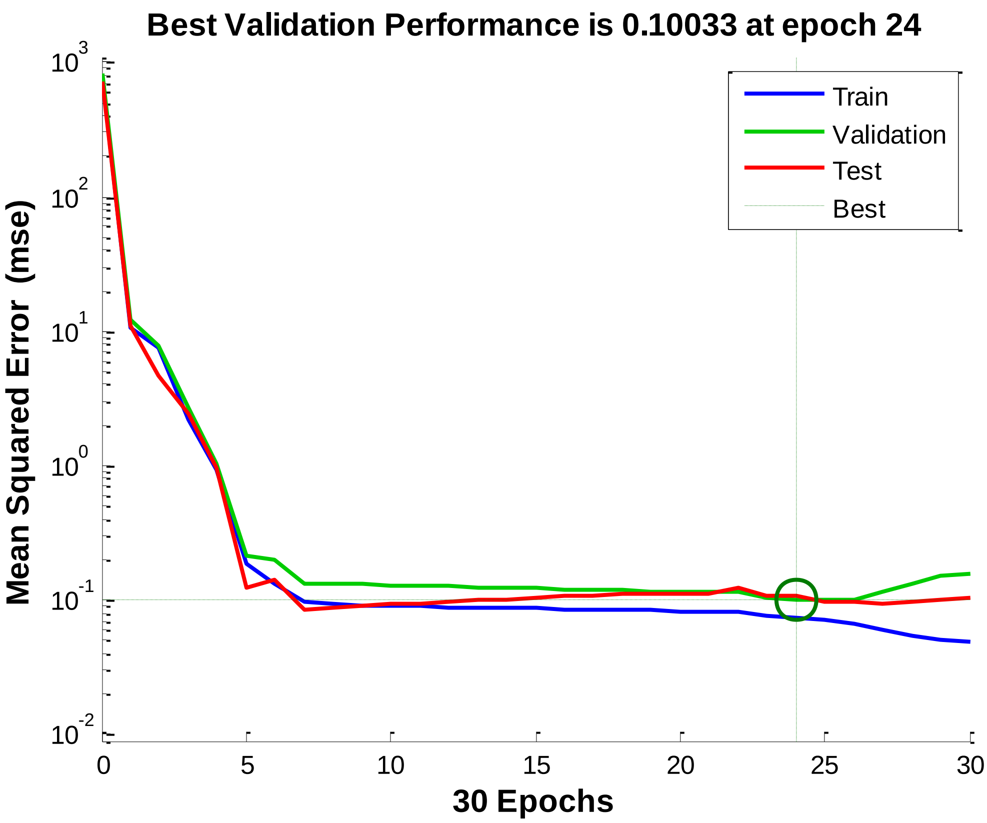

Figure 22 – MSE plot for all data subsets illustrating early stopping for a 3-[25-15-10]-2 network trained with the *trainlm* algorithm.

Generally the error of the validation set is allowed to increase for certain number of training iterations before the training is ceased. The network is then I returned back to the minimum validation error weights and biases. This technique is set up as the default method for improving generalization within the MATLAB software. The division of the data is adjustable through various division functions. Each utilises parameters that customize the networks training and behaviour.

Bayesian Regularization

The second alternative method implemented in improving generalization is regularization. The Bayesian Regularization training function trainBR performs regularization automatically within its training process. The training function essentially achieves improved generalization through modification of the performance function by the addition of a term containing the mean square weights (MSW) of the network (*Demuth and Beale 2013, Jie et al. 2007*). This is demonstrated within Equation 3.1, where γ is the performance ratio. MSW is given by equation 3.2, representing the mean of the sum of squares of the network weights (*Demuth and Beale 2013*).

$$\text{MSEreg} = \gamma\text{MSE} + (1 - \gamma)\text{MSW} \tag{3.1}$$

$$msw = \frac{1}{n}\sum_{j=1}^{n} w_j^2 \tag{3.2}$$

Updating the bias and weight values of the network is generally performed with the Levenberg-Marquardt training function, *trainlm*. To establish a well generalising network it minimises and then determines the correct combination of weights and squared errors. The networks trained with this function typically result in lower weight and bias values throughout the network. Consequently, a smoother network response is forced and data over fitting is less expected (*Demuth and Beale 2013*).

Referring back to the equation 3.1 it is evident that the performance ratio influences the weighting the MSE and MSW have on the regularized MSE used in training. If it is set to 0.5, there will be equal weight given to both. Over fitting often occurs when the parameter is too large and a smaller value may result in significant under fitting.

Determining the optimal value for the performance ratio is the difficult problem with regularisation (*Demuth and Beale 2013*).

As mentioned, the trainBR function performs automatic regularization during training. The division of the data into a validation set is stopped and all validation assigned values are added to the training set. This is performed so the regularised MSE is secluded from early stopping (*Demuth and Beale 2013*). Another notable feature of the algorithm measures how many effective network parameters (weights and biases) are being utilised by the network.

3.4.2 Validation

To assist in the validation of the networks performance and ability to generalise the MSE curve of the test set is plotted alongside the training MSE, as previously demonstrated in Figure 20. For further validation regression plots, as depicted previously in Figure 3.4, are also generated for each of the data subsets. Within the plots the ANN predictions are plotted against the experimental data showing the general relationship trend between the two.

A solid best fit linear regression line between the targets and outputs is plotted to depict this relationship trend. An R value, also termed the correlation coefficient, is derived from this regression line. This value provides an indication of the relationship between the targets and outputs. A dashed line is also presented to indicate the perfect result where the output and target values are equal. This line represents an R value of one, as there would be an exact linear relationship (*Demuth and Beale 2013*). The R values for the test data sets provide an indication of the accuracy and generalisation of the model. Typically, acceptable correlations have R values that are greater than 0.9 (*Nasir et al 2009*).

3.4.3 Train and Test Generalised Model

The most optimal generalisation training setup and network configuration was established based on comparing the performances obtained for the MSE and regression

R values. As mentioned the comparison between the trained networks involved simulating and comparing the R values for the extracted test data in order to give an indication of the networks ability to generalise and predict for new unseen data points. Further or continued training for the most optimal model was considered and carried out with the R value for the networks test data being closely monitored.

To verify a robust network performance the increased training session for the optimal model was trained repeatedly with new initialised weights and divided data subsets in each session. The continued training was ultimately stopped once the MSE gradient for the training data reached the minimum assigned value of 0.001. The most robust network weight and bias setup was stored so that the model may be reused to assess its ability simulate and make generalised predictions.

3.5 Simulate and Compare ANN Results

In the previous training and testing stage the final model was required to simulate for the extracted test data subset. The test data set is not implemented within the training of the network and is therefore considered as a set of new unseen data points. The network essential made predictions for these unseen situations. The MSE and R values for the test data set were produced and compared during the training. These were obviously used to indicate the networks ability to make closely accurate generalized predictions.

This is verified through additional simulations and plots of ANN predictions overlayed with experimental data for some specific input parameters. The ability to interpolate other points not within either of the training or test data is also assessed based on the approximated trend lines within these plots. Both the ability to interpolate inside and outside the domain of the total given data set is considered.

3.6 Resource Analysis

Required resources for the successful completion of this project are the experimental data supplied from research supervisors and the use of the MATLAB Neural Network

Toolbox. The Toolbox extension is supplied with the USQ MATLAB software. Access to this software is provided in any of the engineering computer laboratories.

4 Results and Discussion

4.1 Generate Optimal Model

4.1.1 Select Transfer Function

A particular transfer function was to be selected for each of the hidden layers. As identified there are three primary transfer functions that are commonly employed. As established from the literature the pure linear function is implemented in the output layer of all the networks trained since the output is desired to be in the form of any numerical value (*Demuth and Beale 2013, Nasir et al 2009*). Since this is a preliminary step, the training function implemented in this process was the Levenberg-Marquardt algorithm. This is defined as the *trainlm* function within Matlab and is the NN Toolbox default function.

The training of all networks at this stage of the network development was limited to a total of 100 epochs. This is identified as the number of times the networks is able to adjust the weights and biases to achieve a better result (*Demuth and Beale 2013*). This was deemed as suitable termination point enabling a sufficient amount training to then identify which of the transfer functions would provide the best performance from the network. This training termination point was also based on the work by *Nasir et al* (*2009*) which invovles a similar selection processes.

Single Hidden Layer Networks

To cover all possibilities, various transfer function combinations were tested for each of the network layer setups. Initially a single hidden layer system was trained and assessed with each of the three transfer functions in the hidden layer of the network. To ensure that the performance results and preference of the transfer function was not influenced by the neuron volume, a second single hidden layer system was trained. Therefore each network varied in neuron volume. The first consisted of twenty five neurons, whilst the second system contained forty. The results for both networks trained with each transfer function are presented and compared in Figure 23.

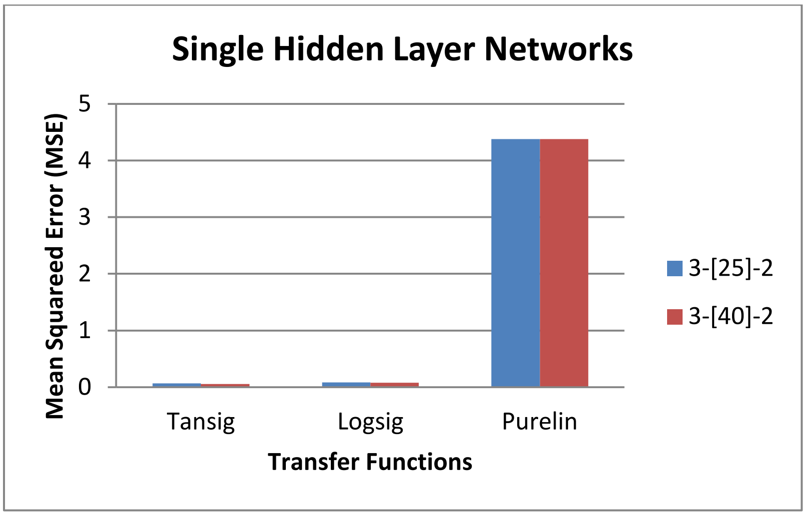

Figure 23 – Comparison of transfer function performance of single hidden layer networks

Figure 23 clearly indicates the pure linear transfer function as impractical for use within the single hidden layer. During the training with the *purelin* transfer function an error was produced and the networks training ceased after three epochs. This was as expected, validating the information presented in the reviewed literature. The performance would cease improving once it would reach a limit of 4.38 for the MSE. Thus, the gradient of the performance curve would reduce below the minimum gradient of 1^{-5}. This minimum value is set as one of the termination training parameters causing the network to stop training after only three iterations. Since the pure linear function is clearly not viable the other two possibilities are compared more closely in Figure 24.

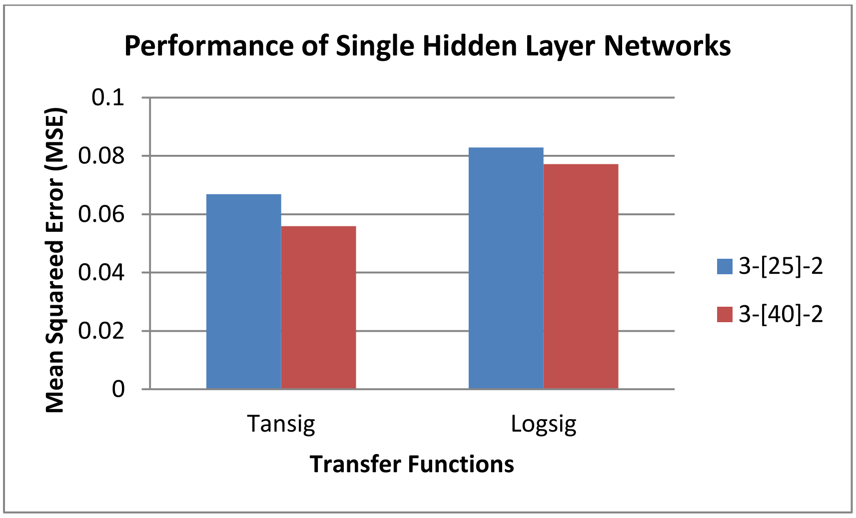

Figure 24 – Comparison of transfer function performance of single hidden layer networks

It's evident that the tan-sigmoid function outperforms the log-sigmoid function in the single hidden layer networks. In both networks the MSE is lower when the tan-sigmoid function was implemented. This also indicated that there was no change in preference as the node of neuron volume within the layer changed. The additional benefit of comparing the two networks is the indication of improved performance obtained from the greater neuron volume.

Double Hidden Layer Networks

The two double layered networks trained had 3-[25-15]-2 and 3-[15-5]-2 as their neural structure. The first and last figures represent the number of input and output parameters, respectively. Numbers within the brackets indicate the number of neurons in each of the hidden layers. The sequence of the numbers presented in the brackets is associated to the hidden layer sequence in the network. Figure 25 presents performance results for the transfer function combinations used in the 3-[25-10]-2 network structure. The label sequence on the bottom axis associated with layer sequence in which the named transfer function is employed.

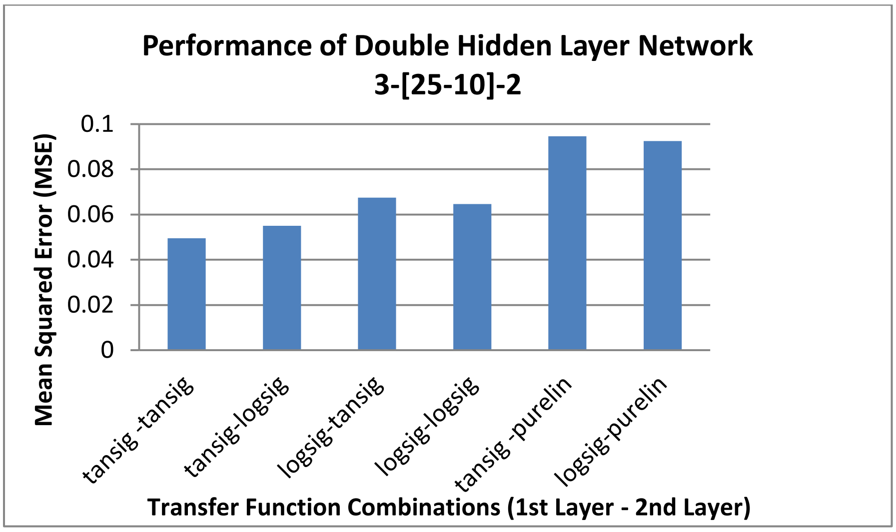

Figure 25 – Performance comparison of transfer function combinations in the 3-[25-10]-2 network.

In this case the *purelin* function does not limit the network performance due to the presence of the other calculative layer. However, based on the performance results on the single hidden layer systems it was expected that the solution implementing the pure linear function within the hidden layer would have diminished performance results. The results presented in Figure 25 clearly validated this expectation. The implementation of this function was therefore neglected when considering the following network structures. The presence of the tan-sigmoid function in the first layer appears to provide the better performance. To validate this and ensure preferences were not influenced by the neural volume the second network structure was trained. Figure 26 displays the results of the four tan-sigmoid and log-sigmoid combinations for both double layer networks.

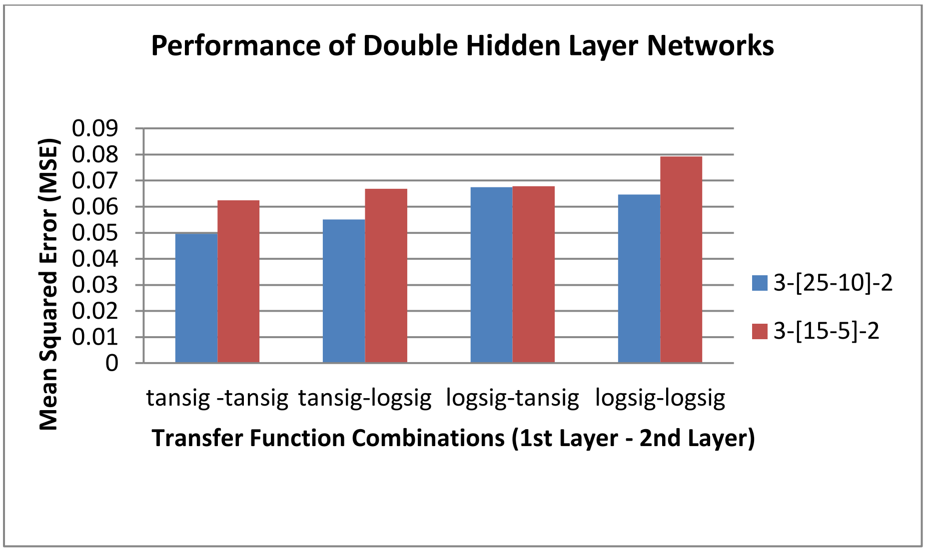

Figure 26 – Performance comparison of transfer function combinations in double hidden layer networks.

The second network, with the lower neural volume, appears to have a minor variation in preference. However, employing the tan-sigmoid function still proves to provide better performance. In comparing the performances of each network, the clear preference is the implementation of the tan-sigmoid transfer function in both hidden layers. This preference may have been assumed from the performance obtained in the single layer networks. The assessment of the single and double layered networks validated the tan-sigmoid function as the preferred transfer function in both setups. Based on this trend it was predicted that the same result would occur in the assessment of the triple hidden layer networks. Again there is a clear trend in the improvement of performance in relation to the neuron volume in the hidden layers.

Triple Hidden Layer Networks

The same process used to assess the double layer networks was performed similarly for the three layer networks. In this case the two networks that were trained were 3-[25-15-10]-2 and 3-[15-10-5]-2. There were eight possible combinations of transfer functions that were assessed. As noted previously the pure linear function was not considered due to its expected lower performance values. The trend of increased performance with increasing neural volume is again evident within the performance of these two networks. Based on the comparison of the performances presented in Figure 27 there is mixture in the preferences of transfer function setup. It was also recognised from the graph that all the results were relatively close, within a 0.01 range of MSE. All combinations could therefore have been considered as viable options.

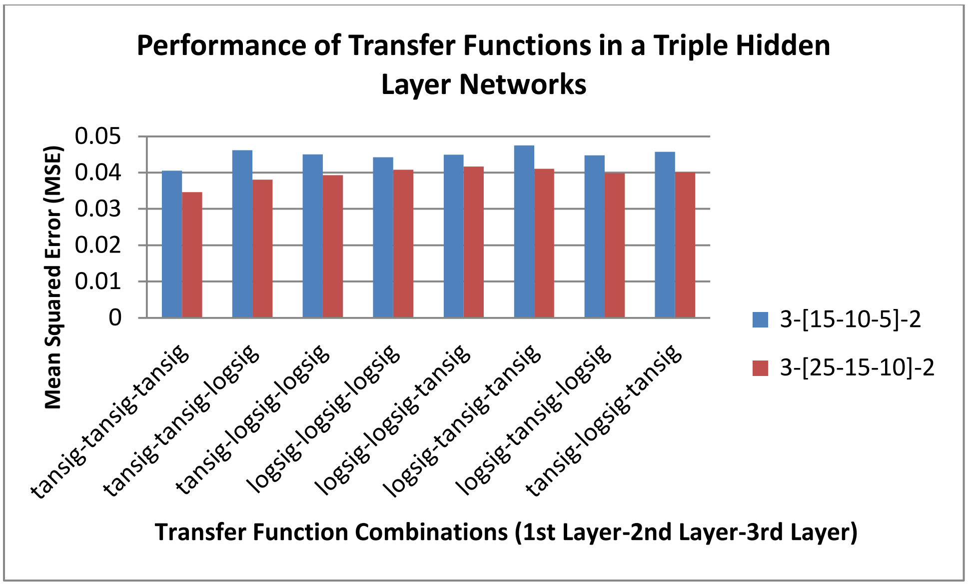

Figure 27 – Performance comparison of transfer function combinations in triple hidden layer networks.

The reason for the minor differences and less clear preference could possibly be due to the fact that the number of layers mitigates the influence of the transfer functions on the performance of the network. To possibly try identifying the preferences more clearly the networks could have been trained with a greater number of epochs. However, it is notable from Figure 27 that the performances of both networks are marginally better when the tan-sigmoid function is implemented in all three layers. This validates the predicted preference for the three layer setups. Therefore, the tan-sigmoid function was indicated as the preferred transfer function regardless of the number of layers within the NN and the volume of the neurons within each layer.

4.1.2 Select Training Function

In conducting this selection the most favourable transfer function configurations were selected for the three alternate hidden layer setups. As identified in the previous section the tan-sigmoid function was the most favourable and was therefore implemented in conducting this next step in the ANN development. A similar process to the identification of the transfer function was carried out in the selection of the most optimal training function. A total of 100 epochs was again deemed a sufficient training limit for assessing the performances. Once more an assessment was desired on variations to neuron volume for each hidden layer setups. This was to provide a little further scope and assess the preferential influence of neuron volumes in the layers. Subsequently, the previous one, two and three hidden layer configurations were again utilised to compare performances achieved with different training functions.

Single Hidden Layer Networks

Each of the identified training functions within MATLAB were implemented and assessed initially within the 3-[25]-2 Network. The comparison of the performances, presented in Figure 28 identify the *traingdm* and *traingd* functions being clearly out performed by the various other functions. The performances obtained from the second network trained with the remainder of these various function are presented and compared with the first network in Figure 29.

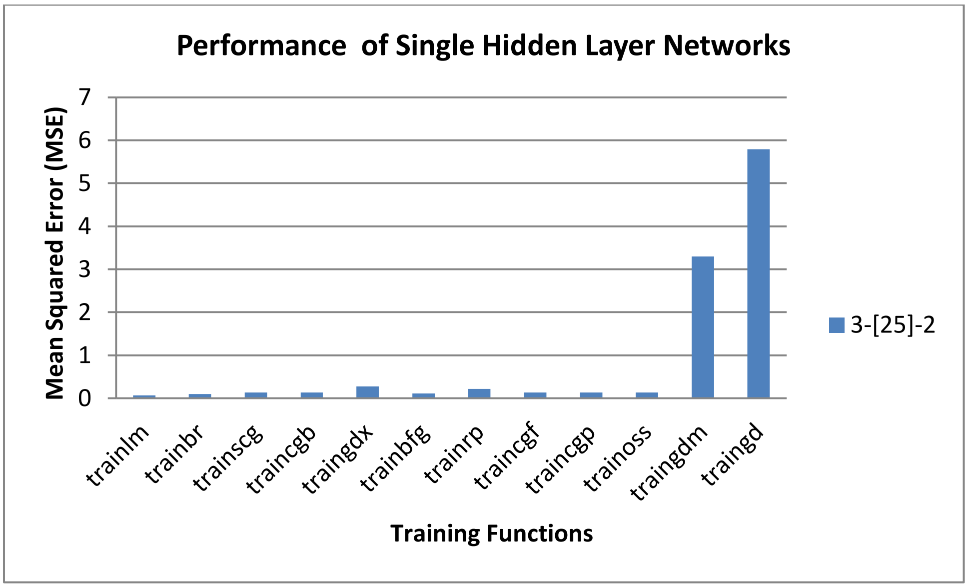

Figure 28 – Performance comparison of training functions in a 3-[25]-2 network

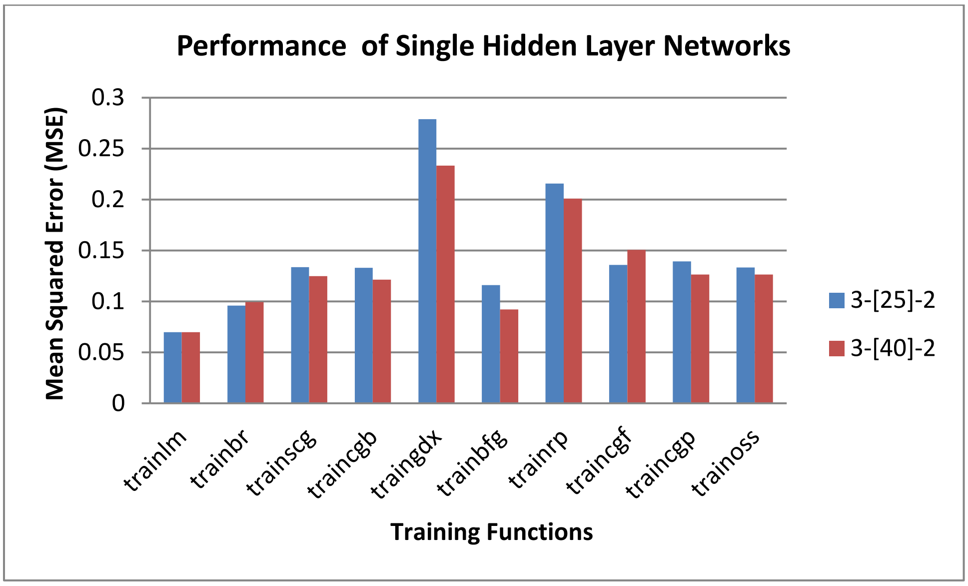

Figure 29 – Performance comparison of training functions in single hidden layer networks

The general trend in the performance from each of the tested functions is relatively similar between the two networks. Thus, the neural volume was identified to have minimal influence on the sequence of the training function preferences. A gain there is a general trend of improved performance with increase in nodes within the hidden layer. Based on these performances the *trainlm* function clearly offers the greatest performance with the least MSE obtained.

Double Hidden Layer Networks

Only a select few of the training functions were considered for the double layer setup based on the performances of the single layer system. The training functions that identifiably produced greater MSE values for the single layer systems were neglected. Figure 30 compares the remainder of the functions used to train the two double hidden layer networks. The three most effective training functions were recognised as *trainlm, trainbr* and *trainbfg*. This was also the case when comparing the results for the single layer networks. Once again the general sequence of preferences between the training functions is similar to that of the single hidden layer systems. This identified that there was no change in preference between numbers of hidden layers employed. Thus, the most optimal training function was again the *trainlm* function.

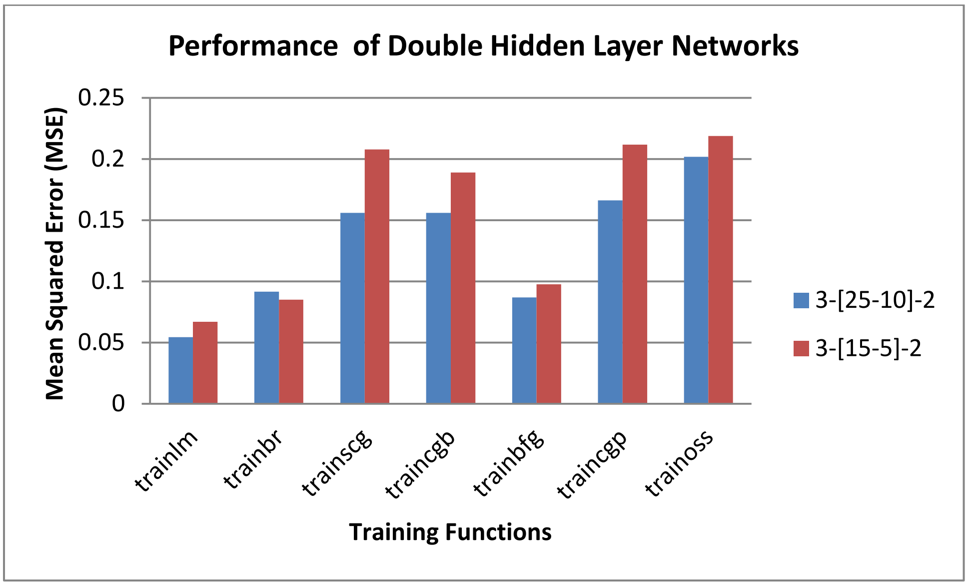

Figure 30 – Performance comparison of training functions in double hidden layer networks

The similarity in performance trends once more highlights and validates the minimal influence of the node volume on the sequence of the training function preferences. The general trend of improved performance with increase in nodes within the hidden layer was also proven for the double layer systems.

Triple Hidden Layer Networks

In a similar manner the two three layer networks were trained and compared. Only the three optimal training functions identified in training the double and single hidden layer networks were assessed. This was conducted to validate that the number of layers and the neuron volumes would not change the preference of the training functions. This is confirmed in Figure 31 as the t*rainlm* function is once again indicated as the preferred training algorithm. The improvement of performance was also noted between the two setups. The greater neural volume within the layers promoted a reduction in the MSE as predicted from the analysis of the single and double layer systems.

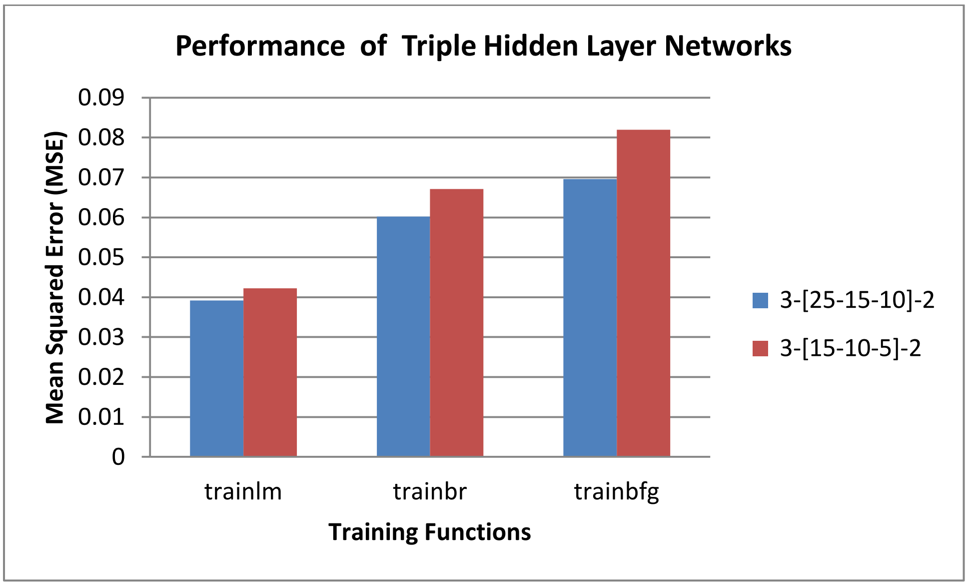

Figure 31 – Performance comparison of training functions in triple hidden layer networks

4.1.3 Select Layer Configuration

The remaining stage of developing the structure required the systematic comparison of network node and layer structures. Since the most optimal use of transfer and training functions had been established direct comparisons were then able to be made regarding the preference of the number of layers within the NN. Some basic trends regarding the preferences of neuron volume were already identified from the comparisons of the two previous development steps. There was a clear sign that the training performances of all network setups would improve as the density of node within the hidden layers increased. It is recognised from previous texts that the increase in performance would eventually saturate at a particular density of neurons (*Yousif & El-Tayeb 2008, Jiang et al. 2007*). However, before the neural volume could be established the number of layers within the networks structure was identified first.

Number of Layers

The optimal number of layers for the network was identifiable from all previous training results during the selection of the transfer and training functions. This was achievable due to the training and assessment of the three different layer configurations. As previously established, only between one to three hidden layers were considered in the analysis as this was deemed sufficient to handle the complexity of the input to output relationships. Therefore the six previously trained neural network structures with optimal transfer and training function setups were directly compared and presented in Figure 32. For this comparison the networks were trained up to 300 epochs. The increase in training epochs was in the hope to identify greater differences between the network performances and present a better view of their training potential.

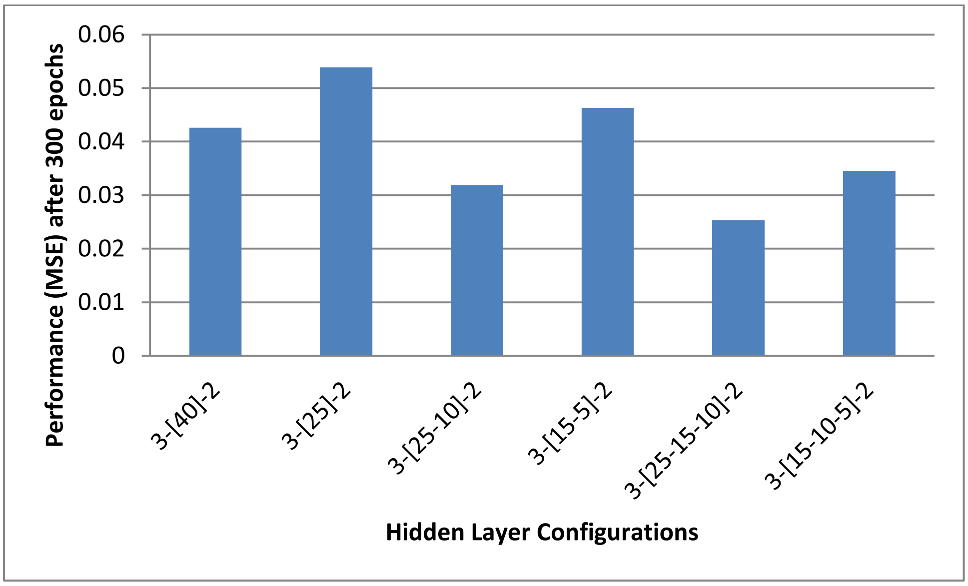

Figure 32 – Performance comparison hidden layer configurations

The performance values presented in the figure identifies two clear trends. The first has already been identified in the previous development steps in regards to the neuron volume. Regardless of the number of layers the performance is improved with greater node volume. The second trend is the improvement of performance with the addition of each hidden layer to the network structure. Therefore Figure 32 suggested that, for the given dataset, the most optimal training performance is achieved by implementing three hidden layers.

Neuron Volume

To establish an optimal neuron volume in each hidden layer of the NN, two three layer configurations previously trained were utilised as the base models. Neuron variations to these models were trained and compared. Each of the network configurations for each series of layer tests were trained to 100, 200 and finally 300 epochs by performing successive training sessions. This was conducted so that the performance at each interval could be compared. Presenting these points within a fitted line graph also indicates the general convergence of each neural configuration.

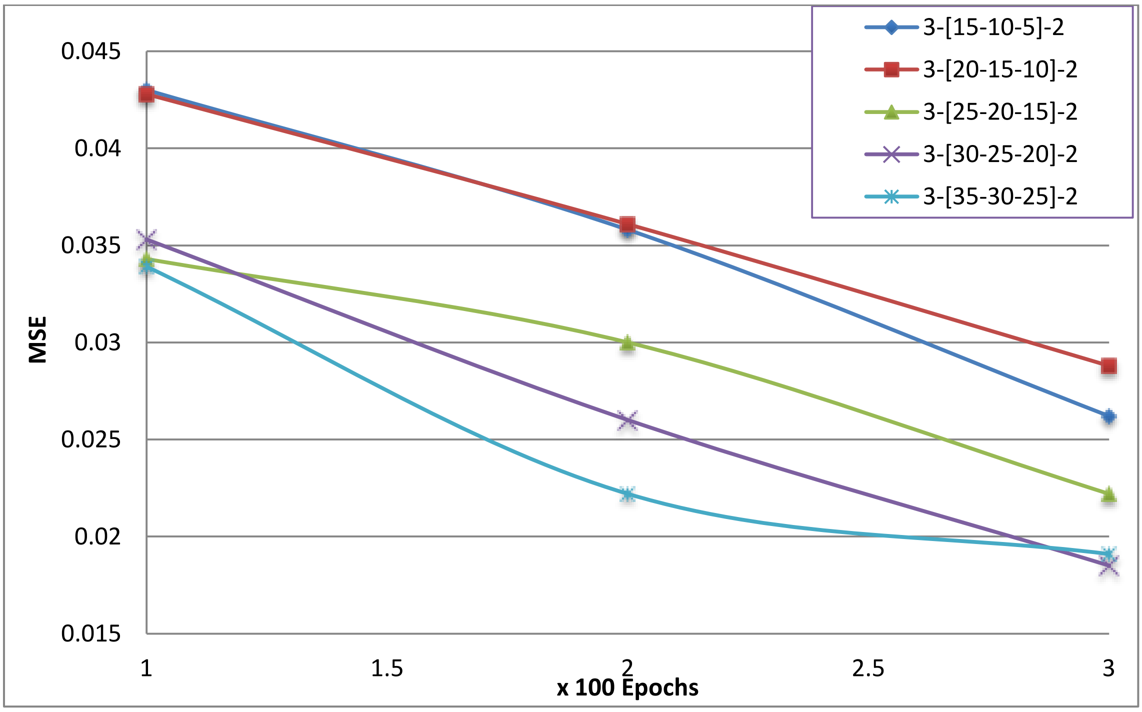

Figure 33 –Performance of various node volumes

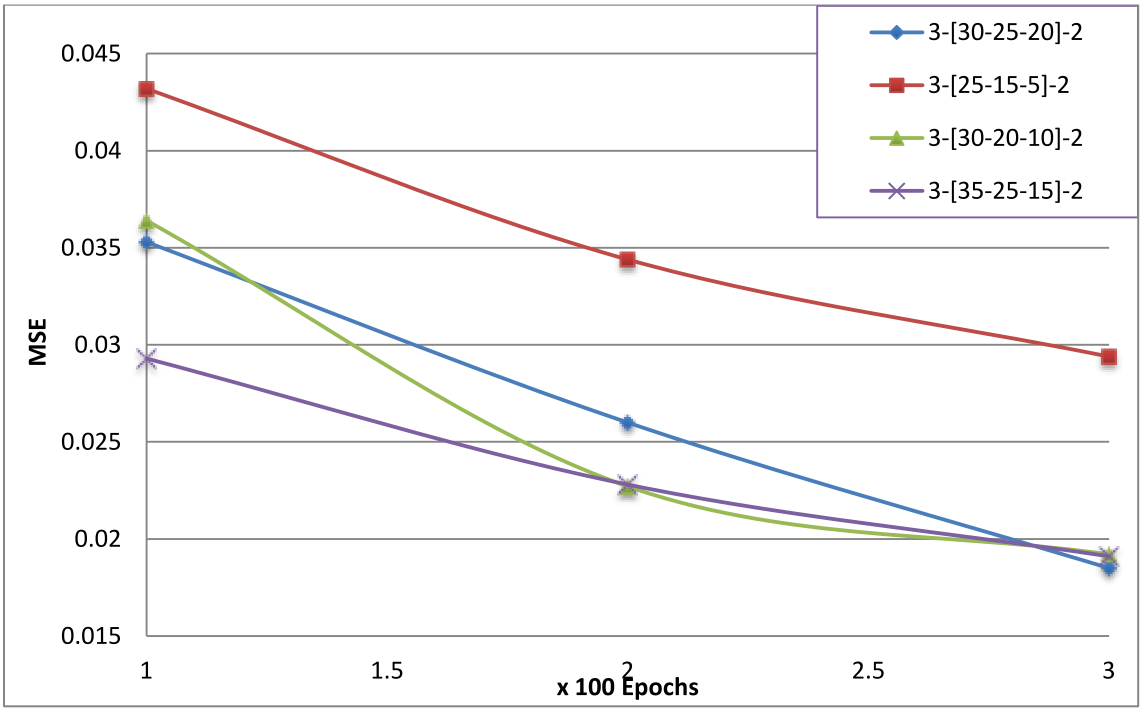

Figure 34 - Performance of various node volumes

The two graphs presented depict the general trend of the MSE as each neural configuration is tested (Figure 33 & 34). It was noted that at this stage the relative improvement in performance between the various configurations would be marginally small. From the above comparisons it was deduced that the 3-[30-25-20]-2 configuration would be one of the more desirable configuration setups for continued training. This is emphasised by a less diminished improvement in performance as the training is continued. This represents a larger gradient of performance convergence. The relatively better performance values for the presented epoch training range also distinguishes the configuration as the most preferred.

4.1.4 Training and Testing without Generalization

Continued training of the ANN has been highlighted to continue improvements in the performance (Yousif & El-Tayeb 2008, Jiang et al 2007). The model accumulates an improved performance as it repetitively adjusts itself for greater accuracy from previous training cycles. Nasir and Yousif (2009) trained a selected optimal model with a *trainlm* function for a total of 3000 epochs. They observed that the system gradually converged to a point where no continued performance improvement was achieved. Employing the same approach, the selected 3-[30-25-20]-2 network configuration was trained up to 2001 epochs with the *trainlm* function. Figure 33 presents the training graph that indicates an initially quick reduction in the MSE. The performance gradually reached a saturation point of 1.28x10^{-5} where no further significant improvement was observed.

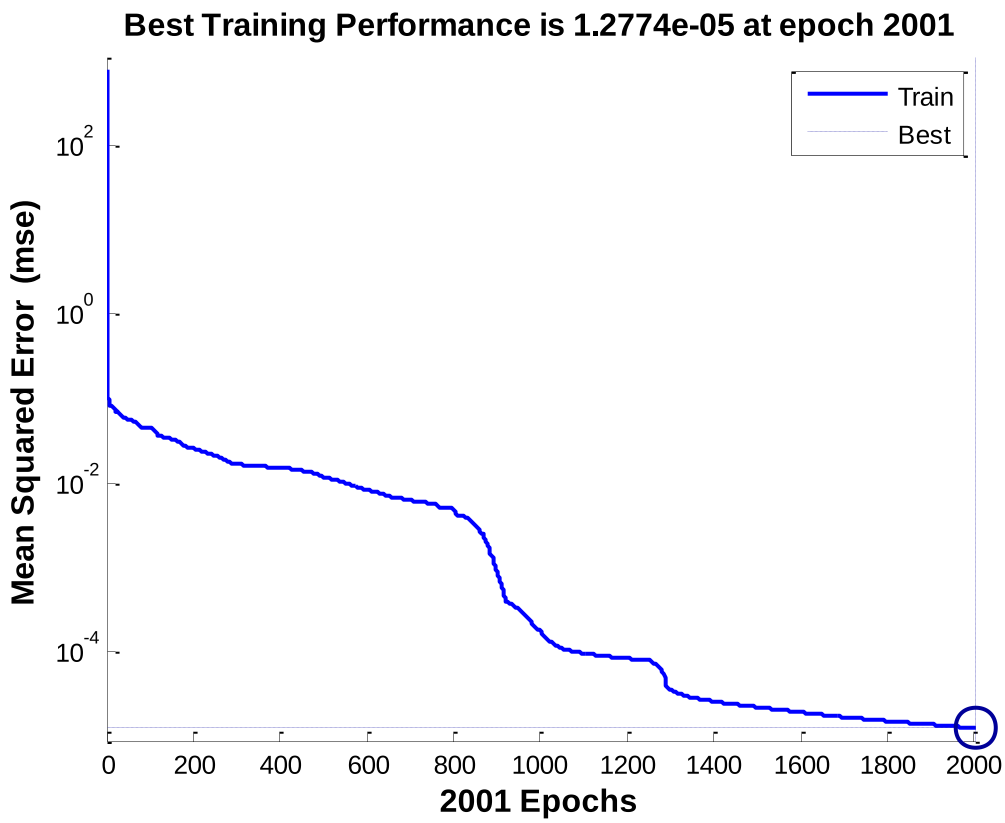

Figure 35 – Selected ANN model training with *trainlm* over 2001 epochs

The final training performance value was considerably small and indicates an exceptionally good fit. The predicted results of the ANN would therefore be expected to exactly reflect the experimental training output data. This perfect fit of the data is clearly evident within the comparison plots presented as Figure 36 and 37. These figures plot the friction coefficient against the sliding distance and give a general sense of the predictability of the network for the data implemented in its training. The standard deviation (SD) of the network outputs from the experimental data is effectively zero as there is clearly no single point predicted by the ANN that differs from the experimental data. However, this presented a considerable issue.

The apparent problem is that over fitting has occurred during the NN training. This evidently occurs when the training data set predictions have been driven to very small error values. In this case the network essentially memorizes the training set, and has not learned to generalize to new conditions (*Demuth and Beale 2013*). Hence there will typically be large errors when unseen data is presented to the network. Therefore the trained network will be ineffective at interpolating new data points, which is the emphasis in developing the NN model.

Comparison of Experimental and ANN Results without Generalizing

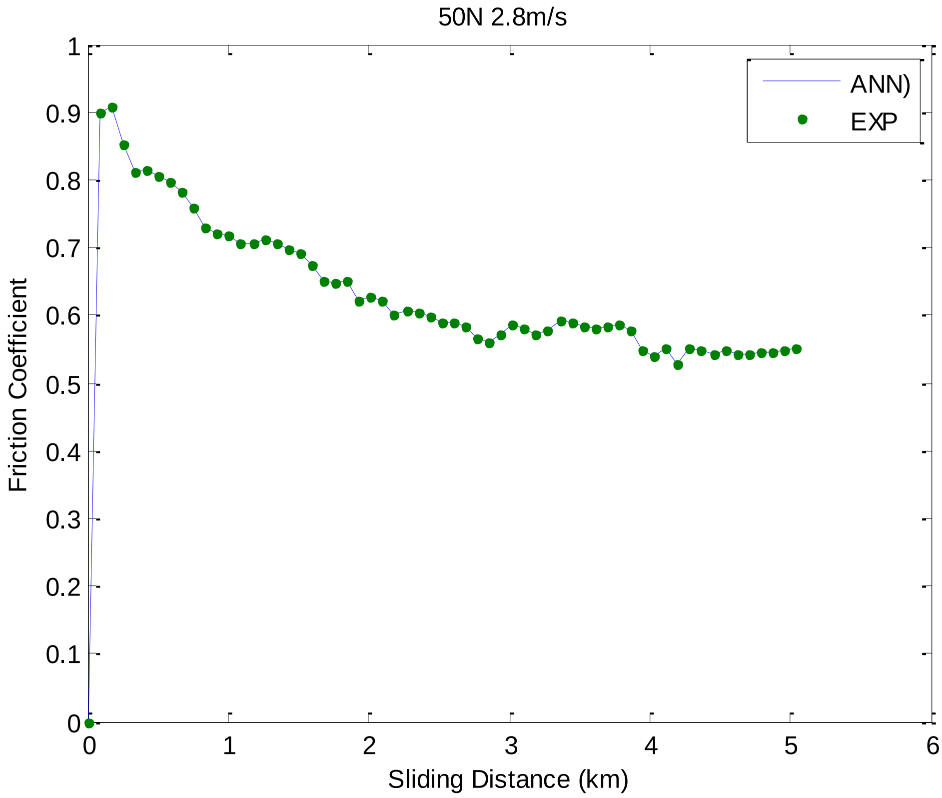

Figure 36 – Friction coefficient results from ANN predictions and Experimental training data at 2.8m/s with 50N force

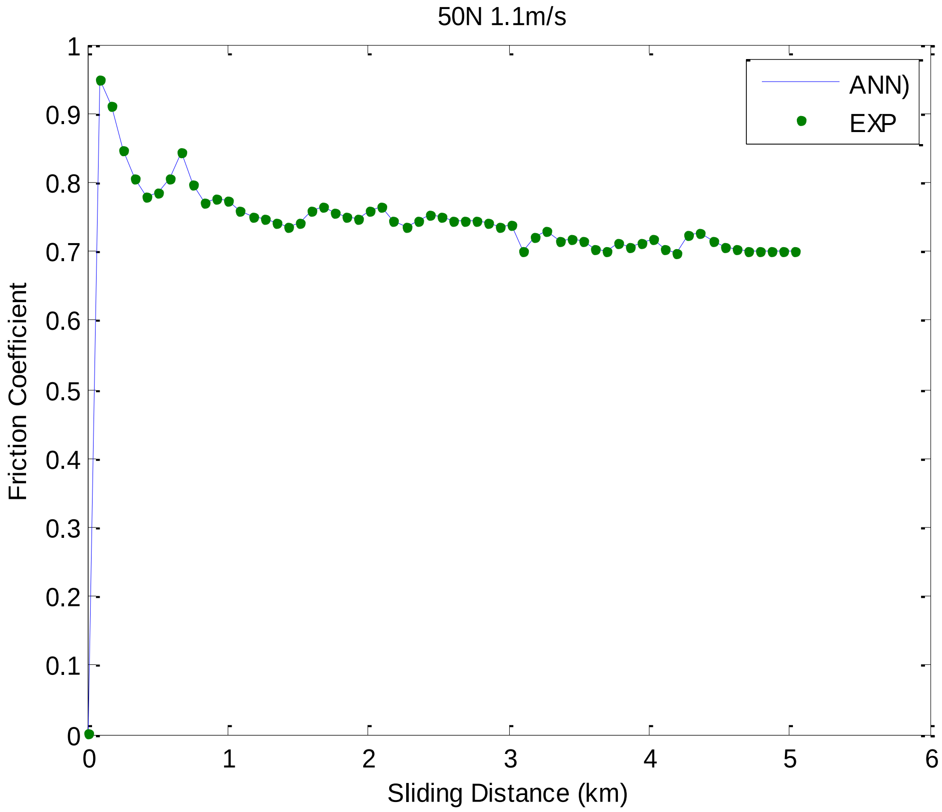

Figure 37 – Friction coefficient results from ANN predictions and Experimental training data at 1.1m/s with 50N force

4.2 Generalizing

4.2.1 Training with Generalization

It is well distinguished that multilayer networks that have been trained properly are capable of producing reasonable answers for inputs that they have not seen within their training. This predictive ability of the network is referred to as generalisation. The property enables the network to be trained with only a set of representative paired inputs and targets. The NNs are therefore capable of obtaining good results without implementing all possible input and output pairs in the training.

4.2.2 Generalising Technique

Early Stopping

As identified within the reviewed literature it was recognised that utilising smaller networks or larger data sets would improve generalisation. Subsequently, in conducting an initial assessment upon training with early stopping three networks were trained. These are three of the six networks used in the previous training conducted. Each varied in the number of hidden layers to once again recognise any trends in the preferred number of hidden layers. All the optimal generated MSE and R values for three repeated tests for each network were tabulated and are presented within Appendix D. Presented in Figure 38 are the resulting training MSE averages. Clear indications from this figure are the decreasing MSE and improved performance as the number of hidden layers is reduced. This is obviously an influence caused by the early stopping as the reverse trends were observed in the non generalizing training methods in previous sections.

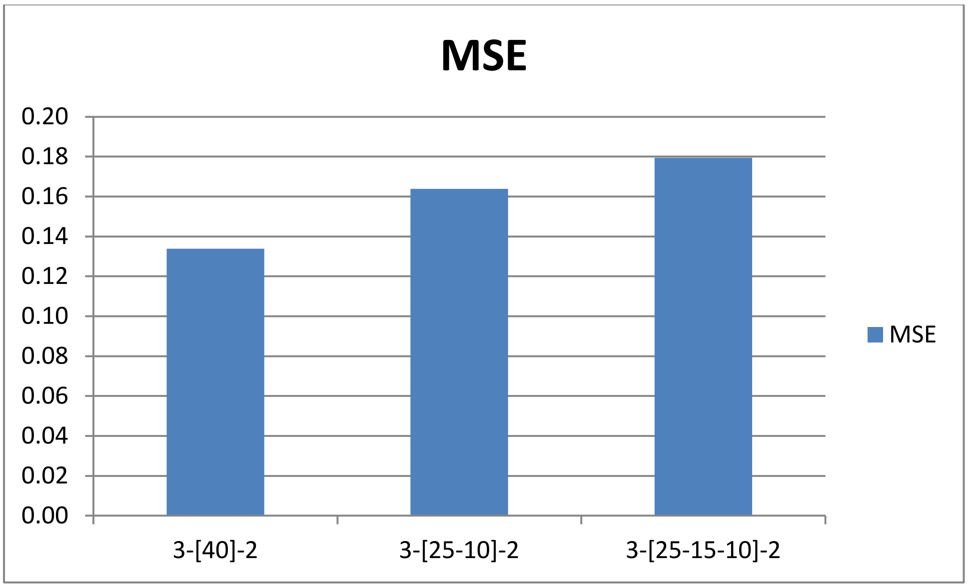

Figure 38 - Average achieved training MSE values for variant hidden layer networks implementing early stopping generalisation, trained with *trainlm*.

Averaged R values or correlation coefficients of the predictions from the trained ANN models are summarised in Figure 39. R values pertain to the ANN predictions for friction coefficients. This is to provide further insight into the training and generalised predictability of the networks. The graph presents the regression values for the three divisions of the data set of each network. The data sets were divided into 70, 20 and 10 percent for the training, validation and testing sets, respectively. The test set error is not used in the training. It does serve as a useful comparison the different models. It often also useful to plot the MSE for the test set during the training process. The R values plotted highlight the single layered networks as having greater correlation of its predictions to the training and validation sets. However, the average R value presented for test set is significantly lower for this network. This may be a resulting influence of the way the data sets were randomly divided.

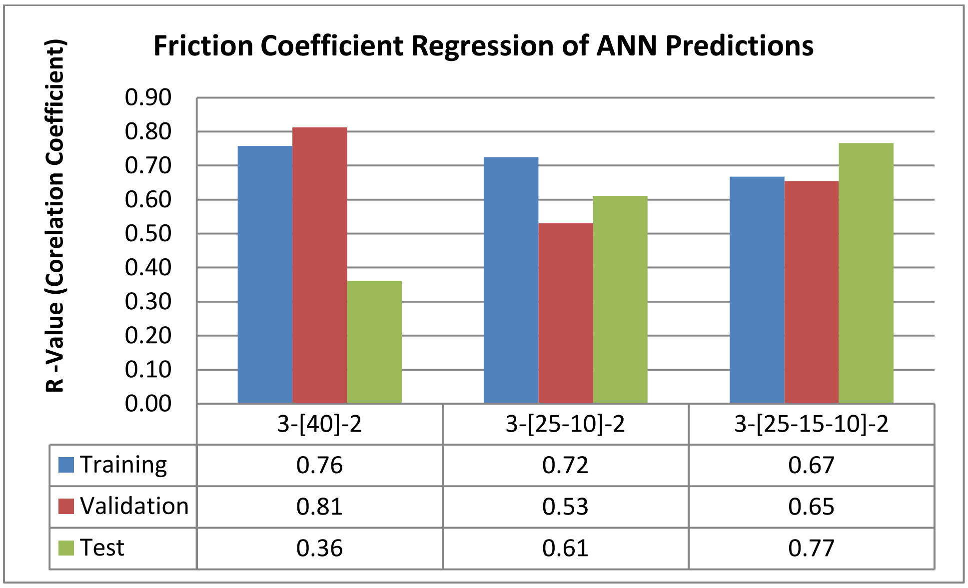

Figure 39 - Average achieved R values for variant hidden layer networks implementing early stopping generalisation, trained with *trainlm*.

It must also be noted that all the sub-sets of data have R values that are closely approaching unity for all networks. Indicating exact correlations for contact surface temperatures. It is apparent from the data that there is a close to linear relationship with the sliding distance (Refer to Appendix A). Hence, most of the comparative analysis is being performed based on the friction coefficient as this clearly involves more complex relationships.

The R values for the validation and test sets give some sense of the networks ability to generalise since the data contained within these sets are unseen by the network during its training. The combined average of these two sets may provide a more effective comparison of the networks generalizing ability. In that sense the three layered network would have the best generalizing result but still the lowest correlation to the training data.

All R-values fall short of the standard satisfactory 0.9 value. It was quite clear from these presented R-values that this generalisation method was incapable of producing trained networks that could correlate outputs and targets satisfactorily.

Bayesian Regularization

It is highlighted by Demuth and Beale (*2013*) that the Bayesian regularization method achieves superior generalization performance than early stopping method when training function fitting networks with relatively small data sets. This is mostly associated to the fact that a validation data set is not required to be separate from the training set. Hence the network utilises more data during the training.

The six networks used in the previous conduct training were trained to again monitor any trends in the preferred number of hidden layers. All NN configurations were trained using the *trainbr* function and all MSE and R values for three repeated tests for each network were tabulated and are presented within Appendix D.

Thus training was halted after 300 epochs. Presented in Figure 40 are the resulting training MSE averages. Clear indications from this figure are the lower MSE and improved performance achieved by the larger neural volumed double layered network.

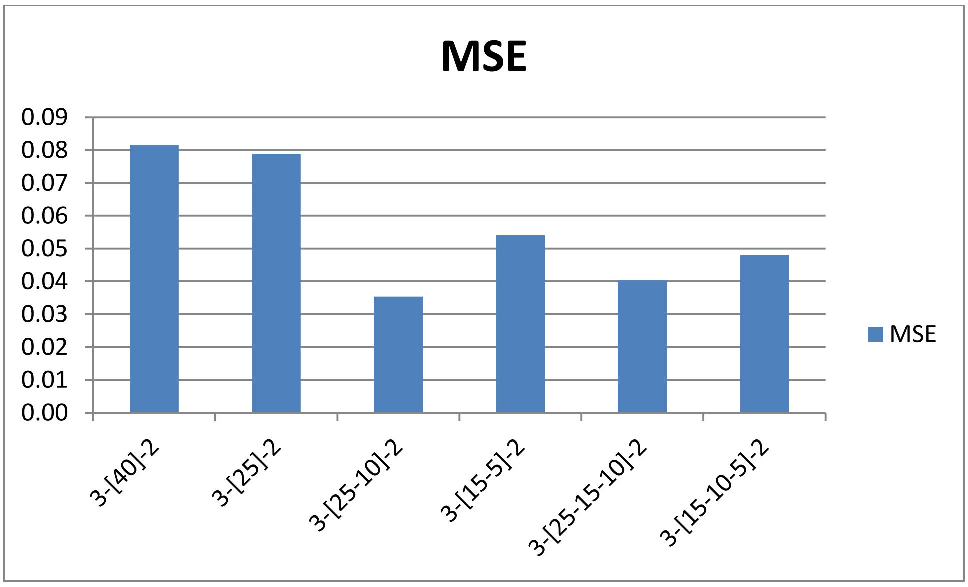

Figure 40 - Average achieved training MSE values for variant hidden layer networks and volumes trained with *trainbr*.

Averaged R values or correlation coefficients of the predictions from the trained ANN models are summarised in Figure 41. These R values again pertain to the ANN predictions for friction coefficients. The graph presents the regression values for the two data set divisions of each network. The data sets were divided into 90 and 10 percent for the training and testing sets, respectively. The R values plotted highlight the 3-[25-10]-2 configured double layer network as having greater correlation of its predictions to the training and test sets.

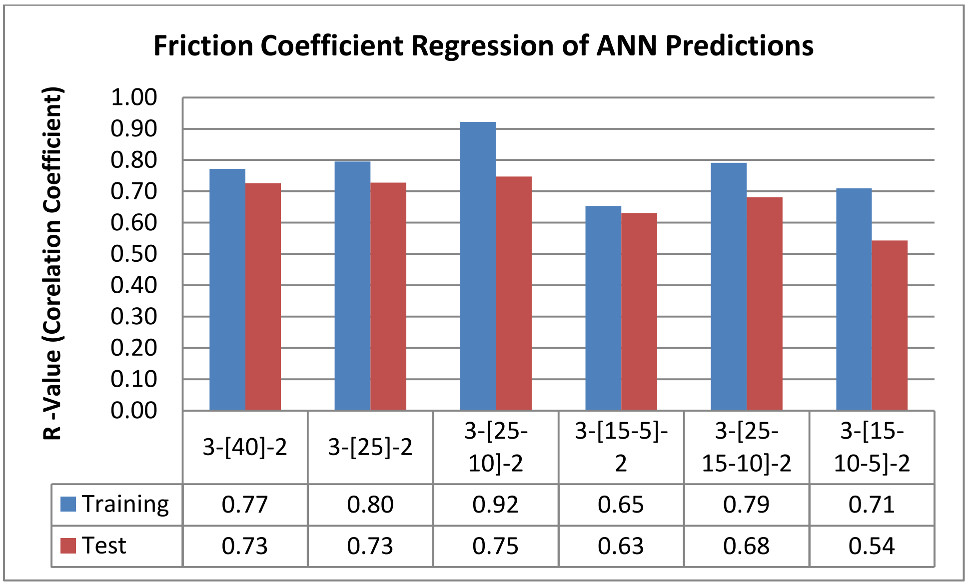

Figure 41 - Average achieved R values for variant hidden layer networks and volumes trained with *trainbr*.

4.2.3 Train and Test Generalised Model

The 3-[25-10]-2 network model configuration trained with the Bayesian Regularization function was clearly identified as the optimal performing model setup in the previous analysis sections. Thus, as performed before with the ungeneralised case further training was carried out on with the selected model. This was achieved by expanding the training session epoch limit. Demuth and Beale (*2013*) highlight that it is desirable to run the regularization algorithm until the effective number of parameters had converged.

It was also recognised that a relatively constant MSE for several epochs also indicates convergence. Thus the training was stopped when the training MSE gradient reduced below a set minimum of 0.001. This was established to be optimal as continued training lead gradually to over fitting as the test MSE would begin to steadily rise. This was observable in the generated MSE performance plot. The training session was reinitialised several times, testing several different initial conditions and data subset divisions to try and verify robust network performance. The final performance plot for the best training session encountered is presented as Figure 42.

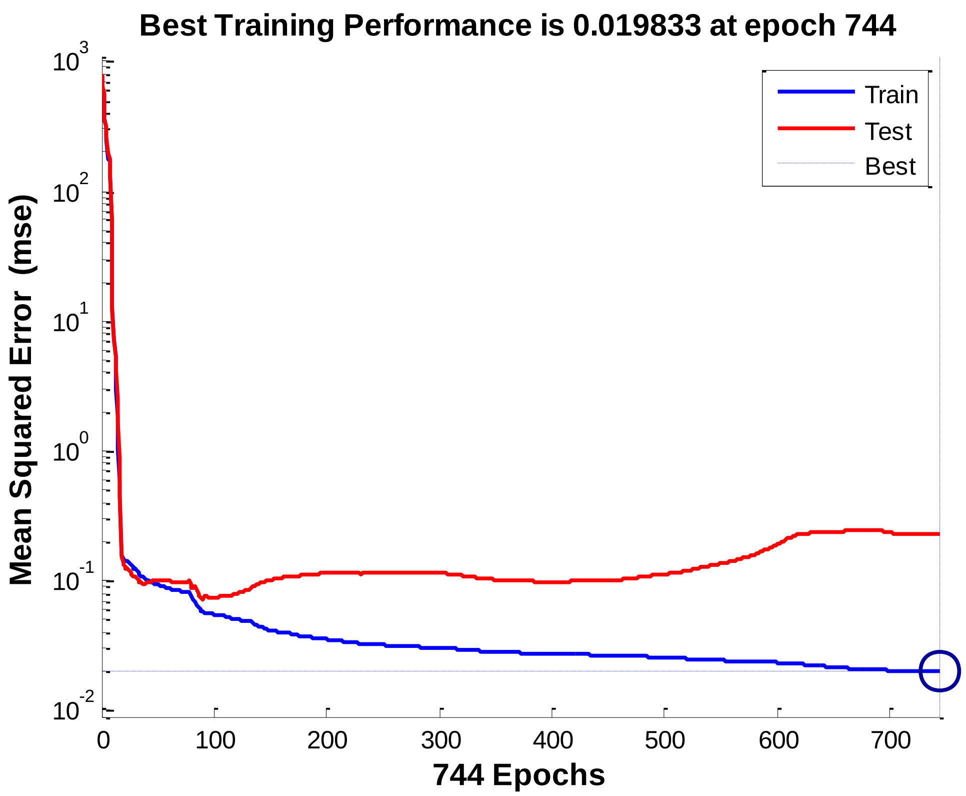

Figure 42 – MSE Performance plot for double hidden layer network 3-[25-10]-2 trained with automatically generalising *trainbr* training function.

The regression plots depicted in Figure 43 of the training and test data sets indicate that the relative trend between the ANN outputs and the real data targets have an almost exceptional correlation. This correlation is expressed by the R value presented at the top of each plot. It is also apparent by the regression lines close alignment to the unity regression line (dashed line). These values clearly well exceed the standard minimum recommended R values of 0.9, as defined by Nasir et al. (*2009*). The network therefore is expected to have good generalised predictability and should be capable of very closely interpolating any parameter inputs.

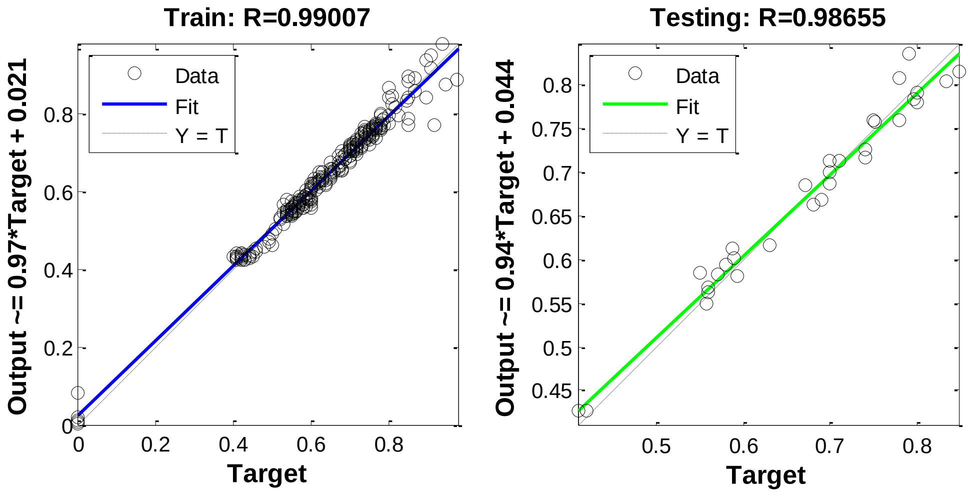

Figure 43 – Final trained optimal model regression plots.

4.3 Simulate and Compare ANN Results

Simulation output values for the ANN for a given set of parameters are plotted alongside the associated target values in some of the following Figures. The key note deduced from these plots is that the R or correlation values presented from Figure 43 are validated by the ANNs close approximation to the targets for each of the sets of parameters compared in these plots.

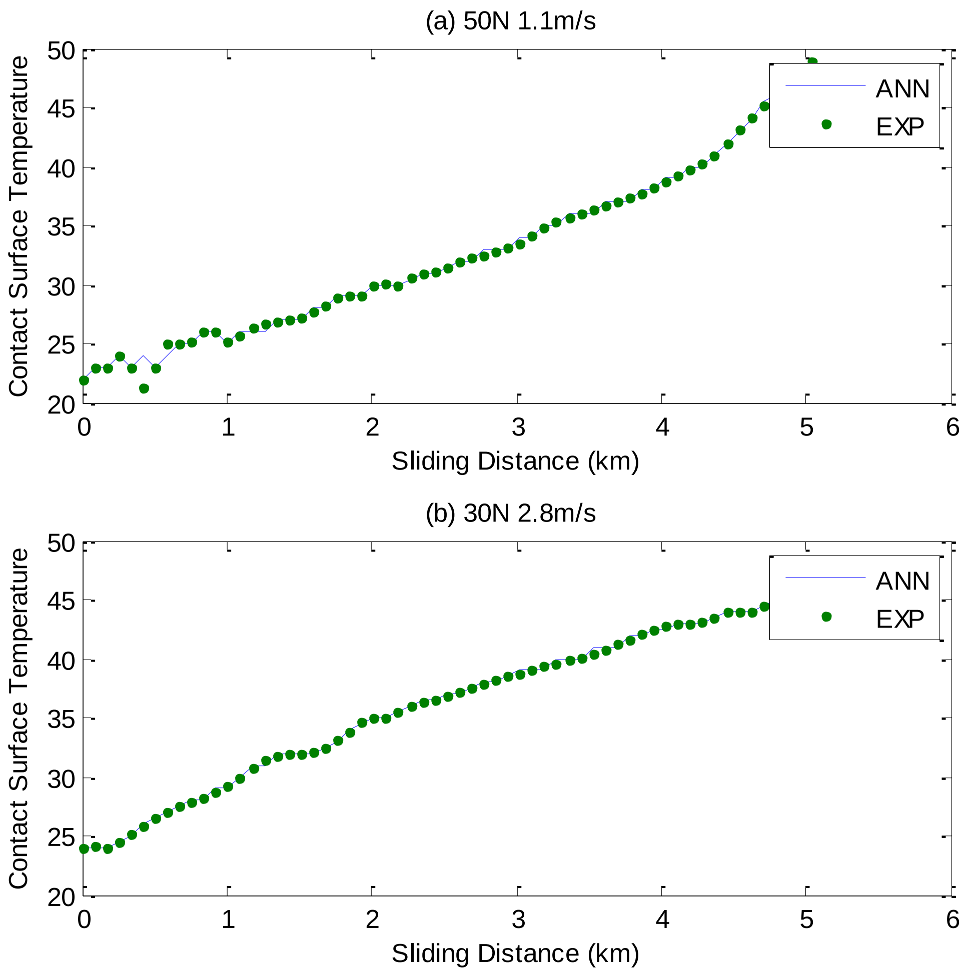

Figure 44-ANN predictions and experimental data for surface temperature for various sliding distances

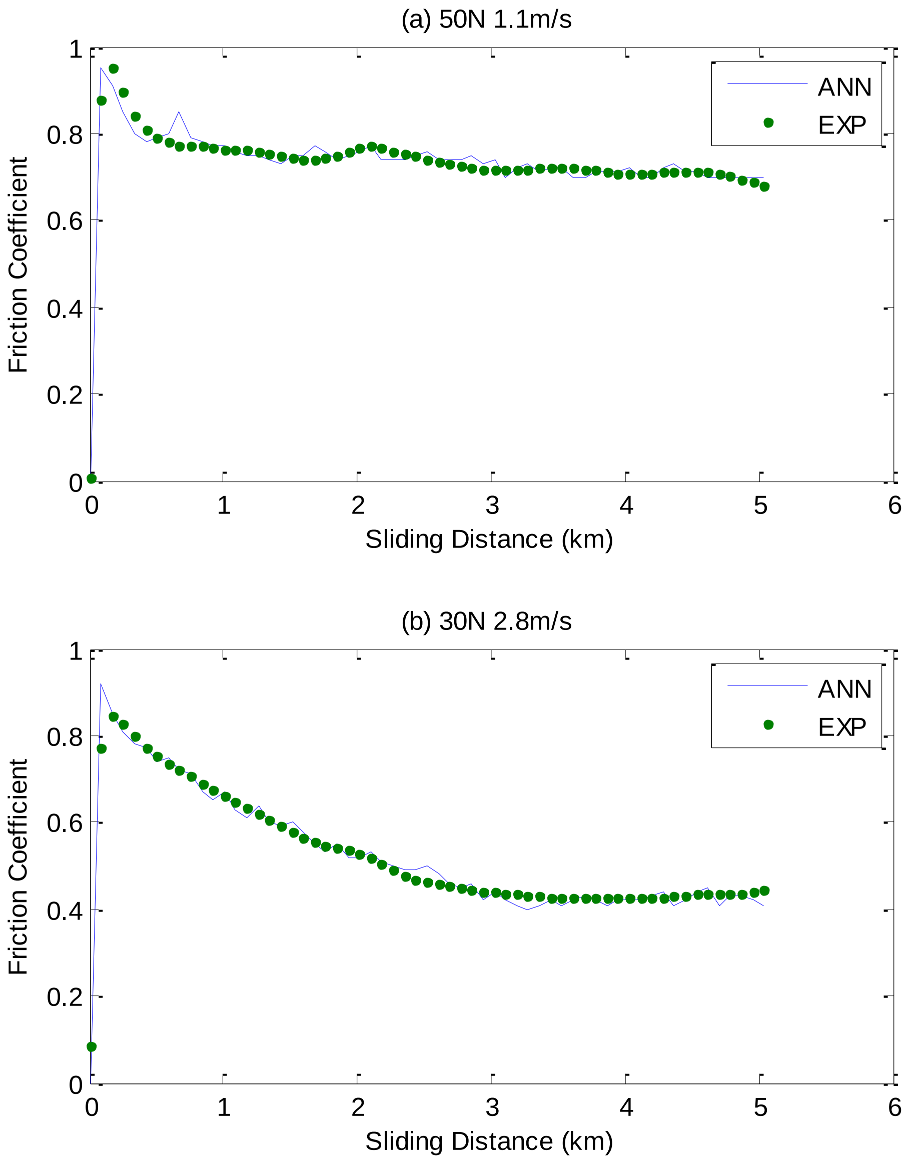

Figure 45- ANN predictions and experimental data for friction coefficient for various sliding distances

4.3.1 Predictability outside Trained Domain

The R value for the test data set indicated that the ANN model can closely interpolate or generalise for unseen data points quite well. Additional simulations are also performed for some data points that exist outside of the domain of the of the training and test data sets. These simulations of the network for points outside the domain of the total data set indicated that the networks ability to extrapolate would be less dependable the further the data points are form the trained domain. Figures 46 and 47 depict at least three data point outside the domain which appears to continue following the general trend of the experimental data trend lines. These points however cannot be validated for accuracy by the current supplied data sets. However, it can be assumed that there is a relatively small reduction in the accuracy of prediction; given the predicted data point input parameters exist within a small percentage from the inputs from the trained domain.

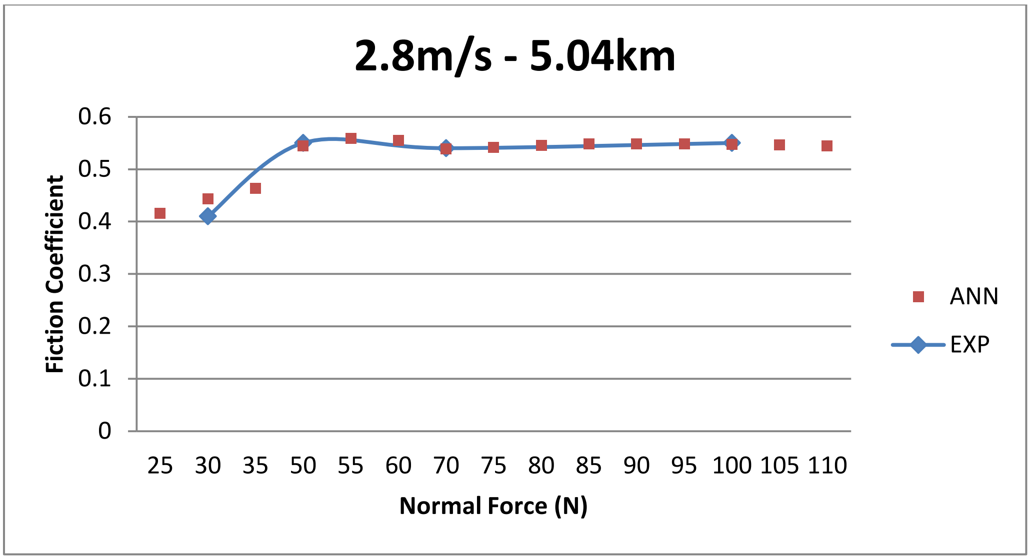

Figure 46 - ANN predictions and experimental data for friction coefficient for various load force

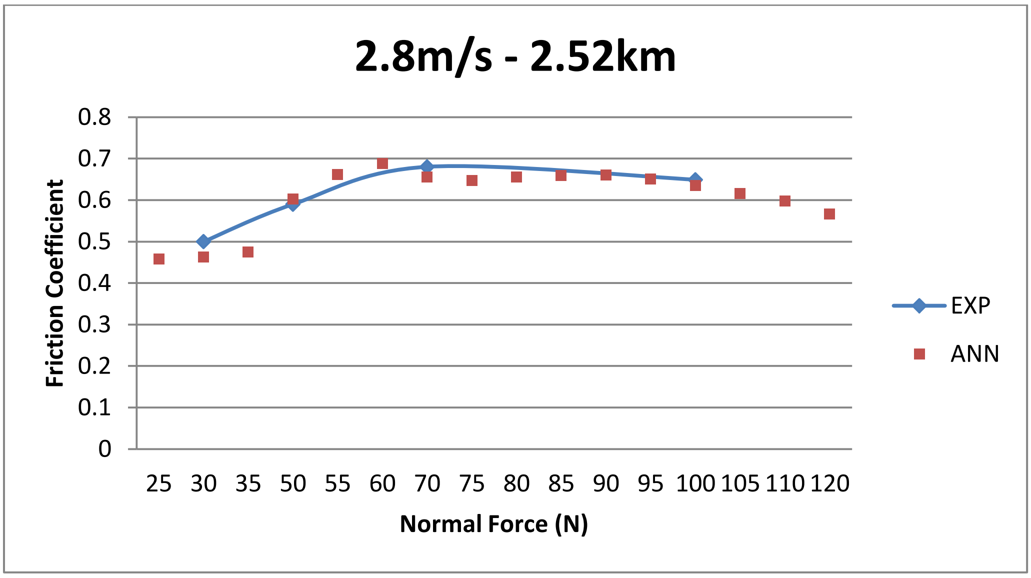

Figure 47 - ANN predictions and experimental data for friction coefficient for various load force.

This reduction in the network ability to extrapolate for points further from the trained domain is illustrated by the comparison plot presented in Figure 48. Within the figure is a trend line for the experimental results for the sliding speed of 2.8m/s plotted alongside the ANN predicted points for the variable of 1.1m/s. The single point within the data set for these input parameters at 1.1m/s also plotted. As expected good correlation to the ANN prediction exist for this point. The expected trend for the ANN points would be to closely follow the trend presented by the experimental data at the elevated speed with a slight offset. However, as the data points move away from only experimental point at that speed, the ANN predictions continue to leave the offset trend of the faster experimental points. These predicted points are therefore less dependable the further the data points are form the trained domain.

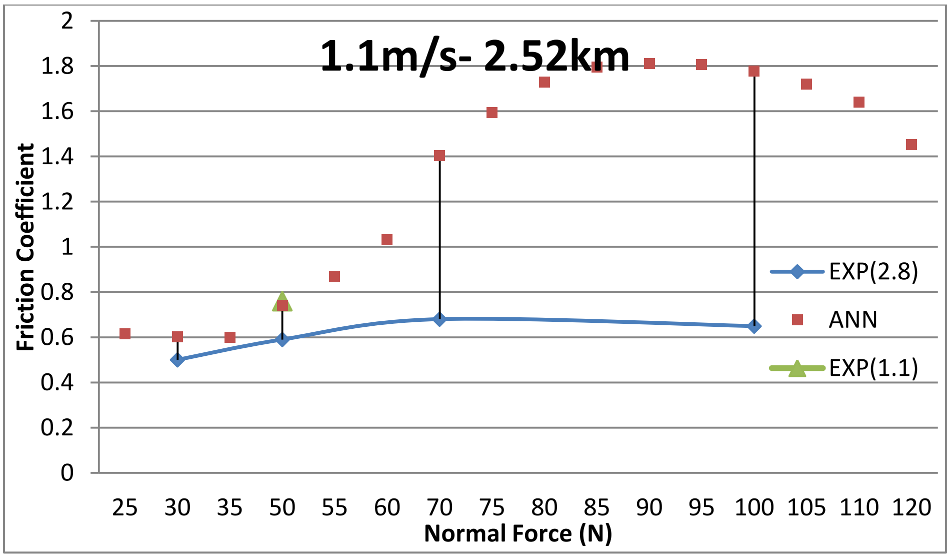

Figure 48 - ANN predictions and experimental data for friction coefficient for various load force

5 Conclusions

5.1 Introduction

This project has investigated and developed an artificial neural network (ANN) that effectively predicts tribological characteristics of kenaf fibre reinforced epoxy composite (KFRE). Through this project an understanding of the benefits of developing and implementing ANNs has been gained.

To develop the network sufficient previously collected data was processed and implemented to establish an optimal ANN model through testing various neural, layer and function configurations. An optimal ANN model was derived with additional consideration to achieving improved network generalisation. The developed optimal generalising network was trained and tested. Further simulations were carried out with the derived network to confirm accuracy of model and also assess ability to extrapolate.

5.2 Derived Model Configuration and Training

The selection of derived models optimal architectural configuration and training set up involved a systematic trial and error approach. Numerous possible transfer function, training function and network configurations were trained, compared and critiqued systematically. Comparison and optimal selection was performed based on the mean square error of ANN outputs from the actual target data.

The systematic analysis established the optimal network configuration as double hidden layer network with 25 and 10 neurons in the first and second layers, respectively. The recognised optimal transfer function for the network utilises the tan sigmoid transfer function within the hidden layers. Training with the Bayesian regularisation training function was determined to provide the most optimal generalised performance from this network. The final trained model achieved an MSE training value of 0.019833.

5.3 Network Testing, Simulation and Comparison

This network achieved output against target regression line R values of 0.99 and 0.9866 for training and testing data sets, respectively. The test data represents a random selection of 10% of the total data. The R value for the test data set indicated that the ANN model can closely interpolate or generalise for unseen data points quite well. Further simulations of the network for points outside the domain of the total data set indicated that the networks ability to extrapolate would be less dependable the further the data points are form the trained domain.

5.2 Conclusion

This study has verified the ability of an artificial neural network to make closely accurate generalised predictions within the given domain of the supplied training data. Improvements to the generalised predictability of the neural network was realised through the selection of an optimal network configuration and training method suited to the supplied training data set.

Hence, the trained network model can be utilised to catalogue the friction coefficient and surface temperature variables in relation to the sliding distance, speed and load parameters. This is limited to the domain of the training data. This will ultimately save time and money otherwise used in conducting further testing.

6 Recommendations

6.1 Introduction

The results and lessons from this project have established specific limitations to the use of the derived ANN model. From this further research and work recommendations were identified.

6.2 Limitations and Challenges

Throughout the project there were certain limitations and challenges encountered in generating the optimal model. These are listed below:

- Establishing and applying an optimal generalising training limit.
- Assessing and finding methods in the current literature to improve extrapolation.

6.3 Recommendations for future work

Questions pertaining to the desirable broader use and application of an ANN model throughout the project were recognised as possible recommend avenues of future work these are listed below:

1) Try integrating additional tribological data and expanding the domain of the produced neural network.
2) Try integrating additional input and output tribological parameters like fibre orientation and specific wear.
3) Analyse composite structure and assess its possibility for specific applications based on ANN predictions.

List of References

Aleksendric, D. and Duboka, ˇ C. 2006, ‘Prediction of automotive friction material characteristics using artificial neural networks-cold performance’, *Wear*, vol 261, issue3-4, pp. 269–282, viewed 10/04/2013, Science Direct, ELSEVIER

Albrecht Schmidt 2000, TECO, viewed 16/4/2013, <http://www.teco.edu/~albrecht/neuro/html/node7.html>

A. Patnaik Siddhartha, A.D. Bhatt 2011, ‘Mechanical and dry sliding wear characterization of epoxy-TiO_2 particulate filled functionally graded composites materials using Taguchi design of experiment’, *Material Design*, vol32, issue 2, pp. 615-627, viewed 15/04/2013, Science Direct, ELSEVIER

American Composite Manufactures Association, Viewed 20/03/2013, <www.mdacomposites.org>

Barnes 2012, *Neurons, Neurotransmission and Communication*, PDF file, Sage Publications, viewed 2/04/2013, <http://sagepub.com/upm-data/52317_Barnes, Essential_Biological_Psych_Chap_1.pdf>

Bunsell A.R. & Renard J, 2005, ‘*Fundamentals of fibre reinforced composite materials*’, CRC Press, USA

B.F. Yousif, U. Nirmal, K.J. Wong 2010, ‘Three-body abrasion on wear and frictional performance of treated betelnut fibre reinforced epoxy (T-BFRE) composite’, *Material Design*, vol 31, issue 9, pp. 4514–4521, viewed 15/04/2013, Science Direct, ELSEVIER

B.F. Yousif, N.S.M. El-Tayeb 2010, ‘Wet adhesive wear characteristics of untreated oil palm fibre-reinforced polyester and treated oil palm fibre-reinforced polyester composites using the pin-on-disc and block-on-ring techniques’ *Engineering Tribolergy Journal*, vol225 issue 2, pp. 123-131, viewed 15/04/2013, Science Direct, ELSEVIER

B.F. Yousif, N.S.M. El-Tayeb 2008, ‘Adhesive wear performance of T-OPRP and UT-OPRP composites’, *Tribol Lett*, vol 32, issue 3, pp. 199–208, viewed 15/04/2013, Science Direct, ELSEVIER, item: S0261306912004335

B.F Yousif , 2012, ‘Design of newly fabricated tribological machine for wear and frictional experiments under dry/wet condition’, *Materials & Design*, Vol 48, pp. 2-13, viewed 10/04/2013, Science Direct, ELSEVIER, item: S0261306912004335

Carnevale and Hines 2013, Neuron, Yale, Viewed 3/05/13, <http://neuron.yale.edu/neuron/what_is_neuron>

Çetinel, H., Öztürk, H., Çelik, E., and Karlık, B. 2006, ‘Artificial neural network-based prediction technique for wear loss quantities in Mo coatings’, *Wear*, vol 261, issue 10, pp. 1064–1068, viewed 10/04/2013, Science Direct, ELSEVIER

Demuth, H. and Beale, M. 2013, ‘*Neural Network Toolbox -User guide*’, The MathWorks Inc., PDF, <http://www.mathworks.com.au/help/pdf_doc/nnet/nnet_ug.pdf>

Engineering abc.com, viewed 25/04/2013, <http://www.tribology-abc.com/>

Fausett 1994, *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*, Prentice-Hall, New Jersey, USA

Gyurova &Friedrich 2010, 'Artificial neural networks for predicting sliding friction and wear properties of polyphenylene sulfide composite', *Tribology International*, Volume 44, Issue 5, pp. 603–609, viewed 8/04/2013, Science Direct, ELSEVIER, item: S0301679X1100003X

Haykin 1999, *Neural networks: a comprehensive foundation*, 2nd ed, Prentice Hall, New Jersey

Jie, X. L.,Davim, J. P., and Cardoso, R. 2007, 'Prediction on tribological behaviour of composite PEEK-CF30 using artificial neural networks', *Material Processes and Technology*, vol 189, issue 1-3, pp. 374–378, viewed 10/04/2013, Science Direct, ELSEVIER

Jiang, Z.,Zhang, Z., and Friedrich, K. 2007, 'Prediction on wear properties of polymer composites with artificial neural networks', *Composite Science and Technology*, vol 67, issue 2, pp. 168–176, viewed 10/04/2013, Science Direct, ELSEVIER

Kaw, K.A 1997, Mechanics of composite materials, CRC Press, USA

Lau 1994, *Neural Networks: Theoretical foundations and analysis,* IEEE Press, New York

Lippmann 1987, *An Introduction into computing with Neural Nets,* IEEE ASSP Magazine, pp.4-22

Lisboa 1992, *Neural Networks: Current applications*, Chapmen and Hall, Department of Electrical engineering and electronics University of Liverpool, Liverpool, UK

Myshkin, Kwon, Grigoriev, Ahn and Kong 1997, 'Classification of wear debris using neural network', *Wear,* Vol 204 (1997), pp. 658–662, viewed 8/04/2013, Science Direct, ELSEVIER, item: S0043164896074327

M. Jawaid, H.P.S. Abdul Khalil, A. Abu Bakar 2011, 'Woven hybrid composites: tensile and flexural properties of oil palm-woven jute fibres based epoxy composites', *Material Science Engineering,* vol 528, issue 15, pp. 5190-5195, viewed 15/04/2013, Science Direct, ELSEVIER

Mano, E B, 1991, '*Polímeros como Material de Engenharia'*, Editora Edgar Blücher Ltda, São Paulo

Matthews F. L & Rawlings Rees D, 1999, '*Composite materials: engineering and science*', Woodhead, UK

Nirmal 2010, 'Prediction of friction coefficient of treated betelnut fibre reinforced polyester (T-BFRP) composite using artificial neural networks', *Tribology*

International, Volume 43, Issue 8,pp. 1417-1429, viewed 27/03/2013, Science Direct, ELSEVIER, item: S0301679X10000277

Neural Ware 2007, Neural Ware, viewed 20/4/2013, <http://neuralpower.com/technology.htm>

Neelakanta & Groff 1994, *Statistical Mechanics and Cybernetics Perspectives,* CRC Press, Florida Atlantic University, Florida

N.S.M El-Tayeb, B.F. Yousif, T.C. Yap 2008, 'An investigation on worn surfaces of chopped glass fibre reinforced polyester through SEM observations', *Tribology International*, vol 41, issue 5, pp. 331-340, viewed 15/04/2013, Science Direct, ELSEVIER

Pai, P. S., Mathew, M. T., Stack, M. M., and Rocha,
L. A. 2008, 'Some thoughts on neural network modelling of microabrasion–corrosion processes', *Tribology Inernational*, vol 41, issue 7, pp. 672–681, viewed 10/04/2013, Science Direct, ELSEVIER

Quain & Rasheed 2010, Optimal Trader, University of Georgia, USA, viewed 15/04/2013, <http://www.optimaltrader.net/old/neural_network.htm>

Rashed, F. S. and Mahmoud, T. S. 2009, 'Prediction of wear behaviour of A356/SiCp MMCs using neural networks', *Tribology Inernational.*, vol 42, issue 5, pp. 642–648, viewed 10/04/2013, Science Direct, ELSEVIER

S. Narish, B.F. Yousif, D. Rilling 2011, 'Adhesive wear of thermoplastic composite based on kenaf fibres', *Engineering Tribolergy Journal*, vol 225, issue 2, pp. 101-109, viewed 15/04/2013, Science Direct, ELSEVIER

Tchaban, Griffin and Taylor 1998, 'A comparison between single and combined backpropagation neural networks in prediction of turnover', *Engineering Applications of Artificial Intelligence*, vol 11, pp. 41–47, viewed 8/04/2013, Science Direct, ELSEVIER, item: S0952197697000596

Talal F. Yusaf, D.R. Buttsworth, Khalid H. Saleh, B.F. Yousif, 2010, 'CNG-diesel engine performance and exhaust emission analysis with the aid of artificial neural network' , *Applied Energy*, Volume 87, Issue 5, pp 1661-1669, viewed 3/04/2013 Science Direct, ELSEVIER, item: S0306261909004371

T Nasir,B FYousif, SMcWilliam,N D Salih, and L T Hui *2009, '*An artificial neural network for prediction of the friction coefficient of multi-layer polymeric composites in three different orientations', *J. Mechanical Engineering Science,* Vol. 224 Part C, viewed 24/03/2013,
<http://eprints.usq.edu.au/8317/2/Nasir_Yousif_McWilliam_Salih_Hui_IMechEC_2010_PV.pdf>

University of Georgia, viewed 28/4/2013,
<www.ccrc.uga.edu>

Velten, Reinicke and Friedrich 2000, 'Wear volume prediction with artificial neural networks', *Tribology International*, Vol 33, pp. 731–736, viewed 8/04/2013, Science Direct, ELSEVIER, item: S0301679X00001158

Widrow & Lehr 1990, *Types of neural net works*, IEEE, Reprinted Proc. Vol. 78, no. 9, pp1415-1442

Zhang, Z., Friedrich, K., and Velten, K. 2002, 'Prediction on tribological properties of short fibre composites using artificial neural networks', *Wear,* vol 252, issue 7-8, pp. 668–675, viewed 10/04/2013, Science Direct, ELSEVIER

Appendix A: Project Specification

University of southern Queensland
FACULTY OF ENGINEERING AND SURVEYING
ENG8411/8412 Research Project
Project Specifications

For: **Tyler Griinke**

Topic: DEVELOPMENT OF AN ARTIFICIAL NEURAL NETWORK (ANN) FOR PREDICTING TRIBOLOGICAL PROPERTIES OF KENAF FIBRE REINFORCED EPOXY COMPOSITES.

Supervisors: Dr. Belal Yousif

Enrolment: ENG4111 – S1, ENG4112 – S2, 2013

Project Aim: The project investigates artificial neural network (ANN) modelling for the accurate prediction of friction coefficient and surface temperature of a kenaf fibre reinforced epoxy (KFRE) composite for specific tribological loading conditions.

Programme: (**Issue A, 27 March 2013**)

1) Research ANNs and other investigations involving the development of neural networks used for the prediction of tribological properties.
2) Collect and process sufficient tribology data to utilize in the development and training the ANN.
3) Develop an optimal ANN model through testing various neural, layer and function configurations.
4) Train developed ANN model and compare results with data to confirm accuracy of model. Consider implementing methods to improve network generalisation.
5) Simulate ANN model and make predictions beyond trained domain.
6) Produce academic dissertation.

As time Permits:

4) Try integrating additional tribological data and expanding the domain of the produced neural network.
5) Try integrating additional input and output tribological parameters like fibre orientation and specific wear.
6) Analyse composite structure and assess its possibility for specific applications based on ANN predictions.

AGREED:

___________________(Student) ___________________(Supervisor)

___ /___ /___ . ___ /___ /___ .

Appendix B: DATA

2.8m/s										1.1m/s	
		Friction				Temp(˚C)				Friction	Temp(˚C)
Time	SD	**30N**	50N	70N	100N	**30N**	50N	70N	100N	50N	50N
0	0	0	0	0	0	24	23	23	24	0	22
0.5	0.084	0.92	0.9	0.85	0.979	24	24	24	25	0.95	23
1	0.168	0.85	0.91	0.8	0.938667	24	24.5	24	25	0.91	23
1.5	0.252	0.81	0.85	0.79	0.898333	24.5	25	25	26	0.85	24
2	0.336	0.78	0.81	0.78	0.869	25	25	26	27	0.8	23
2.5	0.42	0.77	0.82	0.78	0.869	26	26	26	27	0.78	24
3	0.504	0.74	0.8	0.77	0.847	26.5	26	26	28	0.79	23
3.5	0.588	0.75	0.8	0.76	0.847	27	27	27	28	0.8	24
4	0.672	0.72	0.78	0.77	0.832333	27.5	27	27	28	0.85	25
4.5	0.756	0.71	0.76	0.78	0.825	28	27	27	29	0.79	25
5	0.84	0.67	0.73	0.78	0.799333	28	28	28	29	0.78	26
5.5	0.924	0.65	0.72	0.77	0.784667	29	28	29	30	0.77	26
6	1.008	0.67	0.72	0.78	0.795667	29	29	30	31	0.77	25
6.5	1.092	0.63	0.7	0.76	0.766333	30	29	30	31	0.76	26
7	1.176	0.61	0.71	0.78	0.77	31	29	31	31	0.75	26
7.5	1.26	0.64	0.72	0.76	0.777333	31	30	32	32	0.75	26
8	1.344	0.6	0.7	0.75	0.751667	32	30	32	32	0.74	27
8.5	1.428	0.59	0.69	0.75	0.744333	32	31	33	34	0.73	27
9	1.512	0.6	0.7	0.73	0.744333	32	31.5	34	35	0.75	27
9.5	1.596	0.58	0.67	0.72	0.722333	32	32	34	35	0.75	28
10	1.68	0.55	0.65	0.71	0.700333	32.5	32	35	35	0.77	28
10.5	1.764	0.53	0.65	0.72	0.696667	33	32	35	36	0.76	29
11	1.848	0.55	0.65	0.68	0.689333	34	32	35	36	0.74	29
11.5	1.932	0.52	0.62	0.69	0.671	34.5	33	36	36	0.75	29
12	2.016	0.52	0.63	0.68	0.671	35	34	36	37	0.76	30
12.5	2.1	0.53	0.62	0.69	0.674667	35	35	36	37	0.77	30
13	2.184	0.51	0.6	0.68	0.656333	35.5	35	36	38	0.74	30
13.5	2.268	0.5	0.61	0.69	0.66	36	35	37	39	0.74	30.5
14	2.352	0.49	0.6	0.68	0.649	36.5	36	38	39	0.74	31
14.5	2.436	0.49	0.6	0.68	0.649	36.5	37	39	40	0.75	31
15	2.52	0.5	0.59	0.68	0.649	37	37	39	40	0.76	31.5
15.5	2.604	0.48	0.59	0.67	0.638	37	38	40	42	0.74	32
16	2.688	0.46	0.58	0.68	0.630667	37.5	39	40	42	0.74	32
16.5	2.772	0.45	0.57	0.66	0.616	38	40	41	43	0.74	33
17	2.856	0.46	0.56	0.63	0.605	38	41	41	43	0.75	33
17.5	2.94	0.42	0.56	0.62	0.586667	38.5	41	42	44	0.73	33
18	3.024	0.44	0.6	0.63	0.612333	39	42	42	45	0.74	34
18.5	3.108	0.42	0.58	0.63	0.597667	39	43	43	45	0.7	34
19	3.192	0.41	0.56	0.62	0.583	39	43	45	46	0.72	35
19.5	3.276	0.4	0.58	0.62	0.586667	40	43	45	47	0.73	35
20	3.36	0.41	0.6	0.61	0.594	40	44	45	48	0.71	36
20.5	3.444	0.42	0.58	0.64	0.601333	40	44	46	49	0.72	36
21	3.528	0.41	0.58	0.63	0.594	41	45	47	50	0.72	36
21.5	3.612	0.42	0.59	0.62	0.597667	41	46	48	50	0.7	37
22	3.696	0.43	0.57	0.6	0.586667	41	46	48	57	0.7	37
22.5	3.78	0.42	0.6	0.59	0.590333	42	46	49	58	0.71	37
23	3.864	0.41	0.57	0.58	0.572	42	47	50	59	0.71	38
23.5	3.948	0.42	0.55	0.57	0.564667	42.5	48	50	59	0.71	38
24	4.032	0.42	0.54	0.58	0.564667	42.5	48	51	60	0.72	39
24.5	4.116	0.42	0.55	0.59	0.572	43	49	51	61	0.7	39
25	4.2	0.43	0.53	0.56	0.557333	43	49	52	61	0.7	40
25.5	4.284	0.44	0.55	0.56	0.568333	43	50	52	62	0.72	40
26	4.368	0.41	0.55	0.55	0.553667	43.5	51	53	63	0.73	41
26.5	4.452	0.42	0.54	0.54	0.55	44	51	54	64	0.71	42
27	4.536	0.44	0.55	0.56	0.568333	44	52	54	65	0.71	43
27.5	4.62	0.45	0.54	0.56	0.568333	44	52	56	66	0.7	44
28	4.704	0.41	0.54	0.54	0.546333	44.5	53	56	67	0.7	45.5
28.5	4.788	0.43	0.55	0.56	0.564667	45	53	57	67	0.7	46
29	4.872	0.43	0.54	0.54	0.553667	45	54	57	68	0.7	47
29.5	4.956	0.42	0.55	0.55	0.557333	45	54	57	69	0.7	47.5
30	5.04	0.41	0.55	0.54	0.55	45.5	55	58	70	0.7	49

Appendix C: MATLAB Code Example

Three_Hidden_Layer.m (Script File)

```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by NFTOOL
% Created Wed Sep 04 10:46:08 EST 2013
%

inputs = importdata('tribol_inputs.mat');
targets = importdata('tribol_targets.mat');

inputs = inputs';
targets = targets';

% Create a Fitting Network
net = network;
net.numInputs = 1;
net.numLayers = 4;
net.biasConnect = [1; 1; 1; 1];
net.inputConnect = [1; 0; 0 ; 0];
net.layerConnect = [0 0 0 0; 1 0 0 0; 0 1 0 0; 0 0 1 0];
net.outputConnect = [0 0 0 1];

net.layers{1}.size = 25;
net.layers{1}.transferFcn = 'tansig';
net.layers{1}.initFcn = 'initnw';

net.layers{2}.size = 15;
net.layers{2}.transferFcn = 'tansig';
net.layers{2}.initFcn = 'initnw';

net.layers{3}.size = 10;
net.layers{3}.transferFcn = 'tansig';
net.layers{3}.initFcn = 'initnw';


% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{1}.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 10/100;


% For help on training function 'trainlm' type: help trainlm
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainbr';  % Levenberg-Marquardt
```

```
% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean squared error

%Set training stopping criteria
net.trainParam.min_grad = 1e-4;
net.trainParam.epochs = 200;
net.trainParam.max_fail = 6;


% Choose Plot Functions
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotregression', 'plotfit'};


% Train the Network
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs);

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

%View the Network
%view(net)

% Plots
%Find indice values of targets and extract train,val and test outputs
trainOutputs = outputs(:,tr.trainInd');
% valOutputs = outputs(:, tr.valInd');
testOutputs = outputs(:, tr.testInd');

figure(1)
% Regression line for coefficient of friction
plotregression(trainTargets(1,tr.trainInd'),trainOutputs(1,:),'Train',
...
testTargets(1,tr.testInd'),testOutputs(1,:),'Testing')

% valTargets(1,tr.valInd'),valOutputs(1,:),'Validating',...

%figure(2)
% Regression line for Temperature
%plotregression(trainTargets(2,tr.trainInd'),trainOutputs(2,:),'Train'
,...
 % valTargets(2,tr.valInd'),valOutputs(2,:),'Validating',...
%testTargets(2,tr.testInd'),testOutputs(2,:),'Testing')

%

%Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
```

```
%figure, plotfit(net,inputs,targets)
%figure, plotregression(targets,outputs)
%figure, ploterrhist(errors)

%gensim(net,-1)
```

Appendix D: Training, Testing and Simulation Results

Transfer Function Assessment

Single Hidden Layer	3-[25]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
Tansig	0.0659	0.0556	0.0791	0.066867
Logsig	0.092	0.0743	0.0824	0.0829
Purelin	4.38	4.38	4.38	4.38

Transfer Function Assessment

Single Hidden Layer	3-[40]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
Tansig	0.0584	0.0492	0.0602	0.055933
Logsig	0.0838	0.0753	0.0724	0.077167
Purelin	4.38	4.38	4.38	4.38

Transfer Function Assessment

Two Hidden Layers	3-[25-10]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
tansig -tansig	0.0439	0.0613	0.0434	0.049533
tansig-logsig	0.0484	0.0585	0.0583	0.055067
logsig-tansig	0.0601	0.0586	0.0837	0.067467
logsig-logsig	0.0703	0.0501	0.0737	0.0647
tansig -purelin	0.0947	0.0852	0.104	0.094633
logsig-purelin	0.101	0.0902	0.0863	0.0925

Transfer Function Assessment

Two Hidden Layers	3-[15-5]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
tansig -tansig	0.0624	0.0576	0.0674	0.062467
tansig-logsig	0.0687	0.0564	0.0754	0.066833
logsig-tansig	0.0659	0.0746	0.0631	0.067867
logsig-logsig	0.0734	0.0809	0.0833	0.0792

Transfer Function Assessment

Three Hidden Layer	3-[25-15-10]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
tansig-tansig-tansig	0.0321	0.0362	0.0356	0.034633
tansig-tansig-logsig	0.0365	0.0366	0.0411	0.038067
tansig-logsig-logsig	0.0416	0.0403	0.036	0.0393
logsig-logsig-logsig	0.0444	0.0431	0.035	0.040833
logsig-logsig-tansig	0.039	0.0434	0.0426	0.041667
logsig-tansig-tansig	0.0417	0.0446	0.0368	0.041033
logsig-tansig-logsig	0.0395	0.0413	0.0386	0.0398
tansig-logsig-tansig	0.0409	0.0422	0.0372	0.0401

Transfer Function Assessment

Three Hidden Layer	3-[15-10-5]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
tansig-tansig-tansig	0.0351	0.04366	0.0429	0.040553
tansig-tansig-logsig	0.0476	0.04582	0.0451	0.046173
tansig-logsig-logsig	0. 0429	0.0485	0.0416	0.04505
logsig-logsig-logsig	0. 0463	0.0428	0.0457	0.04425
logsig-logsig-tansig	0.0469	0.0411	0.0468	0.044933
logsig-tansig-tansig	0.0525	0.047	0.043	0.0475
logsig-tansig-logsig	0.0404	0.0427	0.0513	0.0448
tansig-logsig-tansig	0.0494	0.0425	0.0452	0.0457

Assessing Hidden Layers

Layer Config	Transer Functions	Training Functions	Test 1	Test 2	Test 3	Average
3-[40]-2	tansig	trainlm	0.0364	0.0474	0.0439	0.042567
3-[25]-2	tansig	trainlm	0.0631	0.0495	0.0491	0.0539
3-[25-10]-2	tansig-tansig	trainlm	0.0321	0.032	0.0316	0.0319
3-[15-5]-2	tansig-tansig	trainlm	0.0428	0.0477	0.0484	0.0463
3-[25-15-10]-2	tansig-tansig-tansig	trainlm	0.0269	0.0245	0.0246	0.025333
3-[15-10-5]-2	tansig-tansig-tansig	trainlm	0.0334	0.0381	0.032	0.0345

Neuron Volume Assessment 1

Neuron Configuration	100 epoch	200 epochs	300 epochs
3-[15-10-5]-2	0.043	0.0358	0.0262
3-[20-15-10]-2	0.0428	0.0361	0.0288
3-[25-20-15]-2	0.0343	0.03	0.0222
3-[30-25-20]-2	0.0353	0.026	0.0185
3-[35-30-25]-2	0.0339	0.0222	0.0191

Neuron Volume Assessment 2

Neuron Configuration	100 epoch	200 epochs	300 epochs
3-[30-25-20]-2	0.0353	0.026	0.0185
3-[25-15-5]-2	0.0432	0.0344	0.0294
3-[30-20-10]-2	0.0364	0.0227	0.0192
3-[35-25-15]-2	0.0293	0.0228	0.0191

Training Function Assessment

Single Hidden Layer	3-[25]-2			
Training functions	Test 1	Test 2	Test 3	Average
trainlm	0.0686	0.0682	0.0723	0.0697
trainbr	0.0908	0.11	0.0865	0.095767
trainscg	0.134	0.131	0.136	0.133667
traincgb	0.139	0.133	0.127	0.133
traingdx	0.22	0.363	0.253	0.278667
trainbfg	0.116	0.113	0.119	0.116
trainrp	0.2	0.225	0.222	0.215667
traincgf	0.131	0.133	0.143	0.135667
traincgp	0.146	0.132	0.14	0.139333
trainoss	0.134	0.132	0.134	0.133333
traingdm	3.31	3.74	2.84	3.296667
traingd	6.15	5.57	5.65	5.79

Training Function Assessment

Single Hidden Layer	3-[40]-2			
Training functions	Test 1	Test 2	Test 3	Average
trainlm	0.0664	0.0658	0.0771	0.069767
trainbr	0.103	0.0914	0.104	0.099467
trainscg	0.128	0.117	0.129	0.124667

traincgb	0.118	0.13	0.116	0.121333
traingdx	0.266	0.198	0.236	0.233333
trainbfg	0.0911	0.0915	0.0939	0.092167
trainrp	0.202	0.23	0.171	0.201
traincgf	0.141	0.191	0.12	0.150667
traincgp	0.132	0.126	0.121	0.126333
trainoss	0.134	0.125	0.12	0.126333
traingdm	2.52	3.43	2.95	2.966667
traingd	5.68	4.88	4.8	5.12

Training Function Assessment

Two Hidden Layers	3-[25-10]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
trainlm	0.0553	0.0577	0.0498	0.054267
trainbr	0.0973	0.0892	0.0878	0.091433
trainscg	0.162	0.164	0.142	0.156
traincgb	0.153	0.148	0.167	0.156
trainbfg	0.0954	0.0815	0.0836	0.086833
traincgp	0.178	0.165	0.155	0.166
trainoss	0.183	0.191	0.231	0.201667

Training Function Assessment

Two Hidden Layers	3-[15-5]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
trainlm	0.0664	0.0577	0.0765	0.066867
trainbr	0.0866	0.0833	0.0851	0.085
trainscg	0.185	0.247	0.191	0.207667
traincgb	0.202	0.206	0.159	0.189
trainbfg	0.0995	0.103	0.0903	0.0976
traincgp	0.173	0.2	0.262	0.211667
trainoss	0.207	0.239	0.21	0.218667

Training Function Assessment

Three Hidden Layer	3-[25-15-10]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
trainlm	0.0398	0.0393	0.0385	0.0392
trainbr	0.0463	0.0504	0.0839	0.0602
trainbfg	0.0593	0.0623	0.0602	0.0696

Training Function Assessment

Three Hidden Layer	3-[15-10-5]-2			
Transfer functions	Test 1	Test 2	Test 3	Average
trainlm	0.0436	0.0432	0.0399	0.042233
trainbr	0.0546	0.064	0.0828	0.067133
trainbfg	0.0862	0.0806	0.0791	0.081967

Bayesian Regularization Training

NET	MSE				R (Friction Coefficient)								Effective parameters			
					Training				Test							
	1	2	3	Ave	1	2	3	Ave	1	2	3	Ave	1	2	3	Ave
3-[40]-2	0.07	0.08	0.09	0.08	0.77	0.74	0.80	0.77	0.67	0.91	0.59	0.73	85	90	75	83
3-[25]-2	0.08	0.08	0.08	0.08	0.79	0.85	0.75	0.80	0.48	0.95	0.76	0.73	83	87	75	82
3-[25-10]-2	0.03	0.03	0.04	0.04	0.97	0.95	0.85	0.92	0.77	0.51	0.97	0.75	157	142	121	140
3-[15-5]-2	0.05	0.05	0.06	0.05	0.63	0.66	0.67	0.65	0.46	0.89	0.54	0.63	89	95	91	91
3-[25-15-10]-2	0.03	0.04	0.05	0.04	0.88	0.77	0.73	0.79	0.50	0.67	0.87	0.68	152	135	112	133
3-[15-10-5]-2	0.05	0.05	0.05	0.05	0.71	0.70	0.72	0.71	0.33	0.75	0.55	0.54	108	115	118	113

Early Stopping Training

NET	MSE				R (Friction Coefficient)								Test				Epochs			
					Training				Validation											
	1	2	3	Average	1	2	3	Average	1	2	3	Average	1	2	3	Average	1	2	3	Average
3-[40]-2	0.11	0.11	0.18	0.13	0.79	0.74	0.75	0.76	0.69	0.93	0.83	0.81	0.24	0.50	0.35	0.36	4	40	25	23
3-[25-10]-2	0.23	0.07	0.19	0.16	0.79	0.64	0.74	0.72	0.45	0.59	0.55	0.53	0.38	0.93	0.53	0.61	5	8	7	6.7
3-[25-15-10]-2	0.33	0.09	0.12	0.18	0.68	0.63	0.70	0.67	0.51	0.72	0.74	0.65	0.84	0.85	0.61	0.77	9	5	24	13