# Chapter 3

# Modeling Work

## 3.1 Yield Strength Models

### 3.1.1 Introduction

The conventional method for developing a new weld metal with desired mechanical properties involves the design of a series of weld metals, varying chemical compositions and welding pa-diameters. These welds are then manufactured and tested. A choice is then made of a particular combination of variables which best meets the requirements. Cost and time savings might be achieved with the help of appropriate models which reduce the number of steps needed.

.

The physical models discussed in Chapter 2, based on strengthening mechanisms, are not sufficiently sophisticated to enable a proper treatment of the problem. At the same time linear regression methods are not capable of representing the real behavior which is far from linear when all the factors are taken into account. On the other hand, the neural network methods described in Chapter 2 is ideally suited to complex phenomena with many variables. In the present work, neural networks are used to model the yield strength of weld metal as a function of weld metal chemical composition, welding parameters and heat treatment conditions.

#### 3.1.1.1 Experimental Data base

All of the data collected are from multi run weld deposits in which the joint is designed to minimize dilution from the base metal, to enable specifically the measurement of all weld metal properties. Furthermore, they all represent electric arc welds made using one of the following processes: manual metal arc (MMAW), submerged arc welding (SAW) and tungsten inert gas (TIG). The welding process itself was represented only by the level of heat input. This is because a large number of published papers did not specify welding parameters in sufficient detail to enable the creation of a dataset without missing values. Missing values cannot be tolerated in the method used here. If the effect of a welding process is not properly represented by the heat input and chemical composition, then neglect of any important parameters will make the predictions more ‘noisy'. As discussed below, the noise in the output was found to be acceptable; a greater uncertainty arises from the lack of a uniform coverage of the input space. The data were collected from a large number of sources [33] to [76].

The aim of the neural network analysis was to predict the yield strength as a function of a large number of variables, including the chemical composition, the welding heat input and any heat treatment. The yield strength database consists of 2121 separate experiments. Neural network methods used in this work cannot cope with missing values of any of the variables.

### 3.1.1.2 Yield Strength Database

Table 3.1 shows the range, mean and standard deviation of each variable including the output(yield strength). The purpose here is simply to list the variables and provide an idea of the range covered. It is emphasized however, that unlike linear regression analysis, the information in Table3.1 cannot be used to define the range of applicability of the neural network model. This is because the inputs are in general expected to interact. We shall see later that it is the Bayesian framework of our neural network analysis which allows the calculation of error bars which define the range of useful applicability of the trained network. A visual impression of the spread of data is shown in Fig. 3.1. It can be concluded from Figure. 4.1(a to q) and Figure. 4.2(a to q) that the effect of Carbon, Silicon, Manganese, Sulphur, Phosphorus, Nickel, Chromium, Molybdenum, Vanadium, Copper, Titanium, Boron, Niobium, Heat_input, Interpass_temperature, Post- weld heat treatment temperature and Post-weld heat treatment time on the Yield Strength of Ferritic Steel Welds have been systematically studied by both the methods BNN and GRNN.[27]

It can be concluded from Figure. 4.3.1 to 4.3.18 that the effect in combination of any two input variables (Independent variables) from Carbon, Silicon, Manganese, Sulphur, Phosphorus, Nickel, Chromium, Molybdenum, Vanadium, Copper, Titanium, Boron, Niobium, Heat_input, Interpass_temperature, Post- weld heat treatment temperature and

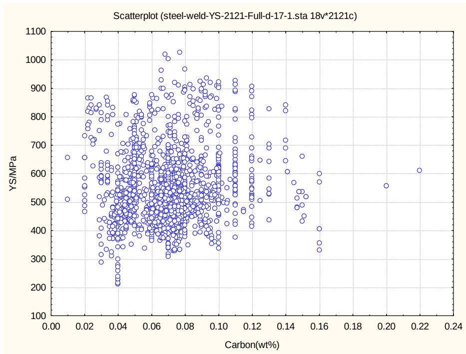
Post-weld heat treatment time on the Yield Strength of Ferritic Steel Welds have been systematically studied by GRNN method.

The prediction of the all Input variables for Targeted Yield Strength by Genetic Algorithms are given in Table 3.4. These can be useful for design of the Ferritic Steel Welds. Genetic Algorithms can be design the Ferritic Steel Welds by extrapolation beyond the existing data.

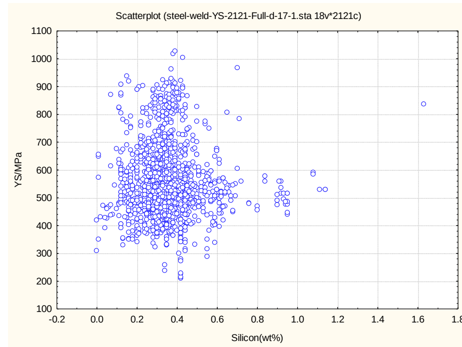
Table 3.1 The Input Variables for Yield Strength Model. “p.p.m .’ corresponds to parts per million by weight.

| Variables | Min | Max | Average | StDev |
|---|---|---|---|---|
| C wt% | 0.01 | 0.22 | 0.0708 | 0.0216 |
| Si wt% | 0 | 1.63 | 0.3467 | 0.1262 |
| Mn wt% | 0.23 | 2.31 | 1.1959 | 0.4175 |
| S wt% | 0.001 | 0.14 | 0.0081 | 0.0051 |
| P wt% | 0.001 | 0.25 | 0.0108 | 0.0075 |
| Ni wt% | 0 | 10.66 | 0.5807 | 1.4971 |
| Cr wt% | 0 | 12.1 | 0.6243 | 1.5961 |
| Mo wt% | 0 | 2.4 | 0.2001 | 0.3591 |
| V wt% | 0 | 0.32 | 0.0191 | 0.0507 |
| Cu wt% | 0 | 2.18 | 0.0659 | 0.2062 |
| Ti ppm | 0 | 1000 | 78.6382 | 122.4481 |
| B ppm | 0 | 200 | 9.2504 | 27.9733 |
| Nb ppm | 0 | 1770 | 53.7704 | 145.3195 |
| HI kJ mm-1 | 0.55 | 7.9 | 1.3573 | 0.9931 |
| IPT C | 20 | 375 | 205.4668 | 42.7739 |
| PWHTT C | 20 | 780 | 328.1428 | 211.1714 |
| PWHTt h | 0 | 50 | 9.4335 | 6.5893 |
| **YS MPa** | **210** | **1026** | **535.7139** | **119.8611** |

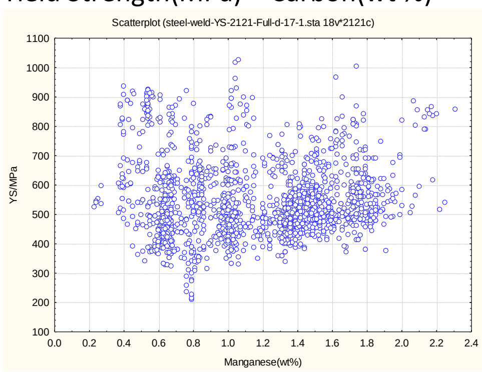
## Scatter Plots of Yield Strength Data – 2121



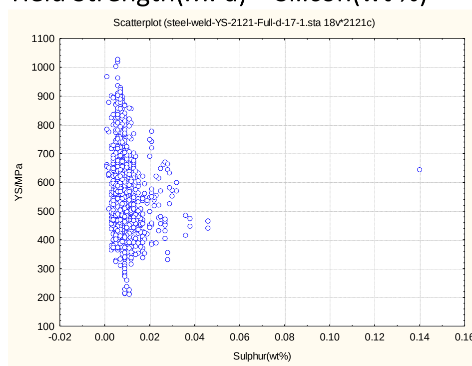

Yield Strength(MPa) – Carbon(wt %)



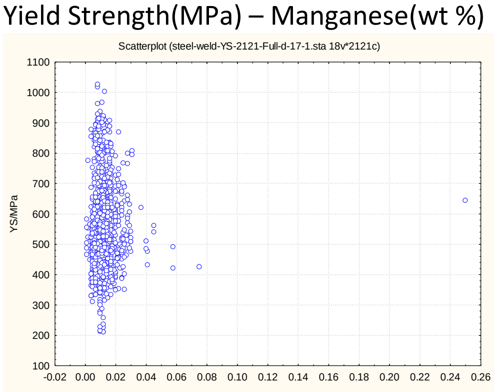

Yield Strength(MPa) – Silicon(wt %)



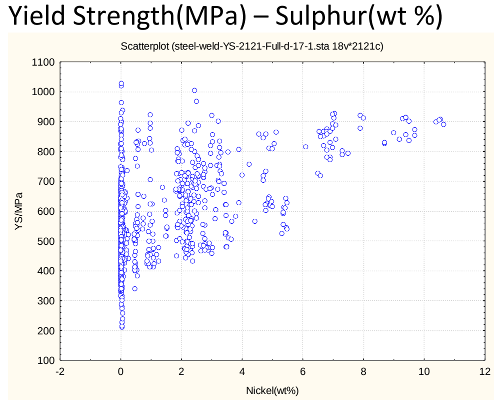

Yield Strength(MPa) – Manganese(wt %)



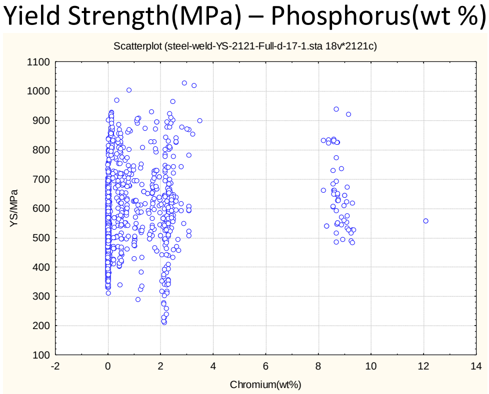

Yield Strength(MPa) – Sulphur(wt %)



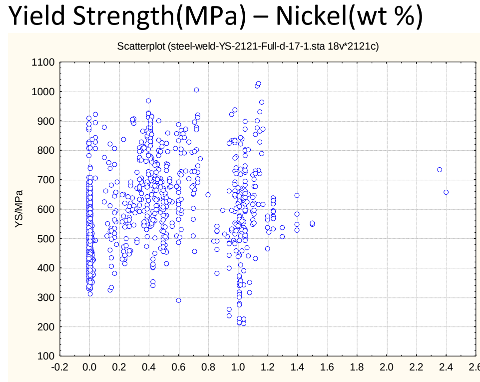

Yield Strength(MPa) – Phosphorus(wt %)




Yield Strength(MPa) – Nickel(wt %)

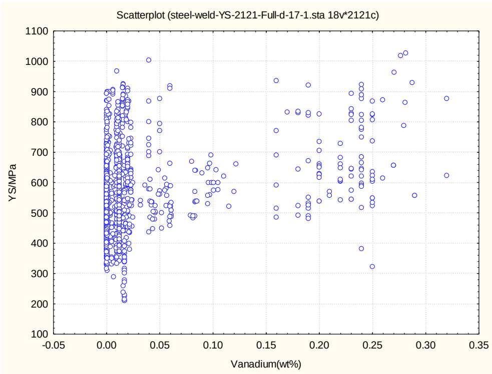



Yield Strength(MPa) – Chromium(wt %)

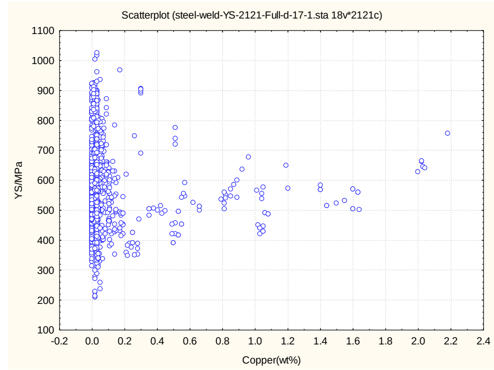



Yield Strength(MPa) – Molybdenum(wt %)

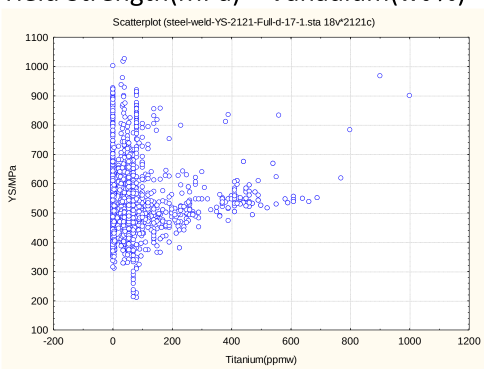
# Scatter Plots of Yield Strength Data- 2121



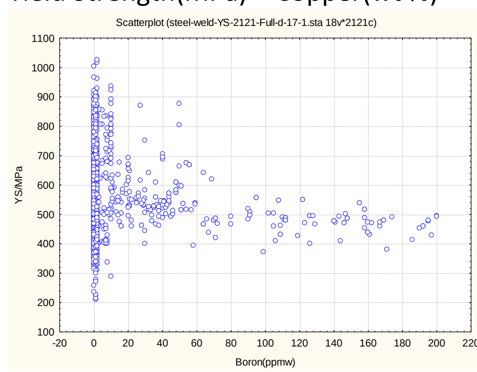

Yield Strength(MPa) – Vanadium(wt %)



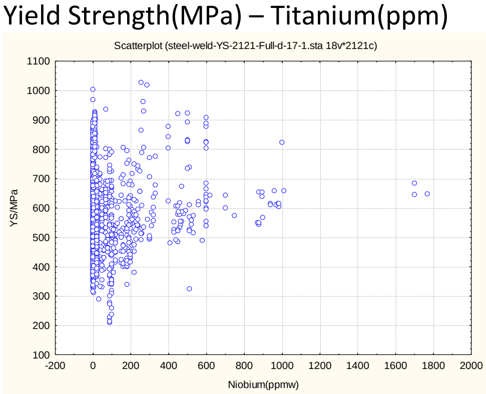

Yield Strength(MPa) – Copper(wt %)



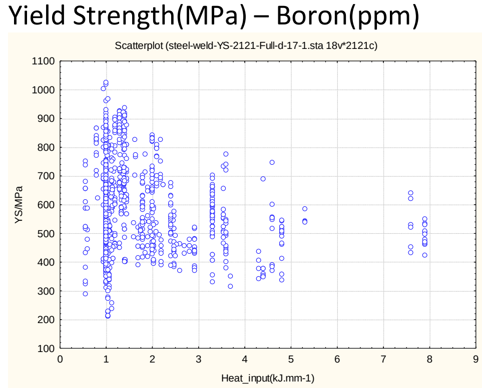

Yield Strength(MPa) – Titanium(ppm)



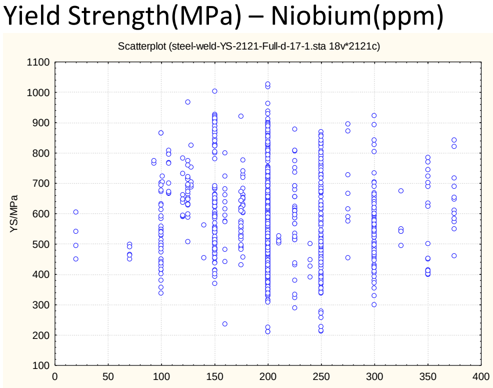

Yield Strength(MPa) – Boron(ppm)



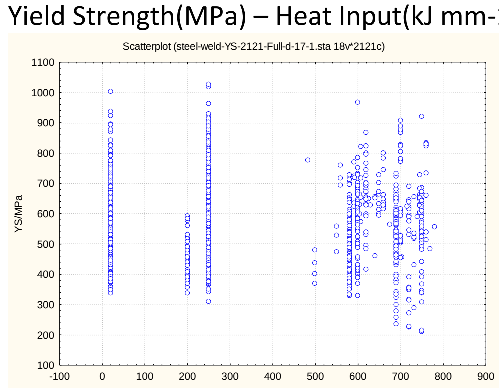

Yield Strength(MPa) – Niobium(ppm)




Yield Strength(MPa) – Heat Input(kJ mm-1)

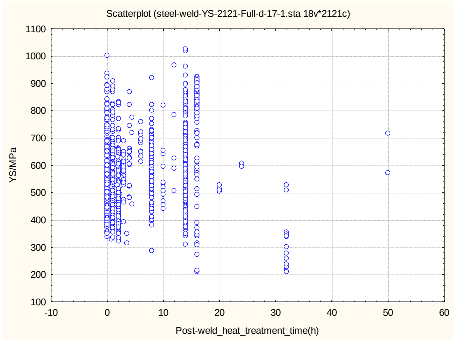



Yield Strength(MPa) – Interpass_temperature(C)




Yield Strength(MPa) – Postweld_HT_Temp(C)

Scatter Plots of Yield Strength Data- 2121




Yield Strength(MPa) – Postweld_HT_Time(h)

Figure 3.1 Database distribution used for yield strength model. "p.p.m .' corresponds to parts per million by weight.

## 3.1.2 Neural Network Models for Yield Strength

### 3.1.2.1 Bayesian Neural Network Modeling and procedure

1 Data of yield strength were collected and plotted in the form of Scatter plots.

2. Data prepared according to the file format required to run in Neural Network Softwares.

(.csv format for Linux base software NeuroMat. And .sta format for Statistica Software)

3. Data were randomly divided into three parts (70% training dataset, 20% validation dataset and 10 % testing dataset). (Training dataset: this data set is used to adjust the weights on the neural network. Validation dataset: this data set is used to minimize overfitting. Testing dataset: this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.)

4. Data of Ferritic Steel Weld Yield Strength 2121 run in NeuroMat Software, which were set in its hyperparameter or Neural network architecture (Software converts the Raw Data into Normalized condition, i.e. it can convert into a specified range like 0 to 1 etc.. for processing) in Chapter 2.10

Neural network architecture for Bayesian Neural Network was set in NeuroMat Software :

Three layers : **Input Layer**(Input Variables), **Hidden Layer** (Algorithms) and **Output Layer** (Yield Strength)

Algorithms : Bigback 5

5. The data of Ferritic Steel Weld Yield Strength 2121 run in NeuroMat Software with above Neural network architecture for best Neural Network Committee model. The best committee model was decided on the basis of smallest test error of the committee model.

6. For best committee model, the data of Ferritic Steel Weld Yield Strength 2121 run in NeuroMat Software repeatedly hundred of times and finalise the best committee model with smallest test error. (NeuroMat gives in single run set of 100 models which required time in hours. Out of these 100 models, the models in the committee are selected on the basis of smallest test error. The number of models in committee varies everytime with repeatedly running the data in NeuroMat. Thus the selection of committee model with a smallest test error is time consuming.)

Some more than hundred yield strength neural network models were trained on a training dataset which consisted of a random selection of 70%of the data 1485 from the yield strength dataset. And 20% of the data 424 from yield strength data set was usedfor cross validation.The remaining 212 data formed the test dataset which was used to see how the model generalizes on unseen data. Each model contained the 17 inputs listed in Table 1 but with different numbers of hidden units or the random seeds used to initiate the values of the weights. Fig. 3.2 shows the results. As expected, the perceived level of noise ($\sigma y$) in the normalised yield strength decreases as the number of hidden units increases, Fig. 3.2a. This is not the case for the test error, which goes through a minimum at five hidden units, Fig. 3.2b, and for the log predictive error which reaches a maximum at seventeen hidden units, Fig. 3.2c.

The error bars presented throughout this work represent a combination of the perceived level of noise σy in the output and the fitting uncertainty estimated from the Bayesian framework. It is evident that there are a few outliers in the plot of the predicted versus measured yield strength for the test dataset, Fig. 3.2f. Each of these outliers has been investigated and found to represent unique data which are not represented in the training dataset, Fig. 3.2e.

It is possible that a committee of models can make a more reliable prediction than an individual model (Chapter 2). The best models are ranked using the values of the log predictive errors Fig. 3.2c. Committees are then formed by combining the predictions of the best L models, where L - 1, 2, . . .; the size of the committee is therefore given by the value of L. A plot of the test error of the committee versus its size gives a minimum which defines the optimum size of the committee, as shown in Fig. 3.2d. The test error associated with the best single model is clearly greater than that of any of the committees Fig. 3.2d. The committee with seven models was found to have an optimum membership with the smallest test error. The committee was therefore retrained on the entire data set without changing the complexity of any of its member models.

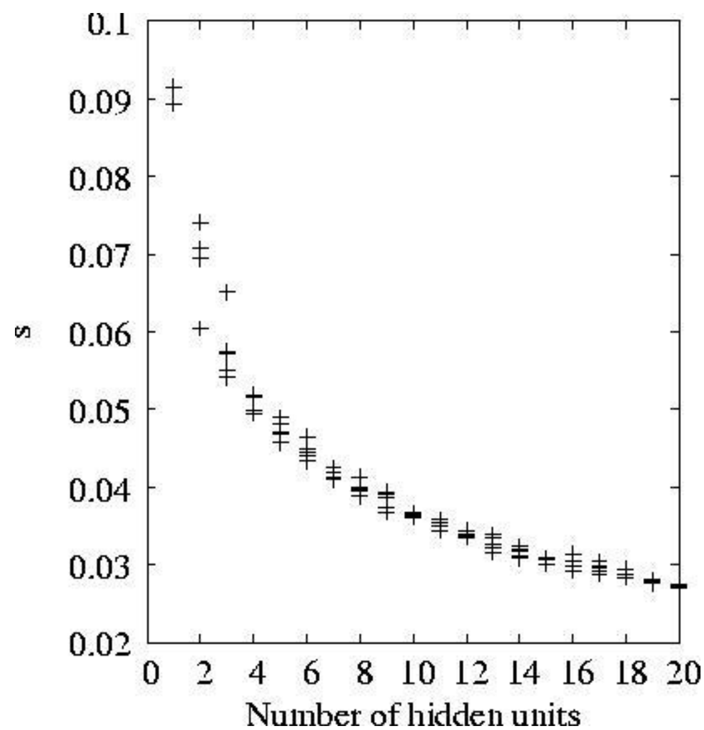



Figure 3.2(a) $\sigma y$ (sigma) vs Hidden units

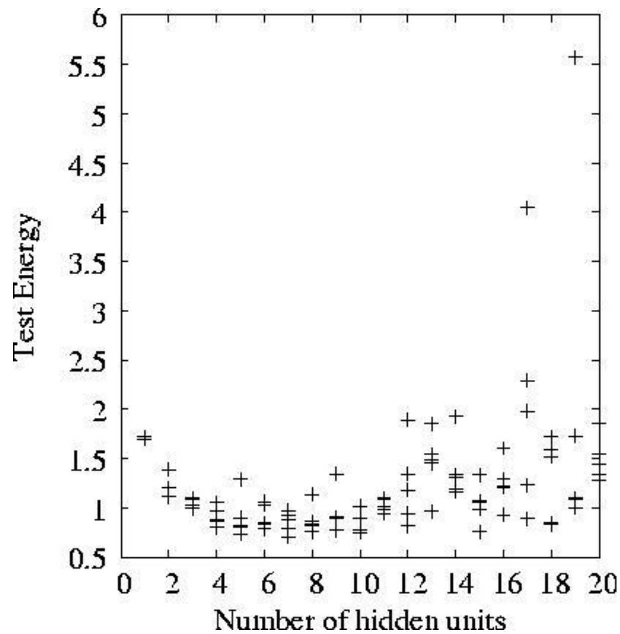



Figure 3.2(b) Test Error (Test Energy) vs Hidden units.

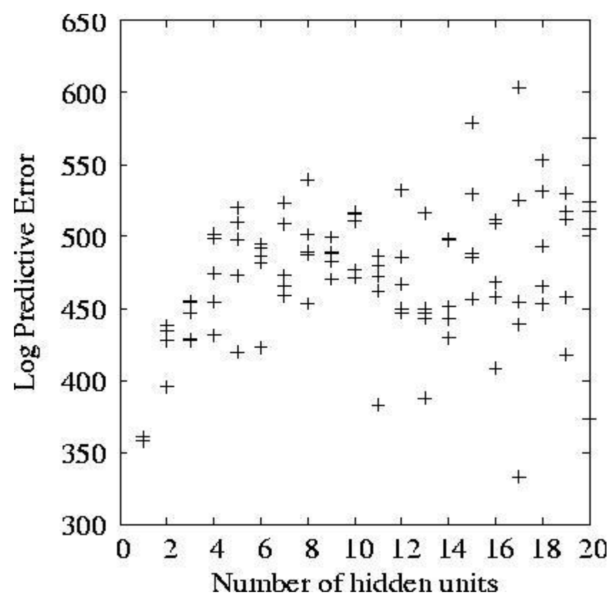



Figure 3.2(c) Log predictive error vs Hidden units

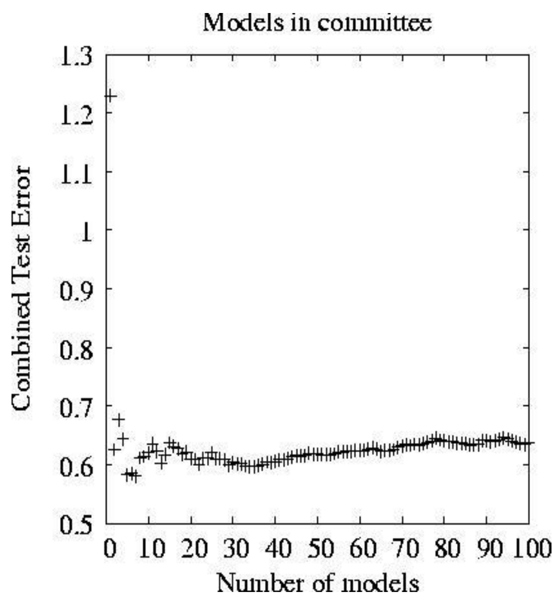



Figure 3.2(d) Test Error vs Models in committee

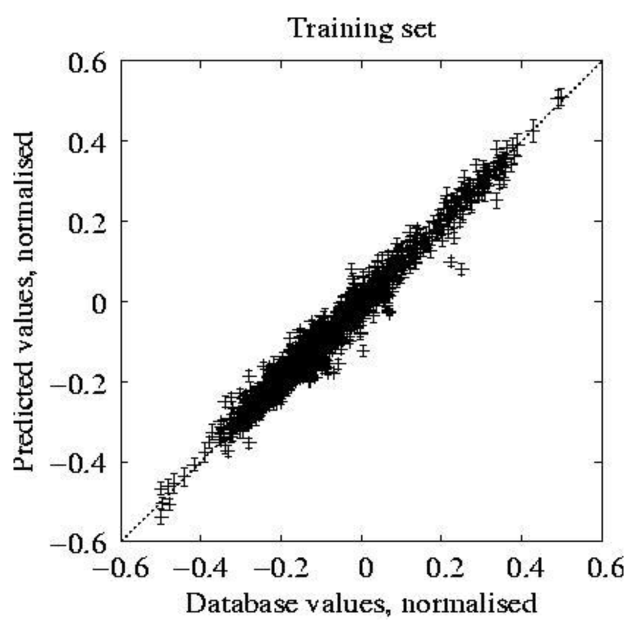



Figure 3.2(e) Predicted normalized Y.S. vs Measured normalized YS. (Training Dataset)

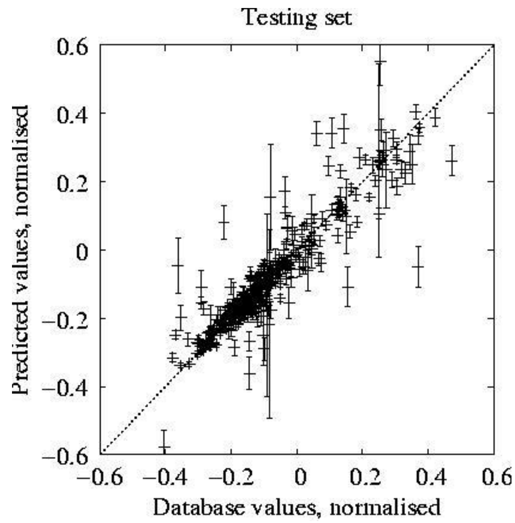



Figure 3.2 (f) Predicted normalized Y.S. vs Measured normalized YS. (Test Dataset)

Figure 3.2 (a to f) Yield Strength (YS) model features.

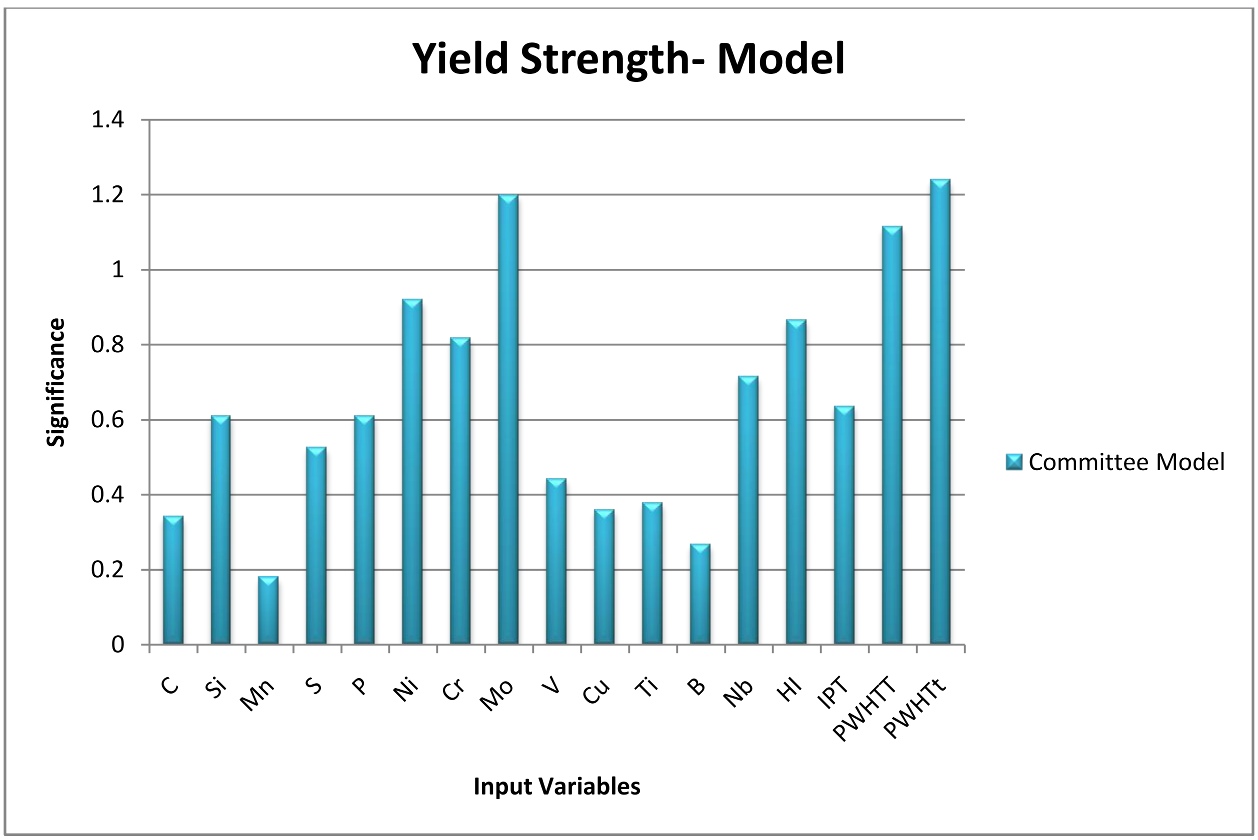



Figure 3.3 The perceived significance $\sigma w$ value of best seven yield strength models in a committee for each of the input variables.

Fig. 3.3indicates the significance σw of each of the input variables, as perceived by first seven neural network models in the committee. The σw value represents the extent to which a particular input explains the variation in the output, rather like a particular correlation coefficient in linear regression analysis. The post-weld heat treatment time on the whole explains a large proportion of variation in the yield strength Figure. 3.3. All variables considered are found to have a significant effect on the output indicating a good choice of inputs.

### 3.1.3 Comparison of Neural network models and procedure (MLP, RBF, and GRNN)

1 Data of yield strength were collected and plotted in the form of Scatter plots.

2. Data prepared according to the file format required to run in Neural Network Softwares.

(.csv format for Linux base software NeuroMat. And .sta format for Statistica Software)

3. Data were randomly divided into three parts (70% training dataset, 20% validation dataset and 10 % testing dataset). (Training dataset: this data set is used to adjust the weights on the neural network. Validation dataset: this data set is used to minimize overfitting. Testing dataset: this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.)

4. Data of Ferritic Steel Weld Yield Strength 2121 run in Statistica Software, which were set in its hyperparameter or Neural network architecture (Software converts the Raw Data into Normalized condition, i.e. it can convert into a specified range like 0 to 1 etc.. for processing) in Chapter 2.11

Neural network architecture was set in Statistica software for MLP, RBF and GRNN :

**MLP 17:17-10-1:1 Algorithms : BP100,CG20,CG18b**

**RBF 17:17-530-1:1 Algorithms : SS,KN,PI**

**GRNN 17:17-1061-2-1:1 Algorithms : SS**

BP Back propagation, CG Conjugate gradient descent, SS (sub) sample, KN K-nearest neighbor (deviation assignment), PI Pseudo-invert (linear least squares), b Best network (the network with lowest selection error in the run was restored)

**A neural network's architecture is of form I:N-N-N:O, where I is the number of input variable, O the number of output variables, N the number of units in each layer.**

5. The data of Ferritic Steel Weld Yield Strength 2121 run in Statistica Software with above Neural network architecture for best Neural Network model in all three MLP, RBF and GRNN.

6. For best model, the data of Ferritic Steel Weld Yield Strength 2121 run in Statistica Software repeatedly hundred of times and finalise the best Neural Network model with smallest training error in all three MLP, RBF and GRNN.

7. The neural network model with the smallest training error was the GRNN model.

Table 3.2 shows the comparison of selected Neural Network models on the basis of their Training Errors. The GRNN models have lowest Training Errors for Yield Strength of Feritic Steel Welds. The GRNN models are selected for modeling from three basic neural network methods (MLP, RBF, and GRNN). Statistica 7.1 software is used for MLP,RBF and GRNN.

Table 3.2 Comparison of Neural network models {MLP, RBF, GRNN}

| **Yield Strength Models** | | | | |
|---|---|---|---|---|
| **MLP** | Train Error | Test Error | Training/Members | Remarks |
| **MLP 17:17-10-1:1 (Model:No.05)** | **0.062442** | **0.078690** | **BP100,CG20,CG18b** | **1 Hidden layer** |
| **MLP 17:17-13-6-1:1 (Model:No.25)** | **0.058963** | **0.067180** | **BP100,CG20,CG59b** | **2 Hidden layers** |
| **MLP 17:17-6-8-13-1:1 (Model:No.14)** | **0.058458** | **0.065638** | **BP100,CG396b** | **3 Hidden layers** |
| **MLP 17:17-14-9-1:1 (Model:No.07)** | **0.036248** | **0.063303** | **BP100,CG458b** | **2 Hidden layers** |
| **MLP 17:17-9-14-1:1 (Model:No.10)** | **0.047847** | **0.058474** | **BP100,CG492b** | **2 Hidden layers** |
| **MLP 17:17-6-7-1:1 (Model:No.18)** | **0.054954** | **0.065891** | **BP100,CG353b** | **2 Hidden layers** |
| **Yield Strength Models** | | | | |
| **RBF** | Train Error | Test Error | Training/Members | Remarks |
| **RBF 17:17-530-1:1 (Model:No.10)** | **0.001791** | **0.002782** | **SS,KN,PI** | **1 H layer** |
| **Yield Strength Models** | | | | |
| **GRNN** | Train Error | Test Error | Training/Members | Remarks |
| **GRNN 17:17-1061-2-1:1 (Model:No.21)** | **0.000668** | **0.004186** | **SS** | **2 H layer** |
| **Note: See Appendix C for Profile String of Statistical Neural Network Software** | | | | |

## 3.1.4 Best GRNN Model for the Yield Srength

The normal behaviour of the Predicted Yield Strength and Observed Yield Strength are observed in the Figure. 3.4 for Training data, Validation data and Testing data. Training of the model is excellent by GRNN method.

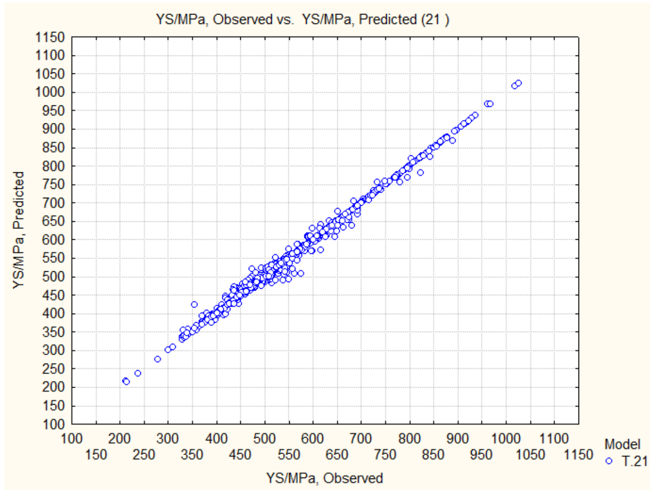


Figure a Training Data for GRNN model of Yield Strength

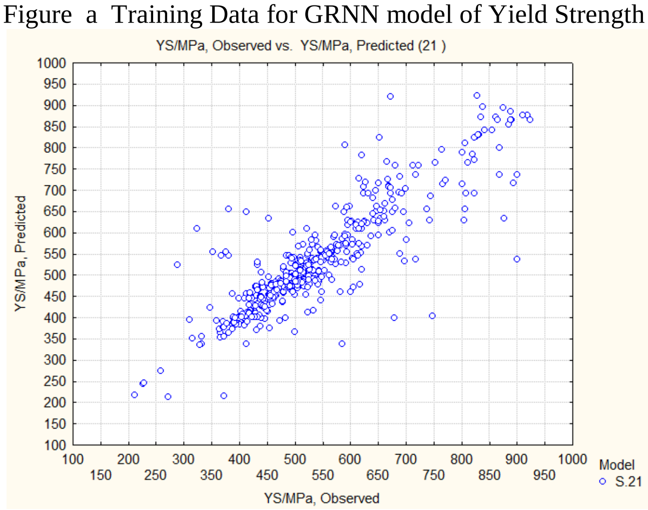


Fig b Validation Data for GRNN model of Yield Strength

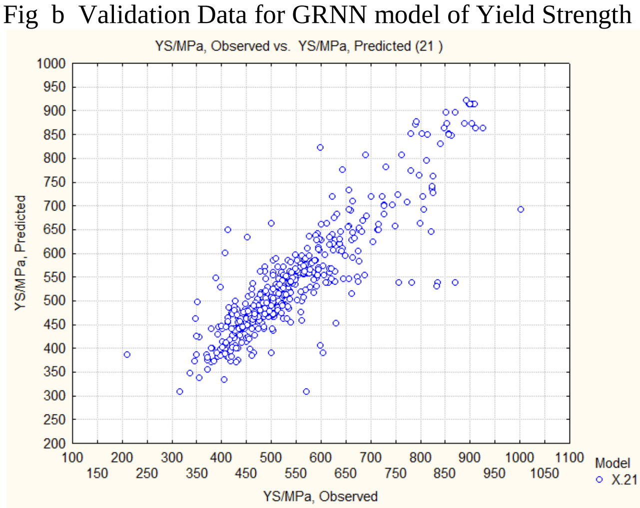


Fig c Test Data for GRNN model of Yield Strength

Figure 3.4 (a to c) Training data, validation data and test data of the Best GRNN model for Yield Strength.

The best model of GRNN has training error 0.000668, validation error (selection error) 0.004426, and testing error 0.004186. This model is used for getting the results in form of various response graphs to understand the trend between the input variables and output variable (Yield Strength).(Figure 4.2)

Table 3.3 Comparison of Significance of Best Trained Models of Yield Strength

| Input Variables | Significance GRNN Model | Significance BNN Model |
|---|---|---|
| Carbon(wt%) | 7 | 15 |
| Silicon(wt%) | 11 | 9 |
| Manganese(wt%) | 3 | 17 |
| Sulphur(wt%) | 17 | 11 |
| Phosphorus(wt%) | 16 | 10 |
| Nickel(wt%) | 1 | 4 |
| Chromium(wt%) | 6 | 6 |
| Molybdenum(wt%) | 4 | 2 |
| Vanadium(wt%) | 5 | 12 |
| Copper(wt%) | 13 | 14 |
| Titanium(ppm) | 14 | 13 |
| Boron(ppm) | 15 | 16 |
| Niobium(ppm) | 10 | 7 |
| Heat_input(kJ.mm-1) | 12 | 5 |
| Interpass_temperature(C) | 8 | 8 |
| Postweld_heat_treatment_temperature(C) | 2 | 3 |
| Post-weld_heat_treatment_time(h) | 9 | 1 |

Table 3.3 shows the comparison of Significance of the GRNN and BNN models. Number 1 indicates highest value of significance and Number 17 lowest value of significance. Most of the Input Variables are closer in significance for both the models. All input variables considered are found to have a significant effect on the output indicating a good choice of inputs.

### 3.1.5 Neural Network and Genetic Algorithms Modelling for Yield Strength of Ferritic Steel Welds

### 3.1.5.1 Genetic Algorithms parameters and procedure

A genetic algorithm has been developed in language C considering the following parameters:

Number of populations = 3

Number of generations = 3000

Population size = 20 chromosomes

When a new generation is created, the following steps are followed: after ranking the 20 chromosomes according to their scores, the first chromosome is copied without change. The chromosomes 2 to 19 are recombined with each others. One gene of one of these chromosomes is mutated between ±0.2%. The chromosome 20, with the worst score, is killed and a new random chromosome is generated and incorporated in the new population.

This program can calculate the best set ( $x_1$ ,$x_2$, ... ,$x$ j) of input parameters for a desired output *y*, which is in this study, the yield strength of ferritic steel welds, for which a Bayesian neural network model was developed[32].

The steps for Genetic Algorithms Modelling:

- First, all the files related to the neural network created for the yield strength were put in the folder "gacode" to optimise. These files were the following:

generate44.exe
norm_test.in
_w*f
*.lu
spec1.t1
outran.x
MINMAX

- Then, the labels of the inputs variables of the neural network were written in the "labels.tct" file
- Then the all inputs variables were define in the "values" file to vary.
- Then, the desired target value of yield strength was normalised and entered it in the "nninput" file, as well as the wanted accuracy.
- Finally, the C program "ga_code" was compiled and executed.

After execution of the Genetic Algorithms program, the output was the values of 17 input variables for given target value of the yield strebgth of Ferritic Steel Weld. The calculation time was in hours.

Three different target values of Ferritic Steel Weld's yield strengths were taken and Genetic Algorithms programs were run. The outputs were given in result and discussion Chapter 4.

# 3.2 Ultimate Tensile Strength Models

## 3.2.1 Experimental Data base

All of the data collected are from multi run weld deposits in which the joint is designed to minimize dilution from the base metal, to enable specifically the measurement of all weld metal properties. Furthermore, they all represent electric arc welds made using one of the following processes: manual metal arc (MMAW), submerged arc welding (SAW) and tungsten inert gas (TIG). The welding process itself was represented only by the level of heat input. This is because a large number of published papers did not specify welding parameters in sufficient detail to enable the creation of a dataset without missing values. Missing values cannot be tolerated in the method used here. If the effect of a welding process is not properly represented by the heat input and chemical composition, then neglect of any important parameters will make the predictions more ‘noisy'. As discussed below, the noise in the output was found to be acceptable; a greater uncertainty arises from the lack of a uniform coverage of the input space. The data were collected from a large number of sources [33] to [76]. The aim of the neural network analysis was to predict the ultimate tensile strength as a function of a large number of variables, including the chemical composition, the welding heat input and any heat treatment. The ultimate tensile strength database consists of 2091 separate experiments. Neural network methods used in this work cannot cope with missing values of any of the variables.

### 3.2.1.1 Ultimate Tensile Strength Database

Table 3.4 shows the range, mean and standard deviation of each variable including the output (ultimate tensile strength). The purpose here is simply to list the variables and provide an idea of the range covered. It is emphasized however, that unlike linear regression analysis, the information in Table 3.4 cannot be used to define the range of applicability of the neural network model. This is because the inputs are in general expected to interact. We shall see later that it is the Bayesian framework of our neural network analysis which allows the calculation of error bars which define the range of useful applicability of the trained network. A visual impression of the spread of data is shown in Fig. 3.5. It can be concluded from Figure. 4.4(a to r) and Figure. 4.5(a to r) that the effect of Carbon, Silicon, Manganese, Sulphur, Phosphorus, Nickel, Chromium, Molybdenum, Vanadium, Copper, Oxygen, Titanium, Boron, Niobium, Heat input, Interpass_temperature, Post- weld heat treatment temperature and Post-weld heat treatment time on the Ultimate Tensile Strength of Ferritic Steel Welds have been systematically studied by BNN and GRNN.[27]

It can be concluded from Figure. 4.6.1 to 4.6.19 that the effect in combination of any two input variables (Independent variables) from Carbon, Silicon, Manganese, Sulphur, Phosphorus, Nickel, Chromium, Molybdenum, Vanadium, Copper, Oxygen, Titanium, Boron, Niobium, Heat_input, Interpass_temperature, Post- weld heat treatment temperature and Post-weld heat treatment time on the Ultimate Tensile Strength of Ferritic Steel Welds have been systematically studied by GRNN.

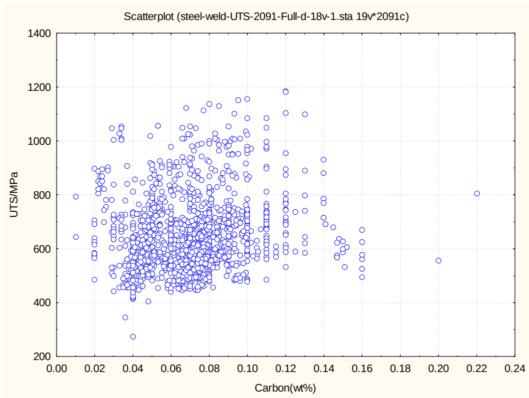
The prediction of the all Input variables for Targeted Ultimate Tensile Strength by Genetic Algorithms are given in Table 4.4 . These can be useful for the design of the Ferritic Steel Welds. Genetic Algorithms can design the Ferritic Steel Welds by extrapolation beyond the existing data.

Table 3.4: The Input Variables for Ultimate Tensile Strength Model.

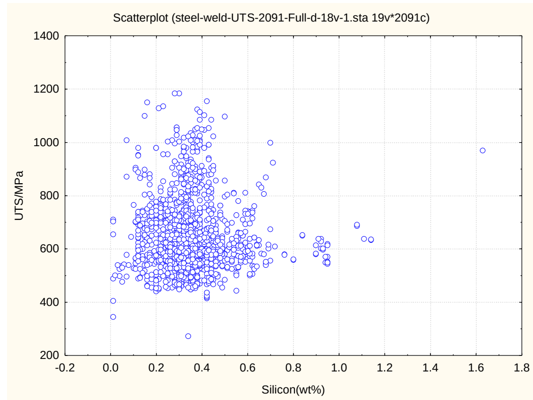
| Variables | Min | Max | Average | StDev |
|---|---|---|---|---|
| C wt% | 0.01 | 0.22 | 0.0705 | 0.021 |
| Si wt% | 0.01 | 1.63 | 0.3477 | 0.1283 |
| Mn wt% | 0.23 | 2.31 | 1.1955 | 0.4156 |
| S wt% | 0.001 | 0.14 | 0.008 | 0.0051 |
| P wt% | 0.001 | 0.25 | 0.0107 | 0.0073 |
| Ni wt% | 0 | 10.66 | 0.581 | 1.5071 |
| Cr wt% | 0 | 12.1 | 0.5869 | 1.4827 |
| Mo wt% | 0 | 2.4 | 0.1988 | 0.3606 |
| V wt% | 0 | 0.32 | 0.0187 | 0.0506 |
| Cu wt% | 0 | 2.18 | 0.0597 | 0.1953 |
| O ppm | 0 | 1650 | 377.6982 | 166.9297 |
| Ti ppm | 0 | 1000 | 80.0548 | 124.85 |
| B ppm | 0 | 200 | 9.3161 | 28.1533 |
| Nb ppm | 0 | 1770 | 51.1751 | 141.6126 |
| HI kJ mm-1 | 0.55 | 7.9 | 1.3392 | 0.9366 |
| IPT C | 20 | 375 | 206.4539 | 41.9047 |
| PWHTT C | 20 | 770 | 333.6054 | 206.2762 |
| PWHTt h | 0 | 50 | 9.7532 | 6.5109 |
| UTS MPa | 273 | 1184 | 621.2198 | 123.4969 |

(“p.p.m .’ corresponds to parts per million by weight.)

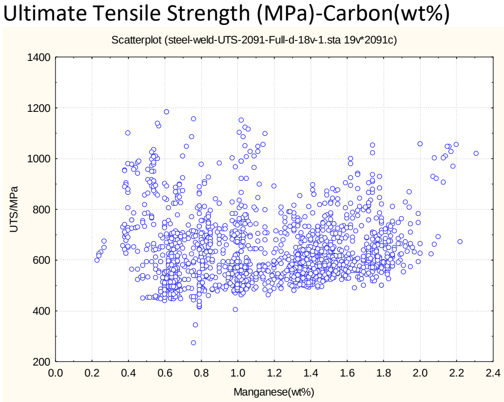
## Scatter Plots of Ultimate Tensile Strength-Data-2091



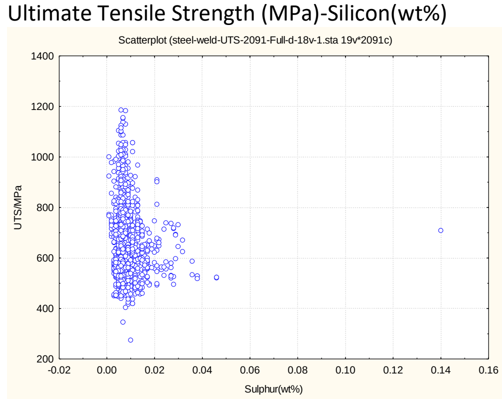
Ultimate Tensile Strength (MPa)-Carbon(wt%)



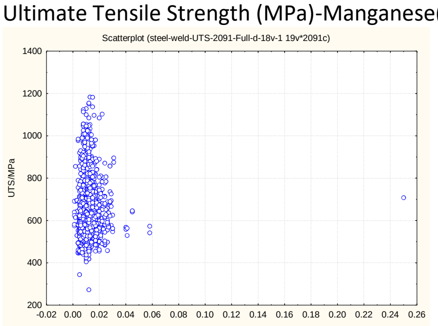
Ultimate Tensile Strength (MPa)-Silicon(wt%)



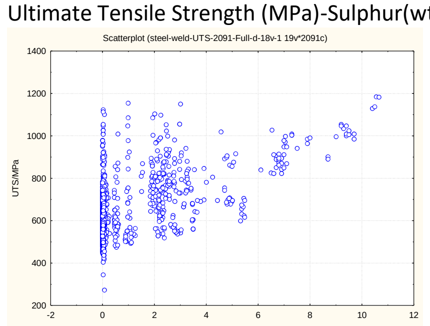
Ultimate Tensile Strength (MPa)-Manganese(wt%)



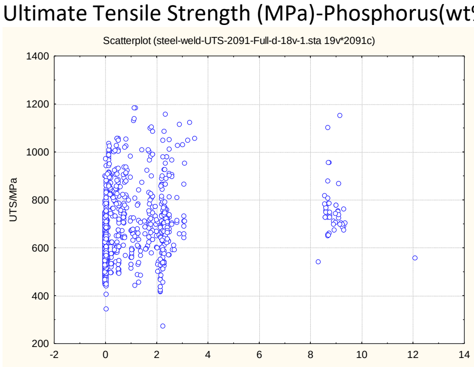
Ultimate Tensile Strength (MPa)-Sulphur(wt%)



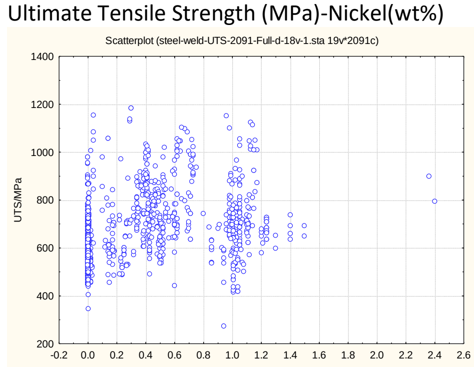
Ultimate Tensile Strength (MPa)-Phosphorus(wt%)



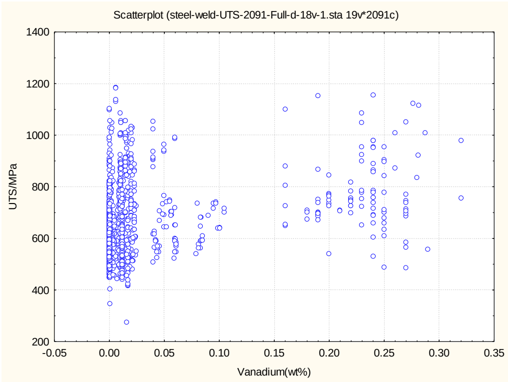
Ultimate Tensile Strength (MPa)-Nickel(wt%)



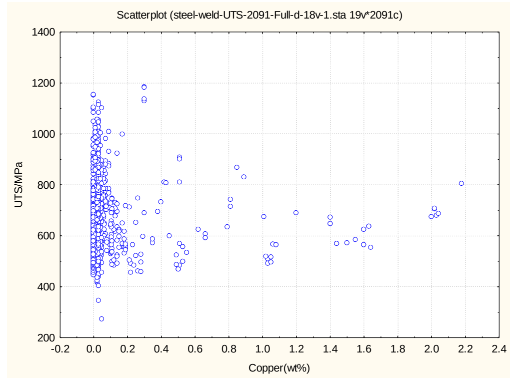
Ultimate Tensile Strength (MPa)-Chromium(wt%)



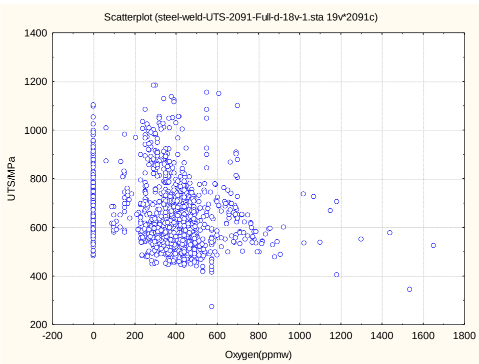
Ultimate Tensile Strength (MPa)-Molybdenum(wt%)



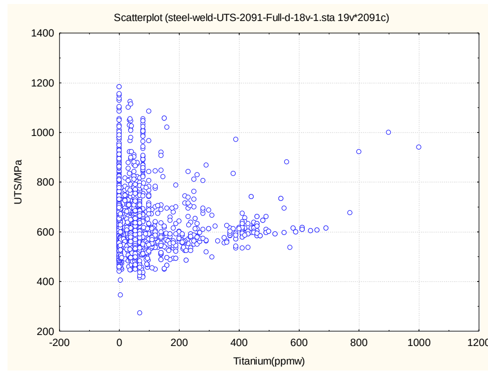
Ultimate Tensile Strength (MPa)-Vanadium(wt%)



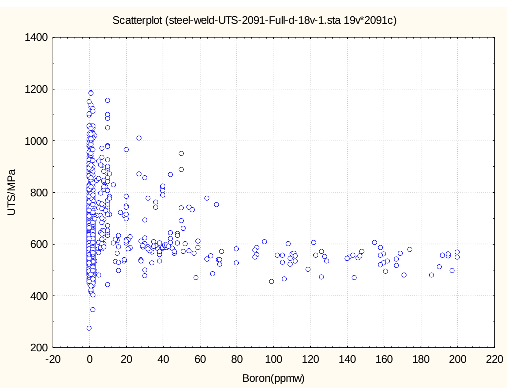
Ultimate Tensile Strength (MPa)-Copper(wt%)



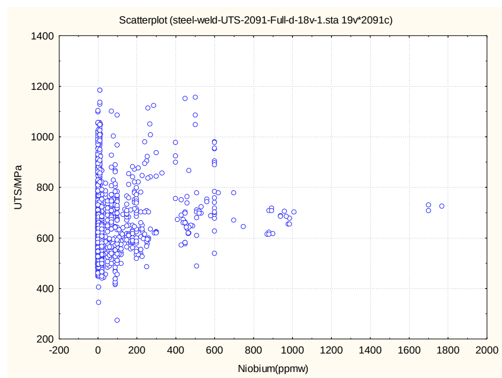
Ultimate Tensile Strength (MPa)-Oxygen(ppm)



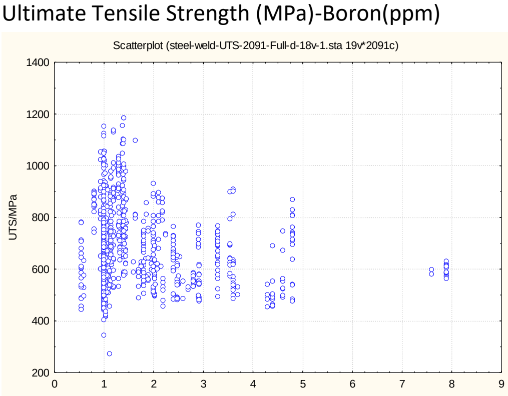
Ultimate Tensile Strength (MPa)-Titanium(ppm)



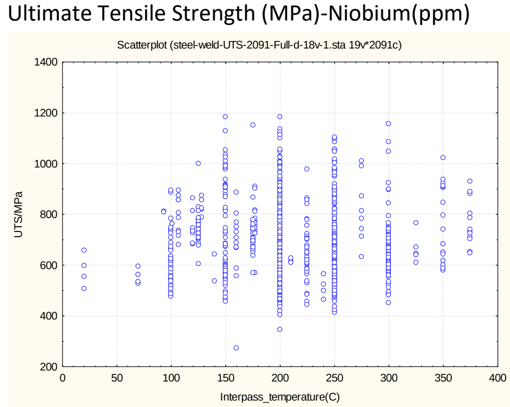
Ultimate Tensile Strength (MPa)-Boron(ppm)



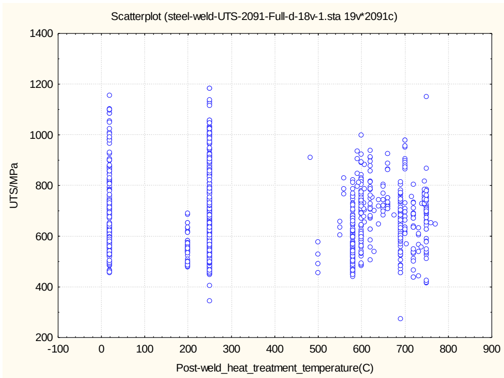
Ultimate Tensile Strength (MPa)-Niobium(ppm)



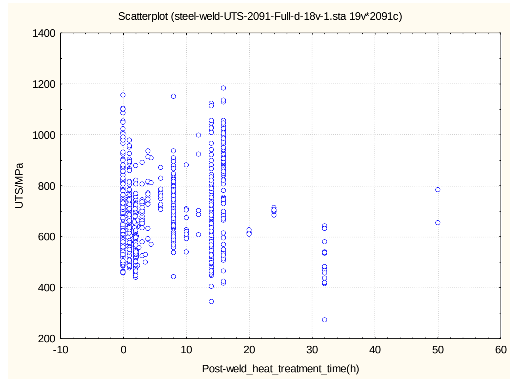
Ultimate Tensile Strength (MPa)-Heat Input(KJ mm-1)



Ultimate Tensile Strength (MPa)-Interpass Temp(C)




Ultimate Tensile Strength (MPa)-Post Weld HT Temp(C)




Ultimate Tensile Strength (MPa)- Post Weld HT Time(h)

Figure 3.5 : Database distribution used for **Ultimate Tensile Strength** model. "p.p.m .' corresponds to parts per million by weight.

## 3.2.2 Neural Network Models for Ultimate Tensile Strength

### 3.2.2.1 Bayesian Neural Network Model and procedure

1 Data of ultimate tensile strength were collected and plotted in the form of Scatter plots.

2. Data prepared according to the file format required to run in Neural Network Softwares.

(.csv format for Linux base software NeuroMat. And .sta format for Statistica Software)

3. Data were randomly divided into three parts (70% training dataset, 20% validation dataset and 10 % testing dataset). (Training dataset: this data set is used to adjust the weights on the neural network. Validation dataset: this data set is used to minimize overfitting. Testing dataset: this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.)

4. Data of Ferritic Steel Weld's Ultimate Tensile Strength 2091 run in NeuroMat Software, which were set in its hyperparameter or Neural network architecture (Software converts the Raw Data into Normalized condition, i.e. it can convert into a specified range like 0 to 1 etc.. for processing) in Chapter 2.10

Neural network architecture for Bayesian Neural Network was set in NeuroMat Software :

Three layers : **Input Layer**(Input Variables), **Hidden Layer** (Algorithms) and **Output Layer** (Yield Strength)

Algorithms : Bigback 5

5. The data of Ferritic Steel Weld's Ultimate Tensile Strength 2091 run in NeuroMat Software with above Neural network architecture for best Neural Network Committee model. The best committee model was decided on the basis of smallest test error of the committee model.

6. For best committee model, the data of Ferritic Steel Weld's Ultimate Tensile Strength 2091 run in NeuroMat Software repeatedly hundred of times and finalise the best committee model with smallest test error. (NeuroMat gives in single run set of 100 models which required time in hours. Out of these 100 models, the models in the committee are selected on the basis of smallest test error. The number of models in committee varies everytime with repeatedly running the data in NeuroMat. Thus the selection of committee model with a smallest test error is time consuming.)

Some more than hundred ultimate tensile strength neural network models were trained on a training dataset which consisted of a random selection of 70%of the data 1464 from the ultimate tensile strength dataset. And 20% of the data 418 from ultimate tensile strength data set was used for cross validation. The remaining 209 data formed the test dataset which was used to see how the model generalizes on unseen data. Each model contained the 18 inputs listed in Table 1 but with different numbers of hidden units or the random seeds used to initiate the values of the weights. Fig. 3.6 shows the results. As expected, the perceived level of noise (σy) in the normalised ultimate tensile strength decreases as the number of hidden units increases, Fig. 3.6a.This is not the case for the test error, which goes through a minimum at sixteen hidden units, Fig. 3.6b, and for the log predictive error which reaches a maximum at seventeen hidden units, Fig. 3.6c.

The error bars presented throughout this work represent a combination of the perceived level of noise $\sigma y$ in the output and the fitting uncertainty estimated from the Bayesian framework. It is evident that there are a few outliers in the plot of the predicted versus measured ultimate tensile strength for the test dataset, Fig. 3.6f. Each of these outliers has been investigated and found to represent unique data which are not represented in the training dataset, Fig. 3.6e. It is possible that a committee of models can make a more reliable prediction than an individual model (Chapter 2).

The best models are ranked using the values of the log predictive errors Fig. 3.6c. Committees are then formed by combining the predictions of the best L models, where L - 1, 2, . . .; the size of the committee is therefore given by the value of L. A plot of the test error of the committee versus its size gives a minimum which defines the optimum size of the committee, as shown in Fig. 3.6d. The test error associated with the best single model is clearly greater than that of any of the committees Fig. 3.6d. The committee with eight models was found to have an optimum membership with the smallest test error. The committee was therefore retrained on the entire data set without changing the complexity of any of its member models.

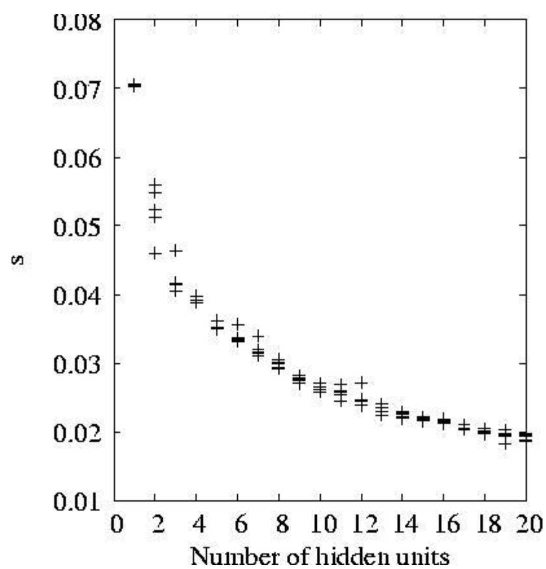



Figure 3. 6 (a) $\sigma y$ vs Hidden units

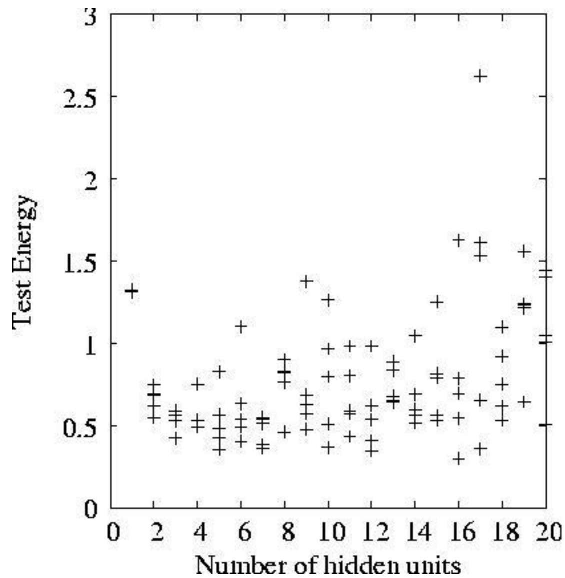



Figure 3. 6 (b) Test Error vs Hidden units.

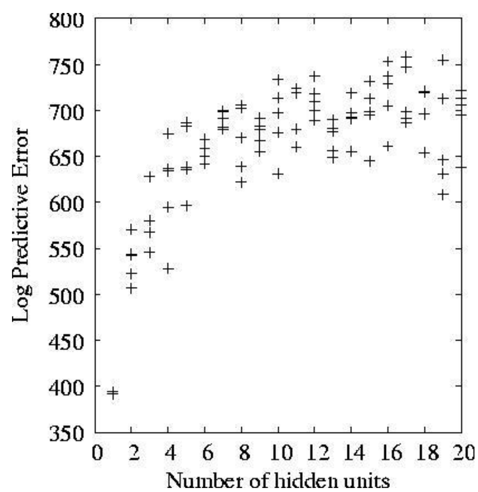



Figure 3. 6 (c) Log predictive error vs Hidden units

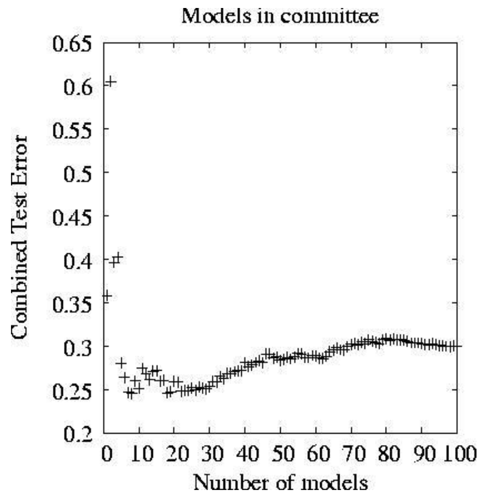



Figure 3. 6 (d) Test Error vs Models in committee

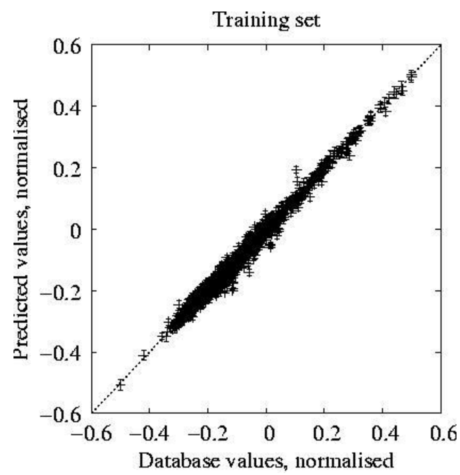



Figure 3. 6 (e) Predicted normalized UTS. vs Measured normalized UTS. (Training Dataset)

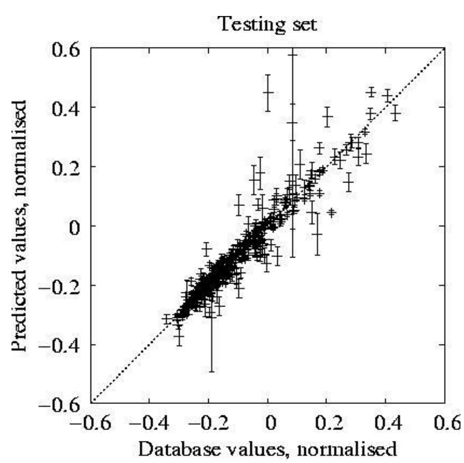



Figure 3. 6 (f) Predicted normalized UTS. vs Measured normalized UTS. (Test Dataset)

Figure 3.6 . (a,b,c,d,e,f) **Ultimate Tensile Strength** (UTS) model features.

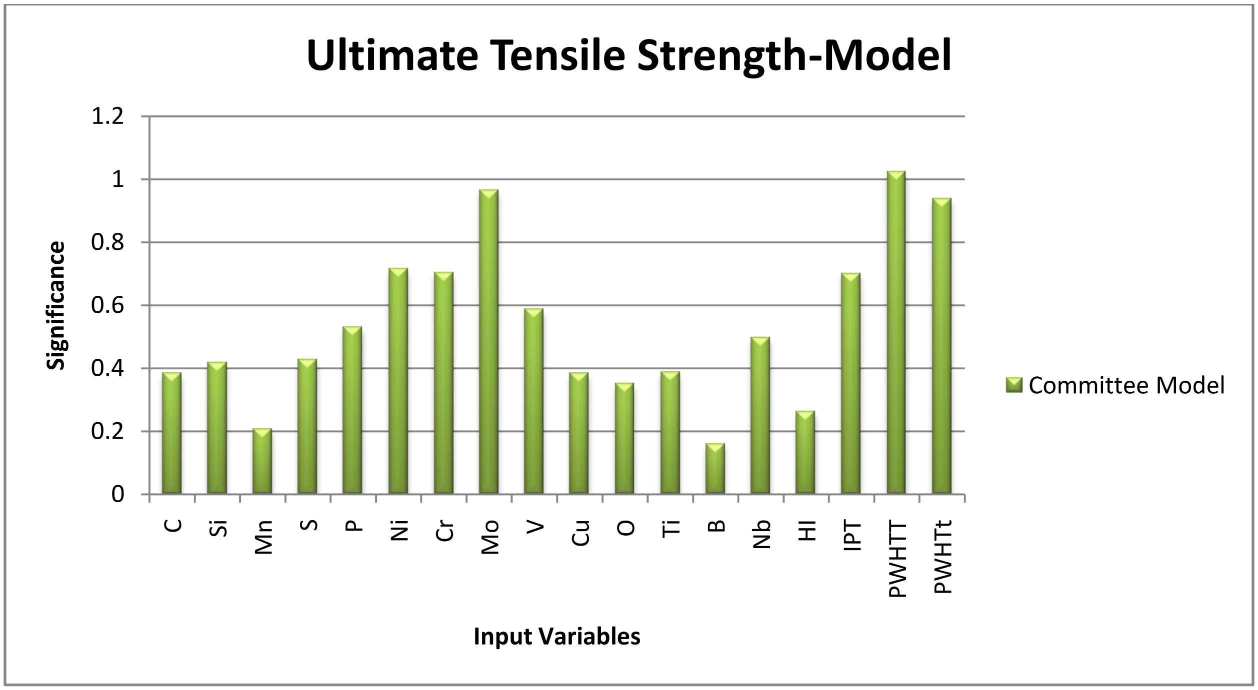



Figure 3.7 The perceived significance $\sigma w$ value of best eight **Ultimate Tensile Strength** models for each of the inputs.

Fig. 3.7 indicates the significance σw of each of the input variables, as perceived by first eight neural network models in the committee. The σw value represents the extent to which a particular input explains the variation in the output, rather like a particular correlation coefficient in linear regression analysis. The post-weld heat treatment temperature on the whole explains a large proportion of variation in the ultimate tensile strength Figure. 3.7. All variables considered are found to have a significant effect on the output indicating a good selection of inputs.

### 3.2.3 Comparision of Neural network models and procedure (MLP, RBF, GRNN)

1 Data of ultimate tensile strength were collected and plotted in the form of Scatter plots.

2. Data prepared according to the file format required to run in Neural Network Softwares.

(.csv format for Linux base software NeuroMat. And .sta format for Statistica Software)

3. Data were randomly divided into three parts (70% training dataset, 20% validation dataset and 10 % testing dataset). (Training dataset: this data set is used to adjust the weights on the neural network. Validation dataset: this data set is used to minimize overfitting. Testing dataset: this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.)

4. Data of Ferritic Steel Weld's Ultimate Tensile Strength 2091 run in Statistica Software, which were set in its hyperparameter or Neural network architecture (Software converts the Raw Data into Normalized condition, i.e. it can convert into a specified range like 0 to 1 etc.. for processing) in Chapter 2.11

Neural network architecture was set in Statistica software for MLP, RBF and GRNN :

**MLP 17:17-10-1:1 Algorithms : BP100,CG20,CG18b**

**RBF 17:17-530-1:1 Algorithms : SS,KN,PI**

**GRNN 17:17-1061-2-1:1 Algorithms : SS**

BP Back propagation, CG Conjugate gradient descent, SS (sub) sample, KN K-nearest neighbor (deviation assignment), PI Pseudo-invert (linear least squares), b Best network (the network with lowest selection error in the run was restored)

**A neural network's architecture is of form I:N-N-N:O, where I is the number of input variable, O the number of output variables, N the number of units in each layer.**

5. The data of Ferritic Steel Weld's Ultimate Tensile Strength 2091 run in Statistica Software with above Neural network architecture for best Neural Network model in all three MLP, RBF and GRNN.

6. For best model, the data of Ferritic Steel Weld's Ultimate Tensile Strength 2091 run in Statistica Software repeatedly hundred of times and finalise the best Neural Network model with smallest training error in all three MLP, RBF and GRNN.

7. The neural network model with the smallest training error was the GRNN model.

Table 3.5 shows the comparision of selected Neural Network models on the basis of their Training Errors. The GRNN models have lowest Training Errors for Ultimate Tensile Strength of Feritic Steel Welds. The GRNN models are selected for modeling from three basic neural network methods (MLP, RBF, GRNN). Statistica 7.1 software is used for MLP,RBF and GRNN.

**Table 3.5 Comparision of Neural network models (MLP, RBF, GRNN)**

| **Ultimate Tensile Strength Models** | | | | |
|---|---|---|---|---|
| **MLP** | Train Error | Test Error | Training/Members | Remarks |
| **MLP 18:18-12-1:1 (Model:No.3)** | **0.035736** | **0.044758** | **BP100,CG481b** | **1 H layer** |
| **MLP 18:18-13-7-1:1 (Model:No.8)** | **0.039741** | **0.060027** | **BP100,CG454b** | **2 H layer** |
| **MLP 18:18-13-8-10-1:1 (Model:No.6)** | **0.039139** | **0.046157** | **BP100,CG478b** | **3 H layer** |
| **Ultimate Tensile Strength Models** | | | | |
| **RBF** | Train Error | Test Error | Training/Members | Remarks |
| **RBF 18:18-81-1:1 (Model:No.18)** | **0.002626** | **0.003150** | **SS,EX,PI** | **1 H layer** |
| **Ultimate Tensile Strength Models** | | | | |
| **GRNN** | Train Error | Test Error | Training/Members | Remarks |
| **GRNN 18:18-1047-2-1:1 (Model:No.1)** | **0.000290** | **0.003402** | **SS** | **2 H layer** |
| **Note: See Appendix C for Profile String of Statistica Neural Network Software** | | | | |

## 3.2.4 Best GRNN Model for Ultimate Tensile Strength

The normal behaviour of the Predicted Ultimate Tensile and Observed Ultimate Tensile Strength are observed in the Figure. 3.8 for Training data, Validation data and Testing data. Training of the model is excellent by GRNN method.

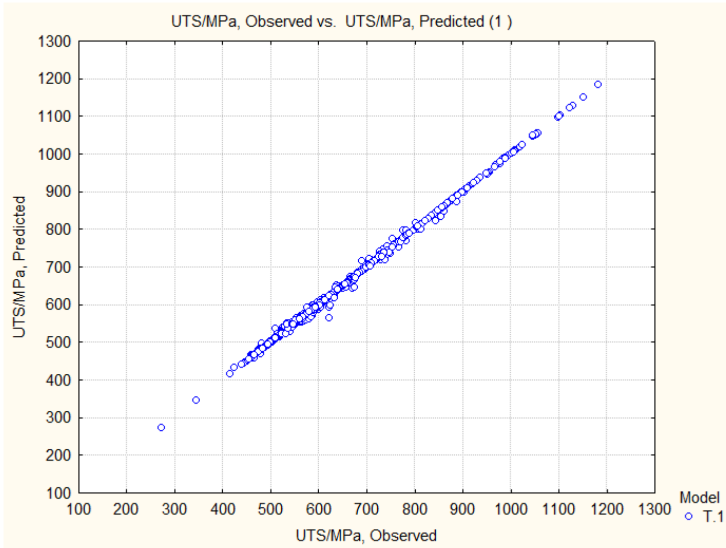


Figure a Training Data for GRNN model of Ultimate Tensile Strength

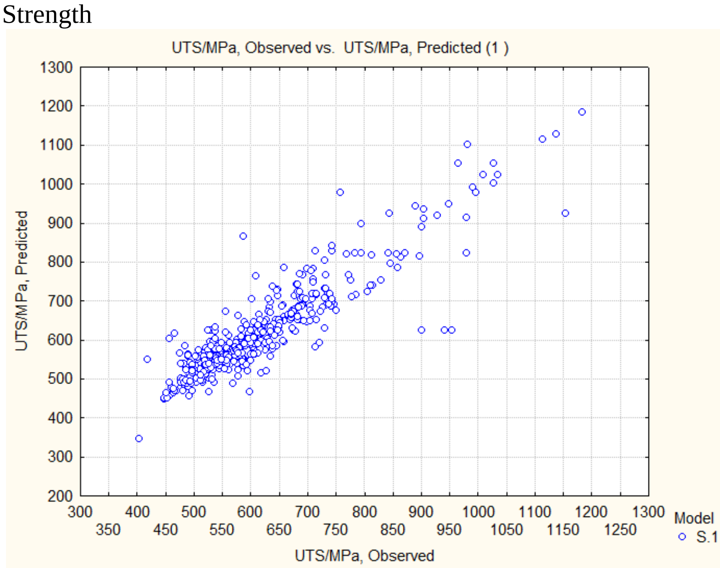


Fig b Validation Data for GRNN model of Ultimate Tensile Strength

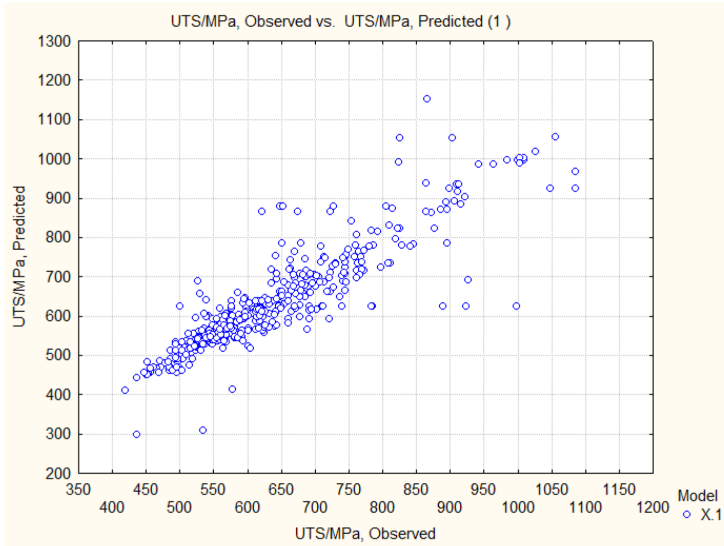



Fig c   Test Data for GRNN model of Ultimate Tensile Strength

Figure 3.8 Training data, validation data and test data of the Best GRNN model for Ultimate tensile Strength.

The best model of GRNN has training error 0.000290, validation error (selection error) 0.003058, and testing error 0.003402. This model is used for getting the results in form of various response graphs to understand the trend between the input variables and output variable (Ultimate Tensile Strength).(Figure 4.5)

Table 3.6 Comparison of Significance of Best Trained Models of Ultimate Tensile Strength

| Input Variables | Significance GRNN Model | Significance BNN Model |
|---|---|---|
| Carbon(wt%) | 9 | 13 |
| Silicon(wt%) | 11 | 11 |
| Manganese(wt%) | 3 | 17 |
| Sulphur(wt%) | 16 | 10 |
| Phosphorus(wt%) | 17 | 8 |
| Nickel(wt%) | 1 | 4 |
| Chromium(wt%) | 6 | 5 |
| Molybdenum(wt%) | 4 | 2 |
| Vanadium(wt%) | 5 | 7 |
| Copper(wt%) | 14 | 14 |
| Oxygen(ppm) | 18 | 15 |
| Titanium(ppm) | 13 | 12 |
| Boron(ppm) | 15 | 18 |
| Niobium(ppm) | 8 | 9 |
| Heat_input(kJ.mm-1) | 12 | 16 |
| Interpass_temperature(C) | 7 | 6 |
| Postweld_heat_treatment_temperature(C) | 2 | 1 |
| Post-weld_heat_treatment_time(h) | 10 | 3 |

Table 3.6. shows the comparison of Significance of the GRNN and BNN models. Number 1 indicates highest value of significance and Number 18 lowest value of significance. Most of the Input Variables are closer in significance for both the models. All input variables considered are found to have a significant effect on the output indicating a good selection of inputs.

## 3.2.5 Neural Network and Genetic Algorithms Modeling for Ultimate Tensile Strength of Ferritic Steel Welds

### 3.2.5.1 Genetic Algorithms parameters and procedure

A genetic algorithm has been developed in language C considering the following parameters:

Number of populations = 3

Number of generations = 3000

Population size = 20 chromosomes

When a new generation is created, the following steps are followed: after ranking the 20 chromosomes according to their scores, the first chromosome is copied without change. The chromosomes 2 to **19** are recombined with each others. One gene of one of these chromosomes is mutated between ±0.2%. The chromosome 20, with the worst score, is killed and a new random chromosome is generated and incorporated in the new population.

This program can calculate the best set ( $x_1$ ,$x_2$, ... ,$x$ j) of input parameters for a desired output *y*, which is in this study, the Ultimate Tensile Strength of ferritic steel welds, for which a Bayesian neural network model was developed[32].

The steps for Genetic Algorithms Modelling:

- First, all the files related to the neural network created for the ultimate tensile strength were put in the folder "gacode" to optimise. These files were the following:

generate44.exe
norm_test.in
_w*f
*.lu
spec1.t1
outran.x
MINMAX

- Then, the labels of the inputs variables of the neural network were written in the "labels.tct" file
- Then the all inputs variables were define in the "values" file to vary.
- Then, the desired target value of ultimate tensile strength was normalised and entered it in the "nninput" file, as well as the wanted accuracy.
- Finally, the C program "ga_code" was compiled and executed.

After execution of the Genetic Algorithms program, the output was the values of 18 input variables for given target value of the ultimate tensile strength of Ferritic Steel Weld. The calculation time was in hours.

Three different target values of Ferritic Steel Weld's ultimate tensile strength were taken and Genetic Algorithms programs were run. The outputs were given in result and discussion Chapter 4.

# 3.3 Elongation Models

## 3.3.1 Experimental Data base

All of the data collected are from multi runweld deposits in which the joint is designed to minimize dilution from the base metal, to enable specifically the measurement of all`weld metal properties. Furthermore, they all represent electric arc welds made using one of the following processes: manual metal arc (MMAW), submerged arc welding (SAW) and tungsten inert gas (TIG). The welding process itself was represented only by the level of heat input. This is because a large number of published papers did not specify welding parameters in sufficient detail to enable the creation of a dataset without missing values. Missing values cannot be tolerated in the method used here. If the effect of a welding process is not properly represented by the heat input and chemical composition, then neglect of any important parameters will make the predictions more ‘noisy'. As discussed below, the noise in the output was found to be acceptable; a greater uncertainty arises from the lack of a uniform coverage of the input space. The data were collected from a large number of sources [33] to [76].

The aim of the neural network analysis was to predict the Elongationas a function of a large number of variables, including the chemical composition, the welding heat input and any heat treatment. The Elongation database consists of 1827 separate experiments. Neural network methods used in this work cannot cope with missing values of any of the variables.

### 3.3.1.1 Elongation Database

Table 3.7 shows the range, mean and standard deviation of each variable including the output (elongation). The purpose here is simply to list the variables and provide an idea of the range covered. It is emphasized however, that unlike linear regression analysis, the information in Table 3.7 cannot be used to define the range of applicability of the neural network model. This is because the inputs are in general expected to interact. We shall see later that it is the Bayesian framework of our neural network analysis which allows the calculation of error bars which define the range of useful applicability of the trained network. A visual impression of the spread of data is shown in Fig. 3.9. It can be concluded from from Figure. 4.7(a to r) and Figure. 4.8(a to r) that the effect of Carbon, Silicon, Manganese, Sulphur, Phosphorus, Nickel, Chromium, Molybdenum, Vanadium, Copper, Oxygen, Titanium, Boron, Niobium, Heat_input, Interpass_temperature, Post- weld heat treatment temperature and Post-weld heat treatment time on the Elongation of Ferritic Steel Welds have been systematically studied by BNN and GRNN.[28]

It can be concluded from Figure. 4.9.1 to 4.9.13 that the effect in combination of any two input variables (Independent variables) from Carbon, Silicon, Manganese, Sulphur, Phosphorus, Nickel, Chromium, Molybdenum, Vanadium, Copper, Oxygen, Titanium, Boron, Niobium, Heat input, Interpass temperature, Post- weld heat treatment temperature and Post-weld heat treatment time on the Elongation of Ferritic Steel Welds have been systematically studied by GRNN.

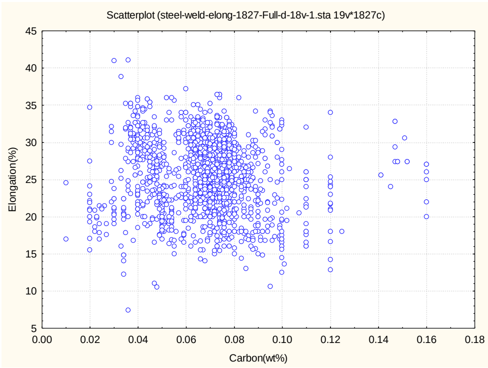
The prediction of the all Input variables for Targeted Elongation by Genetic Algorithms is given in Table 4.8. These can be useful for the design of the Ferritic Steel Welds. Genetic Algorithms can design the Ferritic Steel Welds by extrapolation beyond the existing data.

Table 3.7 shows the range, mean and standard deviation of each variable including the output (Elongation).

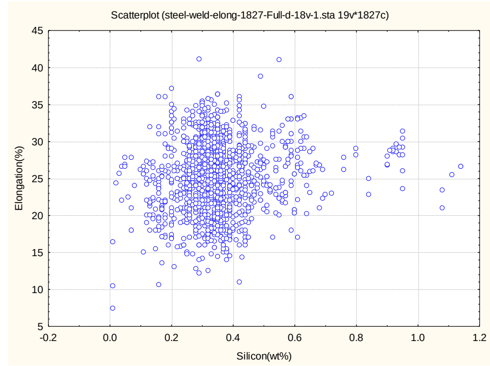
Table 3.7 The Input Variables for **Elongation** Model. “p.p.m .’ corresponds to parts per million by weight.

| Variables | Min | Max | Average | StDev |
|---|---|---|---|---|
| C wt% | 0.01 | 0.16 | 0.0688 | 0.0189 |
| Si wt% | 0.01 | 1.14 | 0.352 | 0.1229 |
| Mn wt% | 0.24 | 2.31 | 1.2102 | 0.3986 |
| S wt% | 0.002 | 0.14 | 0.0078 | 0.0049 |
| P wt% | 0.001 | 0.25 | 0.0101 | 0.0071 |
| Ni wt% | 0 | 10.66 | 0.5374 | 1.5246 |
| Cr wt% | 0 | 9.35 | 0.4452 | 1.1844 |
| Mo wt% | 0 | 2.4 | 0.1798 | 0.3569 |
| V wt% | 0 | 0.32 | 0.0151 | 0.0437 |
| Cu wt% | 0 | 2.04 | 0.0628 | 0.202 |
| O ppm | 63 | 1650 | 411.2567 | 117.9406 |
| Ti ppm | 0 | 1000 | 84.9978 | 126.1291 |
| B ppm | 0 | 200 | 10.306 | 29.8403 |
| Nb ppm | 0 | 1770 | 47.0246 | 139.0368 |
| HI kJ mm-1 | 0.55 | 4.8 | 1.2294 | 0.7057 |
| IPT C | 20 | 350 | 203.8697 | 35.2603 |
| PWHTT C | 20 | 750 | 319.5599 | 188.6206 |
| PWHTT h | 0 | 32 | 10.3452 | 6.1765 |
| Elongation % | 7.4 | 41.1 | 25.6466 | 4.6985 |

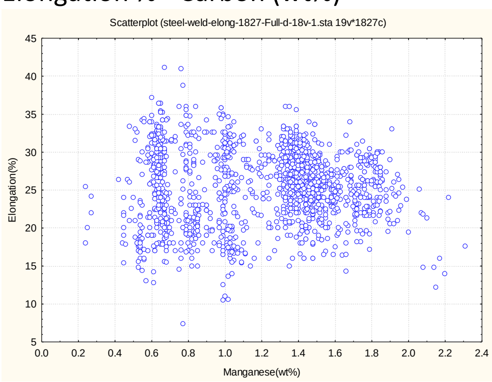
## Scatter Plots of Elongation- Data-1827



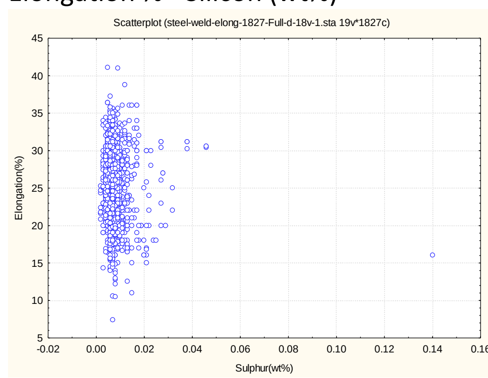

Elongation % - Carbon (wt%)



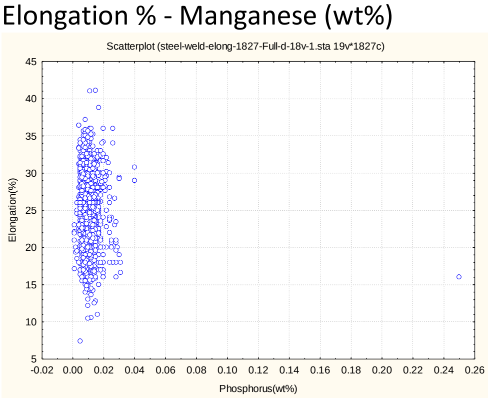

Elongation % - Silicon (wt%)



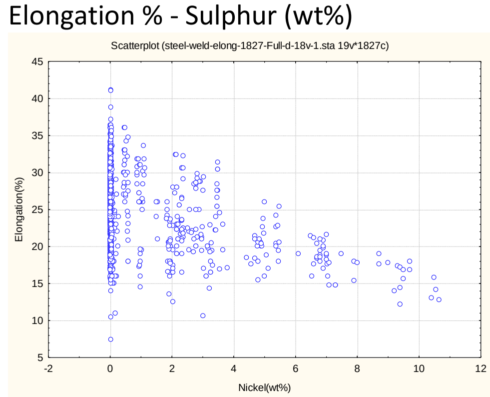

Elongation % - Manganese (wt%)



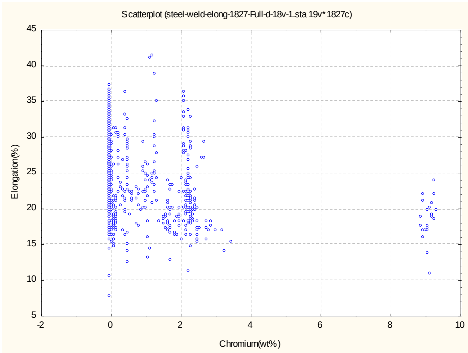

Elongation % - Sulphur (wt%)



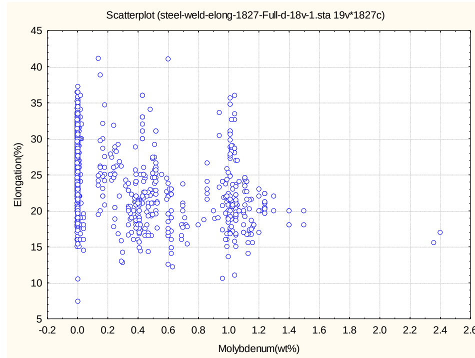

Elongation % - Phosphorus (wt%)




Elongation % - Nickel (wt%)

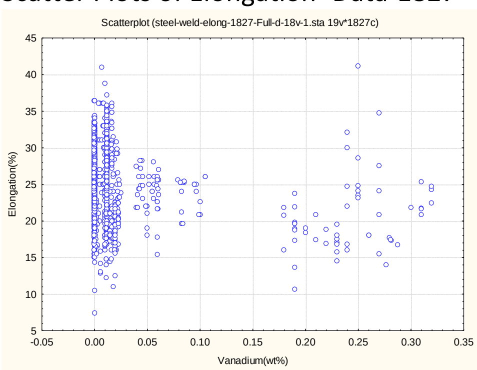



Elongation % - Chromium (wt%)

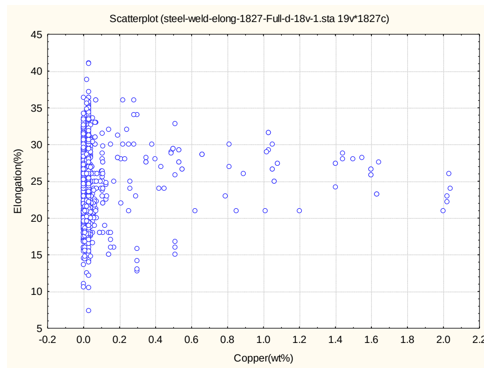



Elongation % - Molybdenum (wt%)

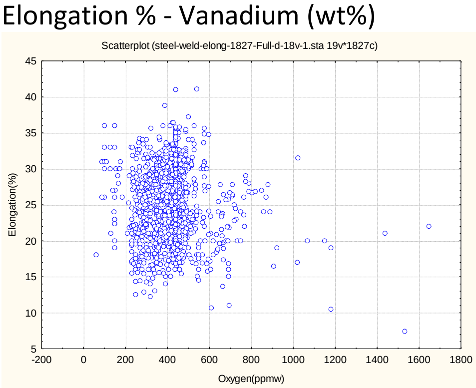
## Scatter Plots of Elongation- Data-1827



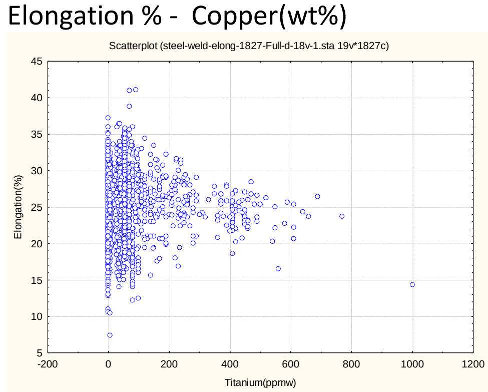

Elongation % - Vanadium (wt%)



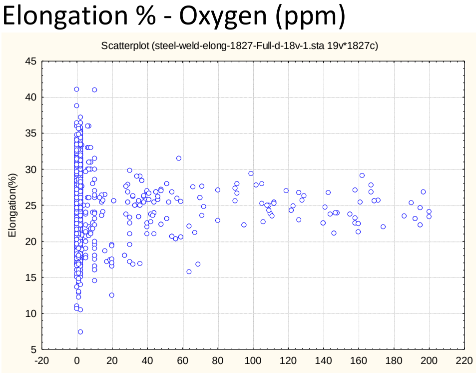

Elongation % -  Copper(wt%)



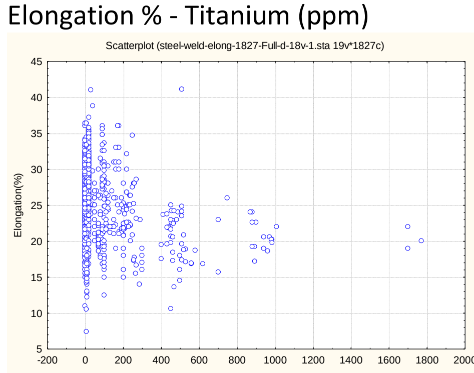

Elongation % - Oxygen (ppm)



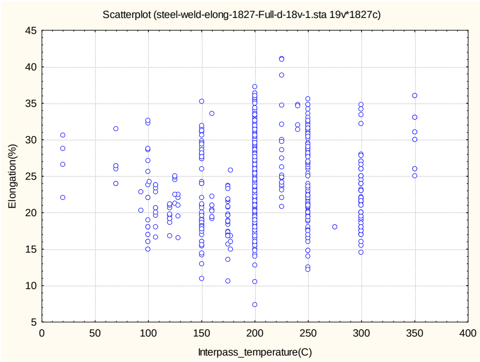

Elongation % - Titanium (ppm)



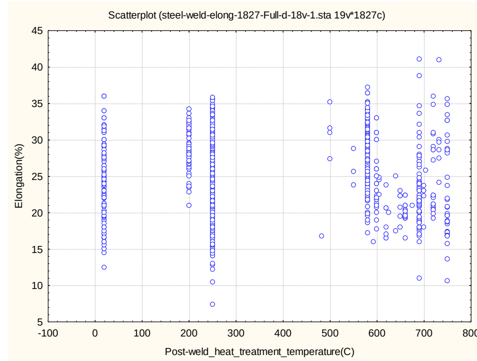

Elongation % - Boron (ppm)




Elongation % -  Niobium(ppm)

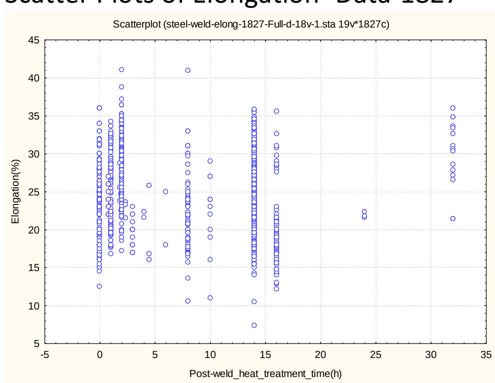



Elongation % - Interpass Temp(C)




Elongation % - Postweld Heat Treatment Temp (C)

Scatter Plots of Elongation- Data-1827




Elongation % - Postweld Heat Treatment Time (h)

Figure 3.9 : Database distribution used for **Elongation** model. "p.p.m .' corresponds to parts per million by weight.

### 3.3.2 Neural Network Models for Elongation

### 3.3.2.1 Bayesian Neural Network Model and procedure

1 Data of elongation were collected and plotted in the form of Scatter plots.

2. Data prepared according to the file format required to run in Neural Network Softwares.

(.csv format for Linux base software NeuroMat. And .sta format for Statistica Software)

3. Data were randomly divided into three parts (70% training dataset, 20% validation dataset and 10 % testing dataset). (Training dataset: this data set is used to adjust the weights on the neural network. Validation dataset: this data set is used to minimize overfitting. Testing dataset: this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.)

4. Data of Ferritic Steel Weld's Elongation 1827 run in NeuroMat Software, which were set in its hyperparameter or Neural network architecture (Software converts the Raw Data into Normalized condition, i.e. it can convert into a specified range like 0 to 1 etc.. for processing) in Chapter 2.10

Neural network architecture for Bayesian Neural Network was set in NeuroMat Software :

Three layers : **Input Layer**(Input Variables), **Hidden Layer** (Algorithms) and **Output Layer** (Yield Strength)

Algorithms : Bigback 5

5. The data of Ferritic Steel Weld's Elongation 1827 run in NeuroMat Software with above Neural network architecture for best Neural Network Committee model. The best committee model was decided on the basis of smallest test error of the committee model.

6. For best committee model, the data of Ferritic Steel Weld's Elongation 1827 run in NeuroMat Software repeatedly hundred of times and finalise the best committee model with smallest test error. (NeuroMat gives in single run set of 100 models which required time in hours. Out of these 100 models, the models in the committee are selected on the basis of smallest test error. The number of models in committee varies everytime with repeatedly running the data in NeuroMat. Thus the selection of committee model with a smallest test error is time consuming.)

Some more than hundred elongation neural network models were trained on a training dataset which consisted of a random selection of 70%of the data 1279 from the ultimate tensile strength dataset. And 20% of the data 365 from ultimate tensile strength data set was used for cross validation.The remaining 183 data formed the test dataset which was used to see how the model generalizes on unseen data. Each model contained the 18 inputs listed in Table 3.7 but with different numbers of hidden units or the random seeds used to initiate the values of the weights. Fig. 3.10 shows the results. As expected, the perceived level of noise (σy) in the normalised ultimate tensile strength decreases as the number of hidden units increases, Fig. 3.10a. This is not the case for the test error, which goes through a minimum at six hidden units, Fig. 3.10b, and for the log predictive error which reaches a maximum at twelve hidden units, Fig. 3.10c.

The error bars presented throughout this work represent a combination of the perceived level of noise $\sigma y$ in the output and the fitting uncertainty estimated from the Bayesian framework. It is evident that there are a few outliers in the plot of the predicted versus measured Elongation for the test dataset, Fig. 3.10f. Each of these outliers has been investigated and found to represent unique data which are not represented in the training dataset, Fig. 3.10e.

It is possible that a committee of models can make a more reliable prediction than an individual model (Chapter 2). The best models are ranked using the values of the log predictive errors Fig. 3.10c. Committees are then formed by combining the predictions of the best L models, where L - 1, 2, . . .; the size of the committee is therefore given by the value of L. A plot of the test error of the committee versus its size gives a minimum which defines the optimum size of the committee, as shown in Fig. 3.10d.

The test error associated with the best single model is clearly greater than that of any of the committees Fig. 3.10d. The committee with two models was found to have an optimum membership with the smallest test error. The committee was therefore retrained on the entire data set without changing the complexity of any of its member models.

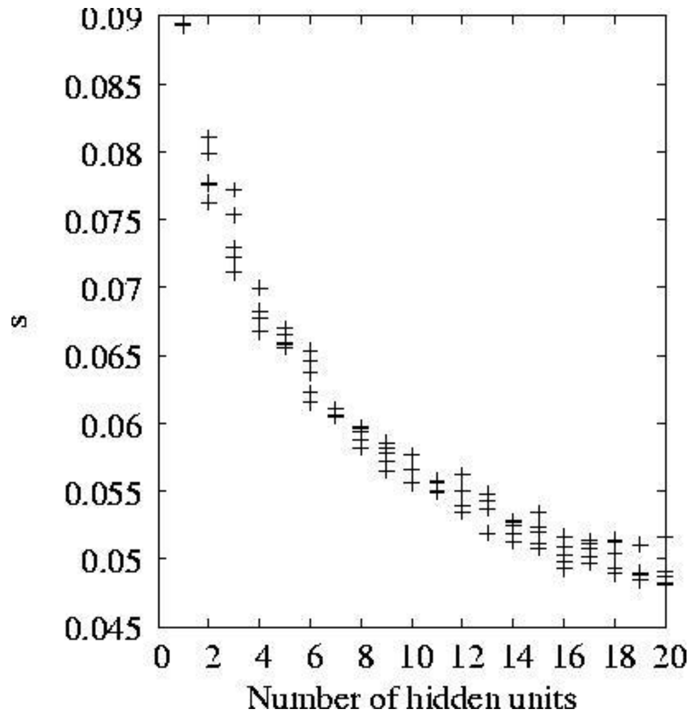



Figure 3.10 (a) $\boldsymbol{\sigma y}$ vs **Hidden units**

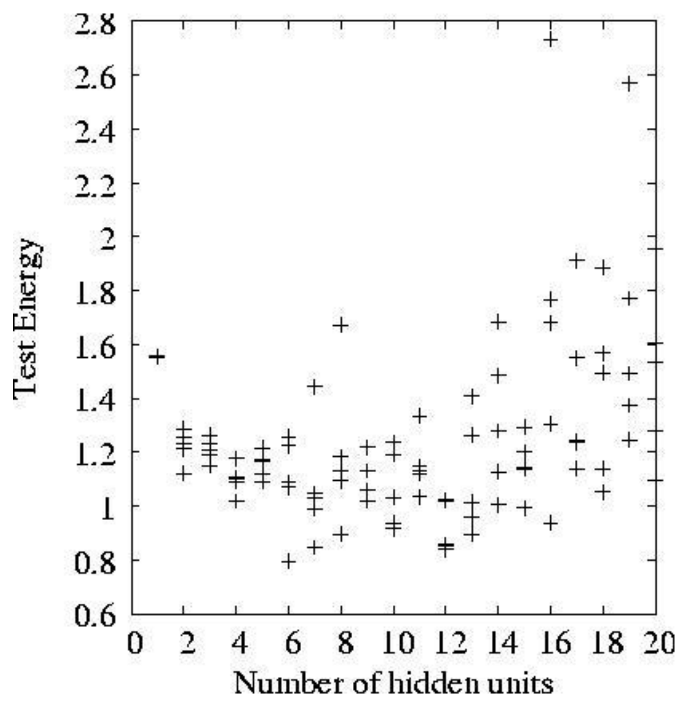



Figure 3.10 (b) **Test Error** vs **Hidden units.**

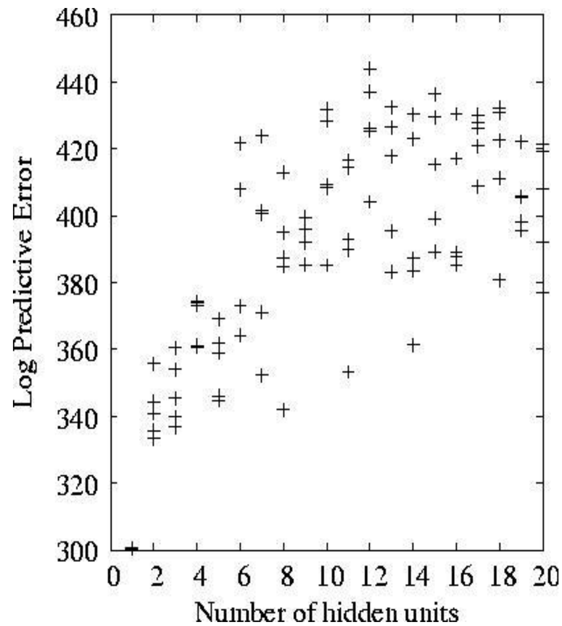



Figure 3.10 (c) **Log predictive error** vs **Hidden units**

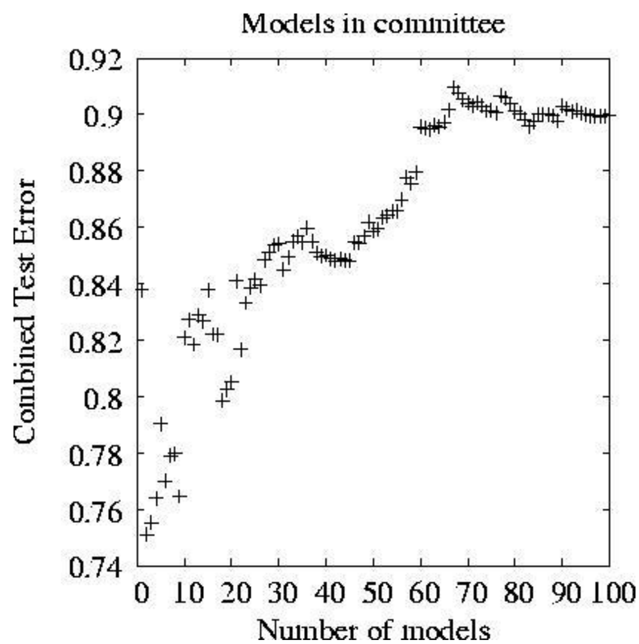



Figure 3.10 (d) **Test Error** vs **Models in committee**

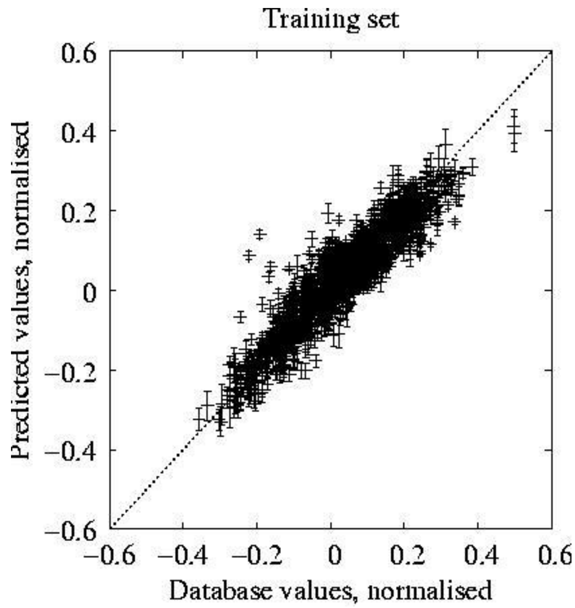



Figure 3.10 (e) **Predicted normalized EL.** vs **Measured normalized EL. (Training Dataset)**

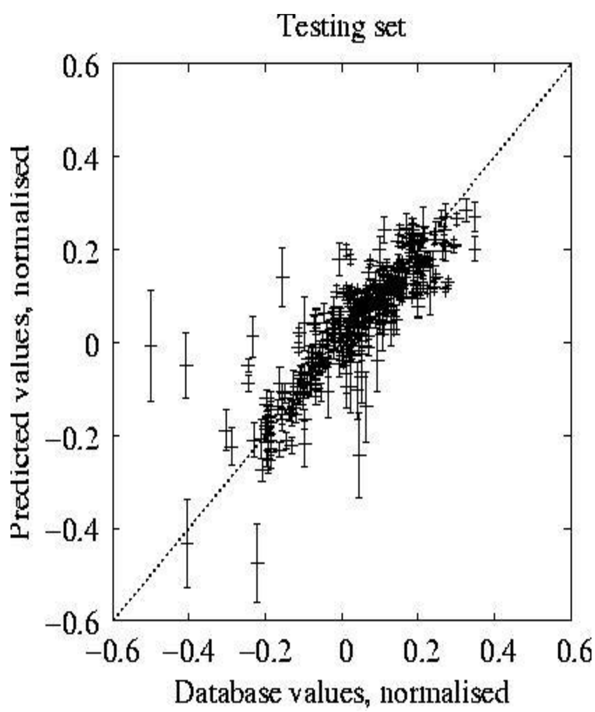



Figure 3.10 (f) **Predicted normalized EL**. vs **Measured normalized EL.** (Test Dataset)

Figure 3.10 (a,b,c,d,e,f) **Elongation (EL) model features**.

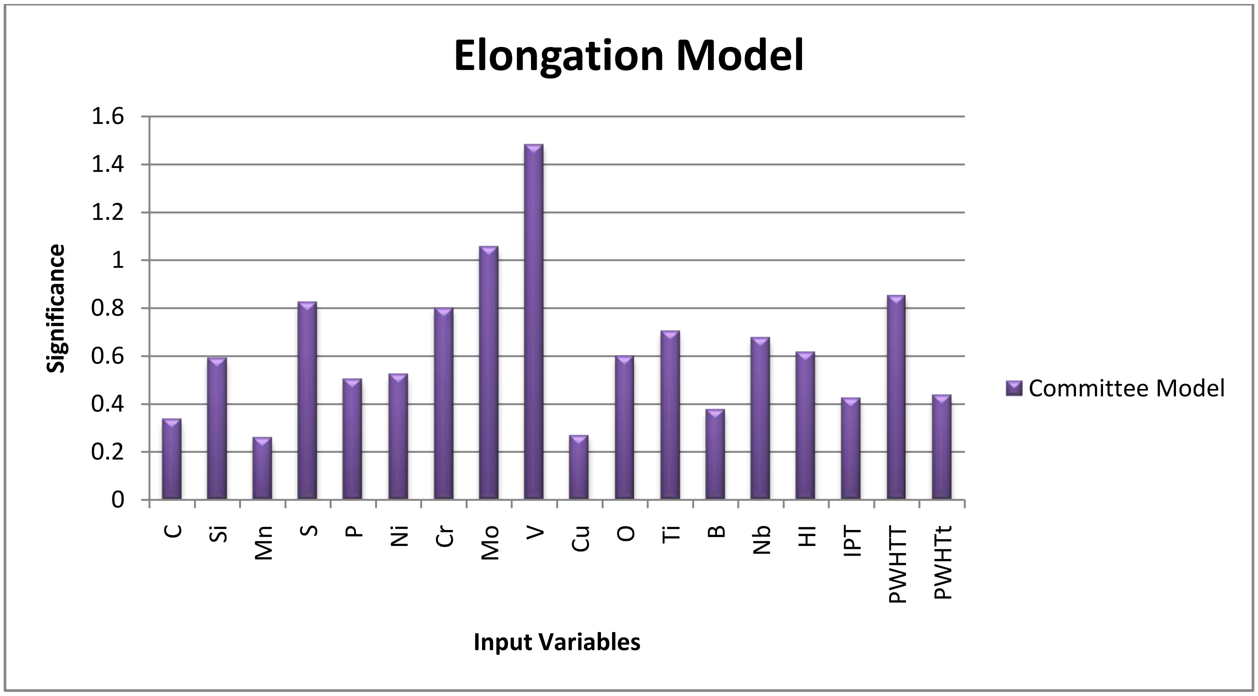



Figure 3.11 The perceived significance $\sigma w$ value of best two **Elongation** models for each of the inputs.

Fig. 3.11 indicates the significance σw of each of the input variables, as perceived by first Two neural network models in the committee. The σw value represents the extent to which a particular input explains the variation in the output, rather like a particular correlation coefficient in linear regression analysis. The Vanadium on the whole explains a large proportion of variation in the Elongation Figure. 3.11. All variables considered are found to have a significant effect on the output indicating a good choice of inputs.

### 3.3.3 Comparision of Neural network models and procedure (MLP, RBF, GRNN)

1 Data of elongation were collected and plotted in the form of Scatter plots.

2. Data prepared according to the file format required to run in Neural Network Softwares.

(.csv format for Linux base software NeuroMat. And .sta format for Statistica Software)

3. Data were randomly divided into three parts (70% training dataset, 20% validation dataset and 10 % testing dataset). (Training dataset: this data set is used to adjust the weights on the neural network. Validation dataset: this data set is used to minimize overfitting. Testing dataset: this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.)

4. Data of Ferritic Steel Weld's Elongation 1827 run in Statistica Software, which were set in its hyperparameter or Neural network architecture (Software converts the Raw Data into Normalized condition, i.e. it can convert into a specified range like 0 to 1 etc.. for processing) in Chapter 2.11

Neural network architecture was set in Statistica software for MLP, RBF and GRNN :

**MLP 17:17-10-1:1 Algorithms : BP100,CG20,CG18b**

**RBF 17:17-530-1:1 Algorithms : SS,KN,PI**

**GRNN 17:17-1061-2-1:1 Algorithms : SS**

BP Back propagation, CG Conjugate gradient descent, SS (sub) sample, KN K-nearest neighbor (deviation assignment), PI Pseudo-invert (linear least squares), b Best network (the network with lowest selection error in the run was restored)

**A neural network's architecture is of form I:N-N-N:O, where I is the number of input variable, O the number of output variables, N the number of units in each layer.**

5. The data of Ferritic Steel Weld's Elongation 1827 run in Statistica Software with above Neural network architecture for best Neural Network model in all three MLP, RBF and GRNN.

6. For best model, the data of Ferritic Steel Weld's Elongation 1827 run in Statistica Software repeatedly hundred of times and finalise the best Neural Network model with smallest training error in all three MLP, RBF and GRNN.

7. The neural network model with the smallest training error was the GRNN model.

Table 3.8 shows the comparision of selected Neural Network models on the basis of their Training Errors. The GRNN models have lowest Training Errors for Elongation of Feritic Steel Welds. The GRNN models are selected for modeling from three basic neural network methods (MLP, RBF, GRNN). ). Statistica 7.1 software is used for MLP, RBF and GRNN.

Table 3.8 Comparision of Neural network models (MLP, RBF, GRNN)

| **Elongation Models** | | | | |
|---|---|---|---|---|
| **MLP** | Train Error | Test Error | Training/Members | Remarks |
| **MLP 18:18-15-10-5-1:1 (Model:No.11)** | **0.056027** | **0.071757** | **BP100,CG462b** | **3 H layers** |
| **MLP 18:18-29-1:1 (Model:No.11)** | **0.05845** | **0.076457** | **BP100,CG498b** | **1 H layer** |
| **MLP 18:18-8-1:1 (Model:No.43)** | **0.056123** | **0.191787** | **DD100,LM187b** | **1 H layer** |
| **MLP 18:18-11-7-1:1 (Model:No.15)** | **0.061902** | **0.077359** | **BP100,CG475b** | **2 H layer** |
| **Elongation Models** | | | | |
| **RBF** | Train Error | Test Error | Training/Members | Remarks |
| **RBF 18:18-193-1:1 (Model:No.46)** | **0.0787** | **0.1125** | **SS,EX,PI** | **1 H layer** |
| **Elongation Models** | | | | |
| **GRNN** | Train Error | Test Error | Training/Members | Remarks |
| **GRNN 18:18-915-2-1:1 (Model:No.01)** | 0.010208 | 0.123726 | **SS** | **2 H layer** |
| **Note: See Appendix C for Profile String of Statistica Neural Network Software** | | | | |

## 3.3.4 Best GRNN Model for Elongation

The normal behaviour of the Predicted Elongation and Observed Elongation are observed in the Figure. 3.12 for Training data, Validation data and Testing data. Training of the model is excellent by GRNN method.

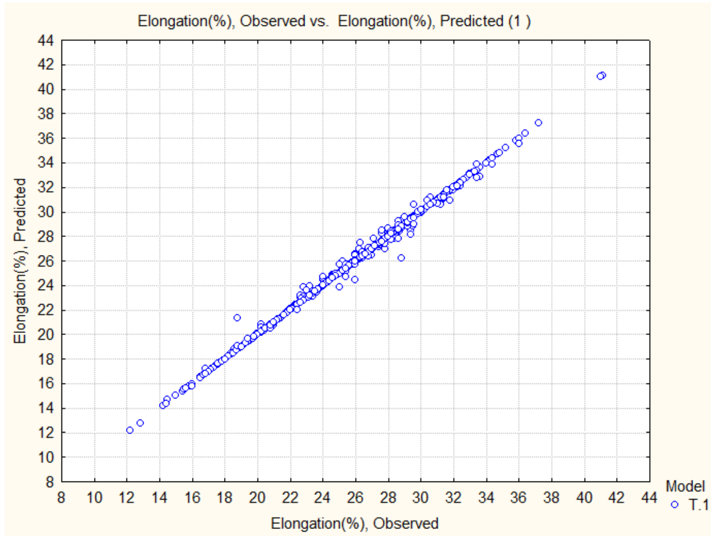


Figure a Training Data for GRNN model of Elongation

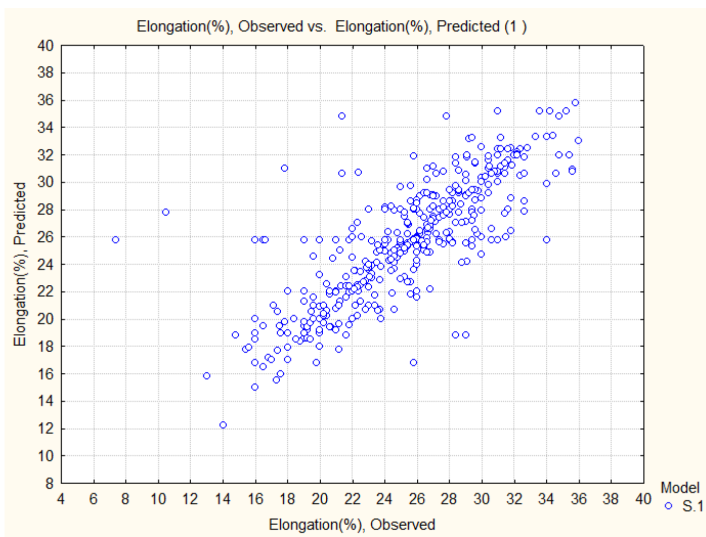


Fig b Validation Data for GRNN model of Elongation

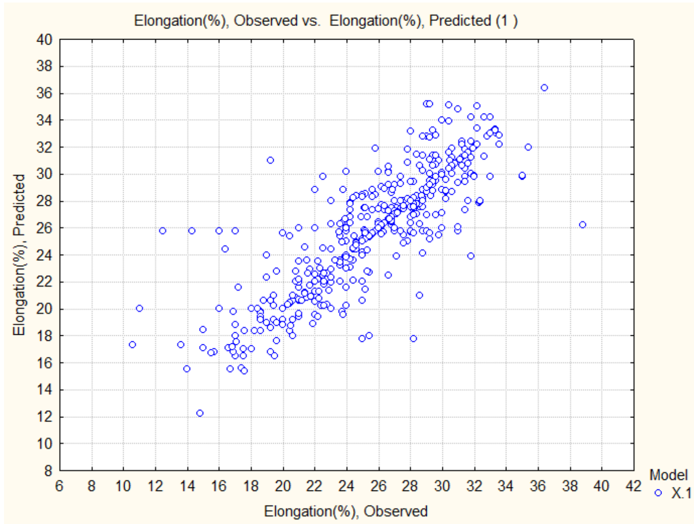


Fig c Test Data for GRNN model of Elongation

Figure 3.12 Training data, validation data and test data of the Best GRNN model for Elongation

The best model of GRNN has training error 0.010208, validation error (selection error) 0.134319 and testing error 0.123726. This model is used for getting the results in form of various response graphs to understand the trend between the input variables and output variable (Elongation). (Figure 4.8)

Table 3.9. Comparison of Significance of Best Trained Models of Elongation

| Input Variables | Significance GRNN Model | Significance BNN Model |
|---|---|---|
| Carbon(wt%) | 7 | 16 |
| Silicon(wt%) | 12 | 10 |
| Manganese(wt%) | 2 | 18 |
| Sulphur(wt%) | **17** | 4 |
| Phosphorus(wt%) | 15 | 12 |
| Nickel(wt%) | 1 | **11** |
| Chromium(wt%) | **3** | 5 |
| Molybdenum(wt%) | 6 | 2 |
| Vanadium(wt%) | 11 | 1 |
| Copper(wt%) | **18** | 17 |
| Oxygen(ppm) | 16 | 9 |
| Titanium(ppm) | 5 | 6 |
| Boron(ppm) | 13 | 15 |
| Niobium(ppm) | 8 | 7 |
| Heat_input(kJ.mm-1) | 10 | 8 |
| Interpass_temperature(C) | 9 | 14 |
| Postweld_heat_treatment_temperature(C) | 4 | 3 |
| Post-weld_heat_treatment_time(h) | 14 | 13 |

Table 3.9 shows the comparison of Significance of the GRNN and BNN models. Number 1 indicates highest value of significance and Number 18 lowest value of significance. Most of the Input Variables are closer in significance for both the models. All input variables considered are found to have a significant effect on the output indicating a good selection of inputs.

### 3.3.5 Neural Network and Genetic Algorithms Modeling for Elongation of Ferritic Steel Welds

#### 3.3.5.1 Genetic Algorithms parameters and procedure

A genetic algorithm has been developed in language C considering the following parameters:

Number of populations = 3

Number of generations = 3000

Population size = 20 chromosomes

When a new generation is created, the following steps are followed: after ranking the 20 chromosomes according to their scores, the first chromosome is copied without change. The chromosomes 2 to **19** are recombined with each others. One gene of one of these chromosomes is mutated between ±0.2%. The chromosome 20, with the worst score, is killed and a new random chromosome is generated and incorporated in the new population.

This program can calculate the best set ( $x_1$ ,$x_2$, ... ,*x* j) of input parameters for a desired output *y*, which is in this study, the elongation of ferritic steel welds, for which a Bayesian neural network model was developed[32].

The steps for Genetic Algorithms Modelling:

• First, all the files related to the neural network created for the elongation were put in the folder "gacode" to optimise. These files were the following:

generate44.exe
norm_test.in
_w*f
*.lu
spec1.t1
outran.x
MINMAX

• Then, the labels of the inputs variables of the neural network were written in the "labels.tct" file
• Then the all inputs variables were define in the "values" file to vary.
• Then, the desired target value of elongation was normalised and entered it in the "nninput" file, as well as the wanted accuracy.

- Finally, the C program "ga_code" was compiled and executed.

After execution of the Genetic Algorithms program, the output was the values of 18 input variables for given target value of the elongation of Ferritic Steel Weld. The calculation time was in hours.

Three different target values of Ferritic Steel Weld's elongation were taken and Genetic Algorithms programs were run. The outputs were given in result and discussion Chapter 4.

# 3.4 Charpy Toughness Models

## 3.4.1 Experimental Data base

All of the data collected are from multi run weld deposits in which the joint is designed to minimize dilution from the base metal, to enable specifically the measurement of all weld metal properties. Furthermore, they all represent electric arc welds made using one of the following processes: manual metal arc (MMAW), submerged arc welding (SAW) and tungsten inert gas (TIG). The welding process itself was represented only by the level of heat input. This is because a large number of published papers did not specify welding parameters in sufficient detail to enable the creation of a dataset without missing values. Missing values cannot be tolerated in the method used here. If the effect of a welding process is not properly represented by the heat input and chemical composition, then neglect of any important parameters will make the predictions more ‘noisy'. As discussed below, the noise in the output was found to be acceptable; a greater uncertainty arises from the lack of a uniform coverage of the input space. The data were collected from a large number of sources [33] to [76].

The aim of the neural network analysis was to predict the Charpy Toughness as a function of a large number of variables, including the chemical composition, the welding heat input and any heat treatment. The Charpy Toughness database consists of 3449 separate experiments. Neural network methods used in this work cannot cope with missing values of any of the variables.

### 3.4.1.1 Charpy Toughness Database

Table 3.10 shows the range, mean and standard deviation of each variable including the output (charpy toughness). The purpose here is simply to list the variables and provide an idea of the range covered. It is emphasized however, that unlike linear regression analysis, the information in Table 3.10 cannot be used to define the range of applicability of the neural network model. This is because the inputs are in general expected to interact. We shall see later that it is the Bayesian framework of our neural network analysis which allows the calculation of error bars which define the range of useful applicability of the trained network. A visual impression of the spread of data is shown in Fig. 3.13. It can be concluded from Figure. 4.10 (a to t) and Figure. 4.11(a to t) that the effect of Carbon, Silicon, Manganese, Sulphur, Phosphorus, Nickel, Chromium, Molybdenum, Vanadium, Copper, Oxygen, Titanium, Nitrogen, Boron, Niobium, Heat input, Interpass temperature, Post- weld heat treatment temperature, Post-weld heat treatment time and Testing Temperature Charpy Toughness on the Charpy Toughness of Ferritic Steel Welds have been systematically studied by BNN and GRNN.[28]

It can be concluded from Figure.4.12.1 to 4.12.18 that the effect in combination of any two input variables (Independent variables) from Carbon, Silicon, Manganese, Sulphur, Phosphorus, Nickel, Chromium, Molybdenum, Vanadium, Copper, Oxygen, Titanium, Nitrogen, Boron, Niobium, Heat input, Interpass temperature, Post- weld heat treatment temperature, Post-weld heat treatment time and Testing Temperature Charpy Toughness on the Charpy Toughness of Ferritic Steel Welds have been systematically studied by GRNN.

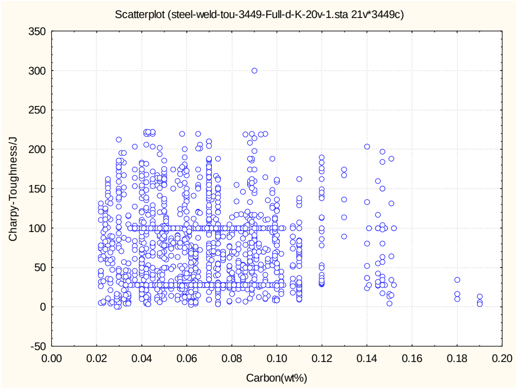
The predictions of the all Input variables for Targeted Charpy Toughness by Genetic Algorithms are given in Table 4.13. These can be useful for the design of the Ferritic Steel Welds. Genetic Algorithms can design the Ferritic Steel Welds by extrapolation beyond the existing data.

Table 3.10 shows the range, mean and standard deviation of each variable including the output (Charpy Toughness).

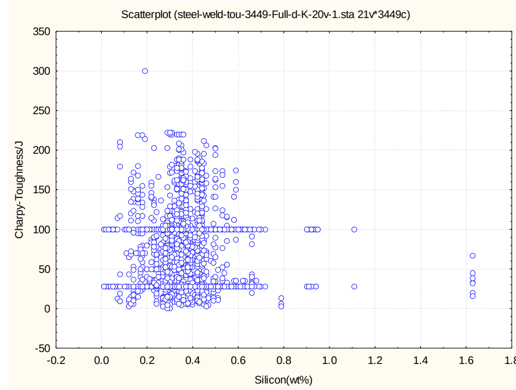
Table 3.10 The Input Variables for **Charpy Toughness** Model. “p.p.m .’ corresponds to parts per million by weight.

| Variables | Min | Max | Average | StDev |
|---|---|---|---|---|
| C wt% | 0.022 | 0.19 | 0.0692 | 0.0207 |
| Si wt% | 0.01 | 1.63 | 0.3527 | 0.1214 |
| Mn wt% | 0.23 | 2.31 | 1.2209 | 0.4446 |
| S wt% | 0.002 | 0.14 | 0.0078 | 0.0075 |
| P wt% | 0.003 | 0.25 | 0.0101 | 0.0128 |
| Ni wt% | 0 | 10.8 | 0.9933 | 2.26 |
| Cr wt% | 0 | 11.78 | 0.4406 | 1.323 |
| Mo wt% | 0 | 1.54 | 0.1818 | 0.3341 |
| V wt% | 0 | 0.53 | 0.0139 | 0.0404 |
| Cu wt% | 0 | 2.18 | 0.0638 | 0.2128 |
| O ppm | 63 | 1535 | 399.6638 | 110.6312 |
| Ti ppm | 0 | 770 | 96.0337 | 132.9401 |
| N ppm | 0 | 979 | 77.5725 | 60.8648 |
| B ppm | 0 | 200 | 13.1739 | 33.4533 |
| Nb ppm | 0 | 1770 | 37.6917 | 133.0933 |
| HI kJ mm-1 | 0.6 | 6.6 | 1.1954 | 0.6596 |
| IPT C | 20 | 350 | 199.0003 | 31.0232 |
| PWHTT C | 0 | 760 | 186.1773 | 249.8889 |
| PWHTt h | 0 | 100 | 3.3429 | 6.6257 |
| TTCT K | 77 | 409 | 227.8425 | 38.3343 |
| Charpy Toughness J | 0 | 300 | 72.714 | 42.8411 |

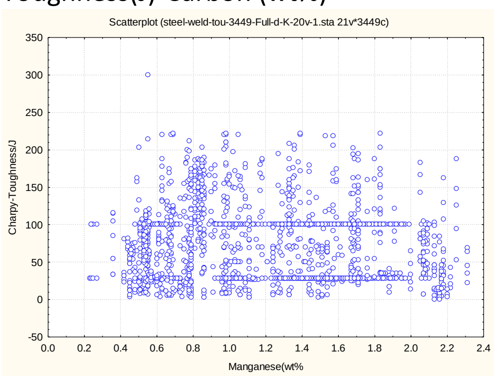
# Scatter Plots of Toughness-Data-3449



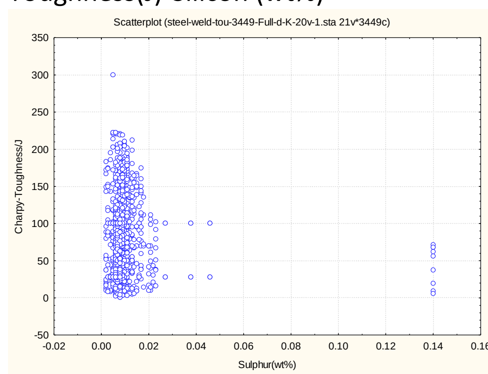

Toughness(J)-Carbon (wt%)



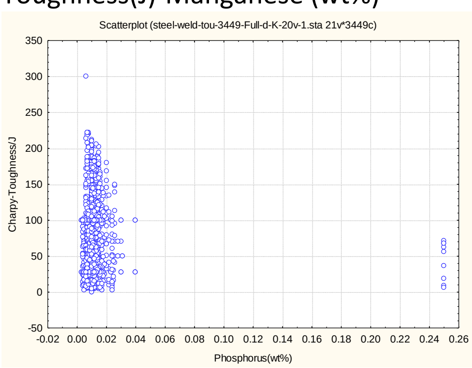

Toughness(J)-Silicon (wt%)



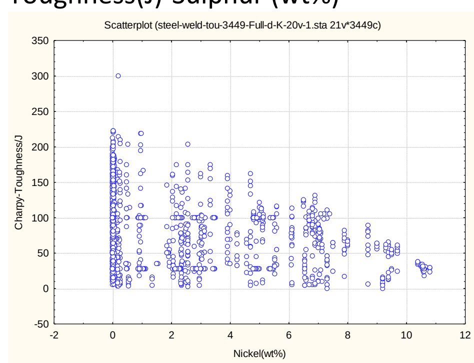

Toughness(J)-Manganese (wt%)



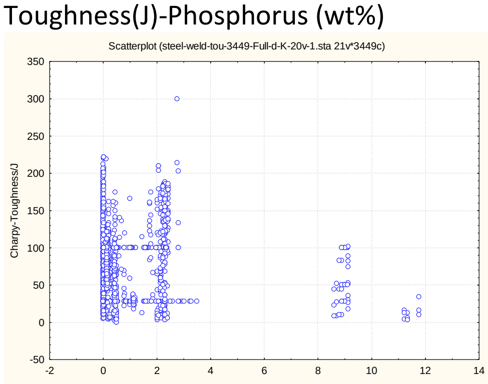

Toughness(J)-Sulphur (wt%)



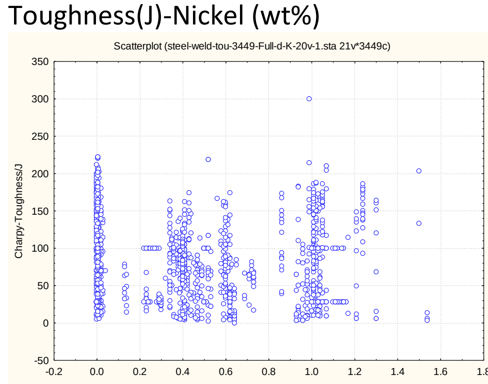

Toughness(J)-Phosphorus (wt%)



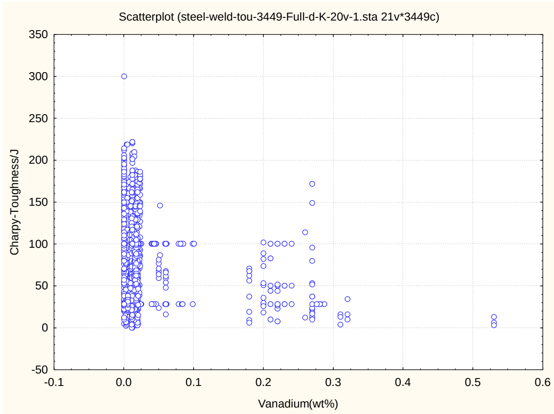

Toughness(J)-Nickel (wt%)



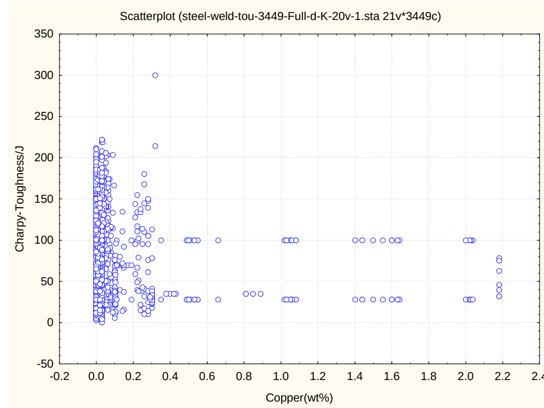

Toughness(J)-Chromium (wt%)



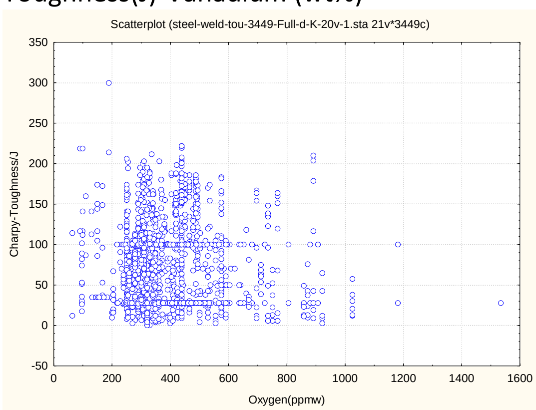

Toughness(J)-Molybdenum (wt%)



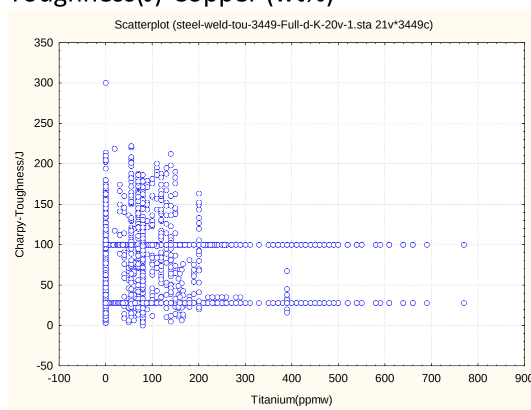

Toughness(J)-Vanadium (wt%)



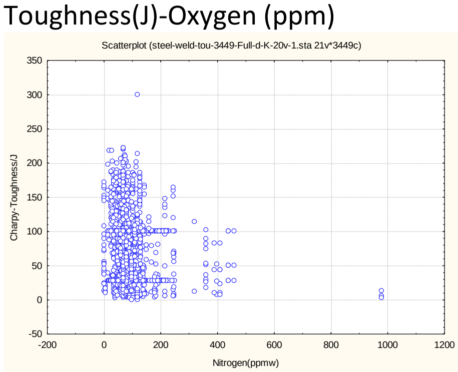

Toughness(J)-Copper (wt%)



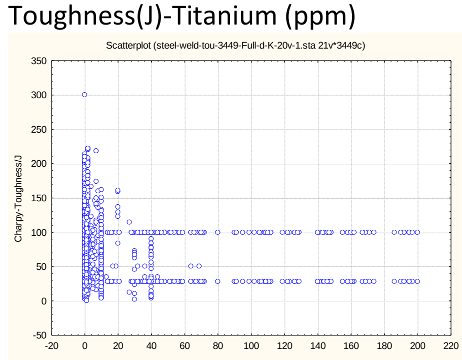

Toughness(J)-Oxygen (ppm)



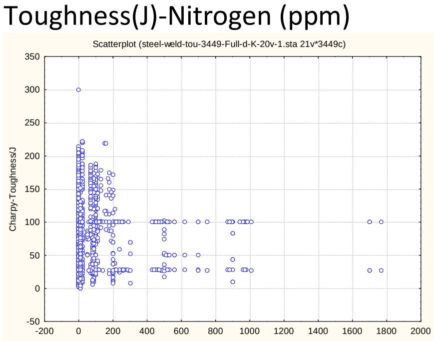

Toughness(J)-Titanium (ppm)



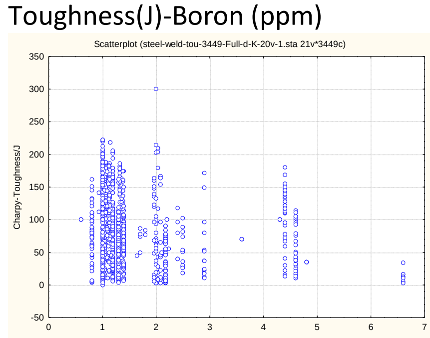

Toughness(J)-Nitrogen (ppm)



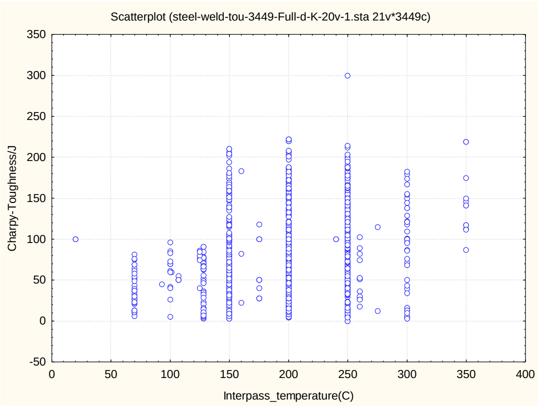

Toughness(J)-Boron (ppm)



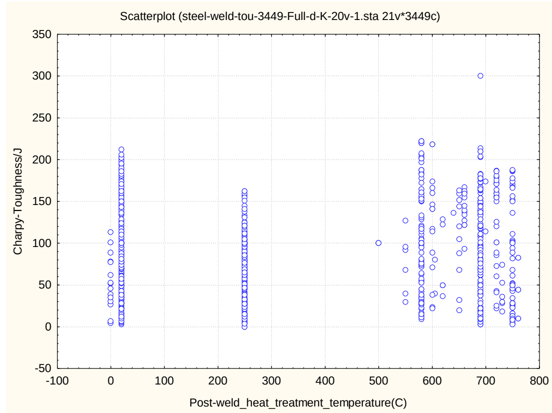

Toughness(J)-Niobium (ppm)



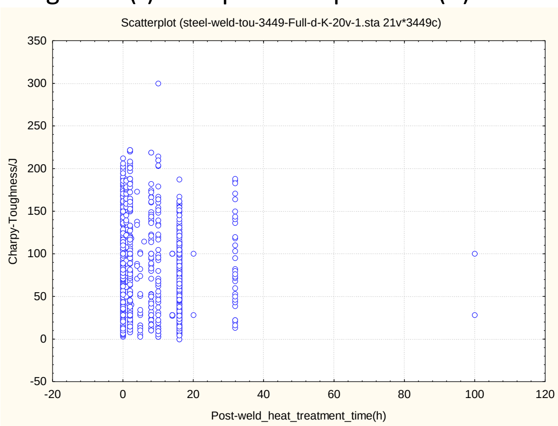

Toughness(J)-Heat Input(KJ mm-1)



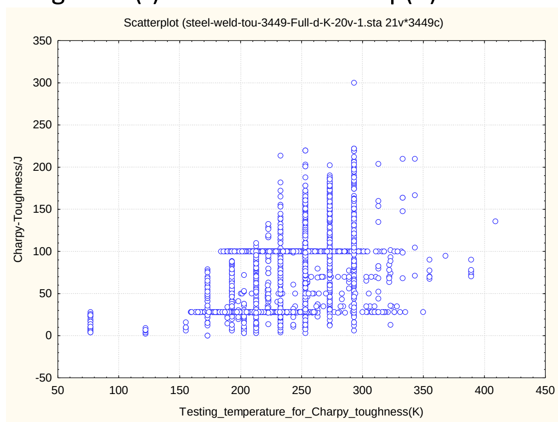

Toughness(J)-Interpass Temperature(C)




Toughness(J)-Post Weld HT Temp(C)




Toughness(J)-Post Weld HT Time(h)




Toughness(J)-Testing Temp-Charpy toughness(K)

Figure 3.13 Database distribution used for **Charpy Toughness** model. “p.p.m .’ corresponds to parts per million by weight.

## 3.4.2 Neural Network Models for Charpy Toughness

### 3.4.2.1 Bayesian Neural Network Model and procedure

1 Data of charpy toughness were collected and plotted in the form of Scatter plots.

2. Data prepared according to the file format required to run in Neural Network Softwares.

(.csv format for Linux base software NeuroMat. And .sta format for Statistica Software)

3. Data were randomly divided into three parts (70% training dataset, 20% validation dataset and 10 % testing dataset). (Training dataset: this data set is used to adjust the weights on the neural network. Validation dataset: this data set is used to minimize overfitting. Testing dataset: this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.)

4. Data of Ferritic Steel Weld's Charpy Toughness 3449 run in NeuroMat Software, which were set in its hyperparameter or Neural network architecture (Software converts the Raw Data into Normalized condition, i.e. it can convert into a specified range like 0 to 1 etc.. for processing) in Chapter 2.10

Neural network architecture for Bayesian Neural Network was set in NeuroMat Software :

Three layers : **Input Layer**(Input Variables), **Hidden Layer** (Algorithms) and **Output Layer** (Yield Strength)

Algorithms : Bigback 5

5. The data of Ferritic Steel Weld's Charpy Toughness 3449 run in NeuroMat Software with above Neural network architecture for best Neural Network Committee model. The best committee model was decided on the basis of smallest test error of the committee model.

6. For best committee model, the data of Ferritic Steel Weld's Charpy Toughness 3449 run in NeuroMat Software repeatedly hundred of times and finalise the best committee model with smallest test error. (NeuroMat gives in single run set of 100 models which required time in hours. Out of these 100 models, the models in the committee are selected on the basis of smallest test error. The number of models in committee varies everytime with repeatedly running the data in NeuroMat. Thus the selection of committee model with a smallest test error is time consuming.)

Some more than hundred charpy toughness neural network models were trained on a training dataset which consisted of a random selection of 70%of the data 2414 from the charpy toughness dataset. And 20% of the data 690 from charpy toughness data set was used for cross validation.The remaining 345 data formed the test dataset which was used to see how the model generalizes on unseen data. Each model contained the 20 inputs listed in Table 1 but with different numbers of hidden units or the random seeds used to initiate the values of the weights. Fig. 3.14 shows the results. As expected, the perceived level of noise ($\sigma y$) in the normalised charpy toughness decreases as the number of hidden units increases, Fig. 3.14a. This is not the case for the test error, which goes through a minimum at nineteen hidden units, Fig. 3.14b, and for the log predictive error which reaches a maximum at nineteen hidden units, Fig. 3.14c.

The error bars presented throughout this work represent a combination of the perceived level of noise σy in the output and the fitting uncertainty estimated from the Bayesian framework. It is evident that there are a few outliers in the plot of the predicted versus measured Charpy Toughness for the test dataset, Fig. 3.14f. Each of these outliers has been investigated and found to represent unique data which are not represented in the training dataset, Fig. 3.14e.

It is possible that a committee of models can make a more reliable prediction than an individual model (Chapter 2). The best models are ranked using the values of the log predictive errors Fig. 3.14c. Committees are then formed by combining the predictions of the best L models, where L - 1, 2, . . .; the size of the committee is therefore given by the value of L. A plot of the test error of the committee versus its size gives a minimum which defines the optimum size of the committee, as shown in Fig. 3.14d.

The test error associated with the best single model is clearly greater than that of any of the committees Fig. 3.14d. The committee with eigth models was found to have an optimum membership with the smallest test error. The committee was therefore retrained on the entire data set without changing the complexity of any of its member models.

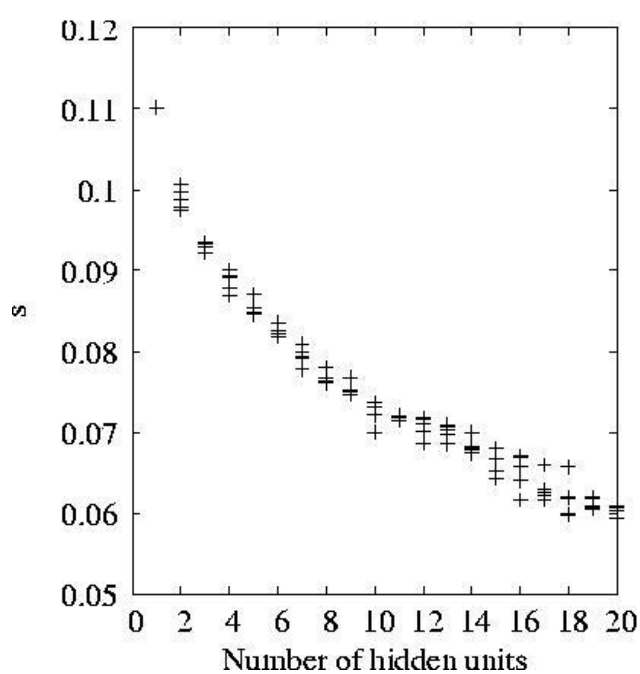


Figure 3.14 (a) $\sigma y$ vs Hidden units

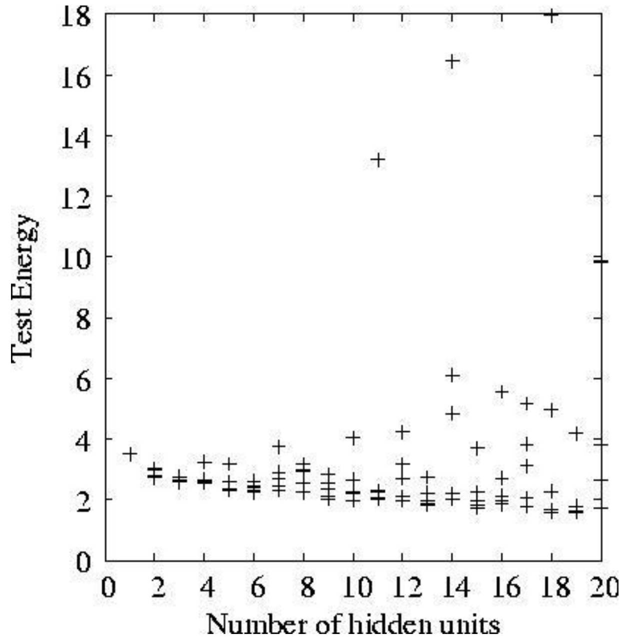



Figure 3.14 (b) Test Error vs Hidden units.

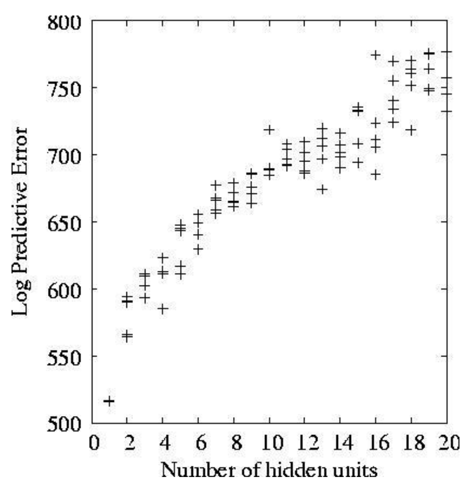



Figure 3.14 (c) Log predictive error vs Hidden units

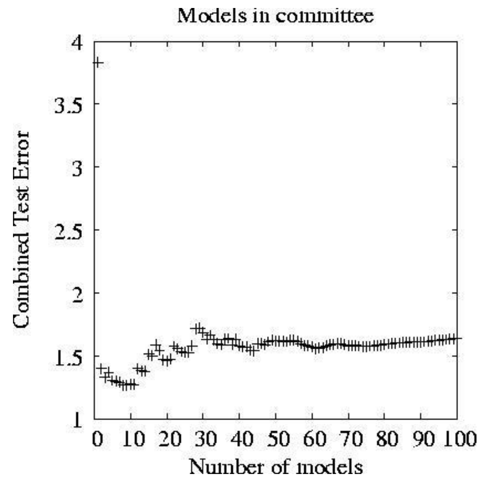



Figure 3.14 (d) Test Error vs Models in committee

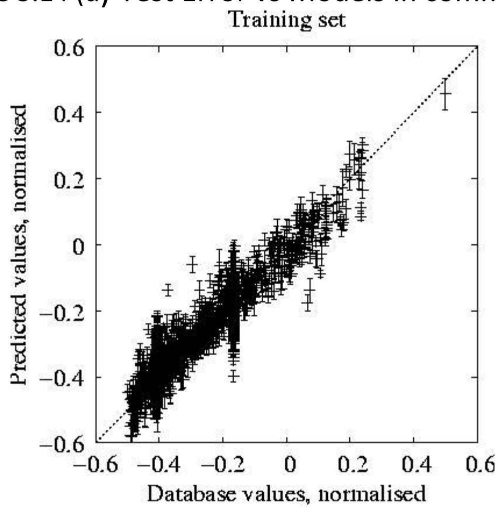



Figure 3.14 (e) Predicted normalized CT. vs Measured normalized CT. (Training Dataset)

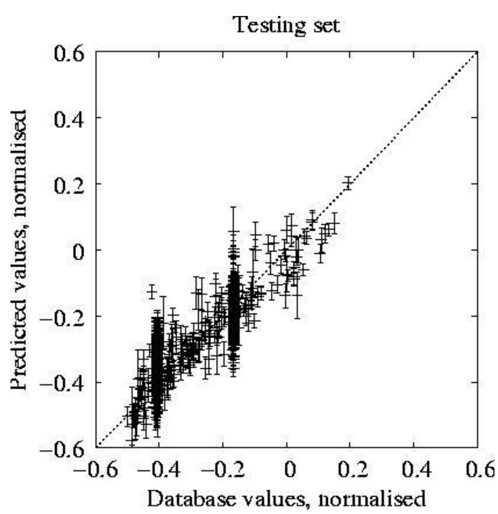



Figure 3.14 (f) Predicted normalized CT. vs Measured normalized CT. (Test Dataset)

Figure 3.14 . (a,b,c,d,e,f) **Charpy Toughness** (CT) model features.

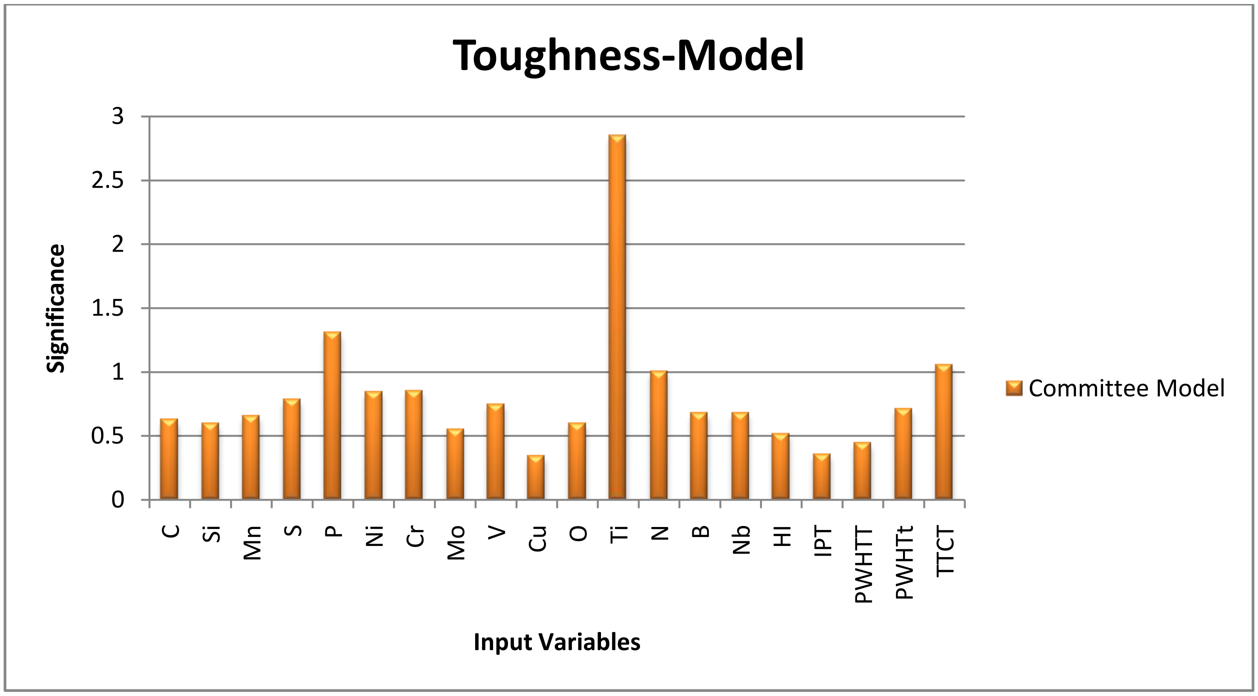



Figure 3.15 The perceived significance $\sigma w$ value of best eight **Charpy Toughness** models for each of the inputs.

Fig. 3.15 indicates the significance $\sigma w$ of each of the input variables, as perceived by first Eight neural network models in the committee. The $\sigma w$ value represents the extent to which a particular input explains the variation in the output, rather like a particular correlation coefficient in linear regression analysis. The Titanium on the whole explains a large proportion of variation in the Charpy Toughness Figure. 3.15. All variables considered are found to have a significant effect on the output indicating a good selection of inputs.

### 3.4.3 Comparisons of Neural network models and procedure (MLP, RBF, GRNN)

1 Data of charpy toughness were collected and plotted in the form of Scatter plots.

2. Data prepared according to the file format required to run in Neural Network Softwares.

(.csv format for Linux base software NeuroMat. And .sta format for Statistica Software)

3. Data were randomly divided into three parts (70% training dataset, 20% validation dataset and 10 % testing dataset). (Training dataset: this data set is used to adjust the weights on the neural network. Validation dataset: this data set is used to minimize overfitting. Testing dataset: this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.)

4. Data of Ferritic Steel Weld's Charpy Toughness 3449 run in Statistica Software, which were set in its hyperparameter or Neural network architecture (Software converts the Raw Data into Normalized condition, i.e. it can convert into a specified range like 0 to 1 etc.. for processing) in Chapter 2.11

Neural network architecture was set in Statistica software for MLP, RBF and GRNN :

**MLP 17:17-10-1:1 Algorithms : BP100,CG20,CG18b**

**RBF 17:17-530-1:1 Algorithms : SS,KN,PI**

**GRNN 17:17-1061-2-1:1 Algorithms : SS**

BP Back propagation, CG Conjugate gradient descent, SS (sub) sample, KN K-nearest neighbor (deviation assignment), PI Pseudo-invert (linear least squares), b Best network (the network with lowest selection error in the run was restored)

**A neural network's architecture is of form I:N-N-N:O, where I is the number of input variable, O the number of output variables, N the number of units in each layer.**

5. The data of Ferritic Steel Weld's Charpy Toughness 3449 run in Statistica Software with above Neural network architecture for best Neural Network model in all three MLP, RBF and GRNN.

6. For best model, the data of Ferritic Steel Weld's Charpy Toughness 3449 run in Statistica Software repeatedly hundred of times and finalise the best Neural Network model with smallest training error in all three MLP, RBF and GRNN.

7. The neural network model with the smallest training error was the GRNN model.

Table 3.11 shows the comparisons of selected Neural Network models on the basis of their Training Errors. The GRNN models have lowest Training Errors for Charpy Toughness of Feritic Steel Welds. . The GRNN models are selected for modeling from three basic neural network methods (MLP, RBF, GRNN). Statistica 7.1 software is used for MLP, RBF and GRNN.

Table 3.11 Comparisons of Neural network models (MLP, RBF, GRNN)

| **Charpy Toughness Models** | | | | | |
|---|---|---|---|---|---|
| SR No. | **MLP** | Train Error | Test Error | Training/Members | Remarks |
| 1 | **MLP 20:20-11-1:1 (Model:No.7)** | **0.090335** | **0.096968** | **BP100,CG289b** | **1 H layer** |
| 2 | **MLP 20:20-14-8-1:1 (Model:No.8)** | **0.085442** | **0.093736** | **BP100,CG488b** | **2 H layer** |
| 3 | **MLP 20:20-14-8-10-1:1 (Model:No.3)** | **0.080723** | **0.091685** | **BP100,CG499b** | **3 H layer** |
| **Charpy Toughness Models** | | | | | |
| SR No. | **RBF** | Train Error | Test Error | Training/Members | Remarks |
| 1 | **RBF 20:20-862-1:1 (Model:No.5)** | **0.07513** | **0.013794** | **SS,KN,PI** | **1 H layer** |
| **Charpy Toughness Models** | | | | | |
| SR No. | **GRNN** | Train Error | Test Error | Training/Members | Remarks |
| 1 | **GRNN 20:20-1725-2-1:1 (Model:No.8)** | **0.011510** | **0.0198988** | **SS** | **2 H layer** |
| 2 | **GRNN 20:20-1725-2-1:1 (Model:No.16)** | **0.011953** | **0.018632** | **SS** | **2 H layer** |
| **3** | **GRNN 20:20-1725-2-1:1 (Model:No.7)** | **0.011404** | **0.018669** | **SS** | **2 H layer** |
| **Note: See Appendix C for Profile String of Statistica Neural Network Software** | | | | | |

## 3.4.4 Best GRNN Model for Charpy Toughness

The normal behavior of the Predicted Charpy Toughness and Observed Charpy Toughness are observed in the Figure. 3.16 for Training data, Validation data and testing data. Training of the model is excellent by GRNN method.

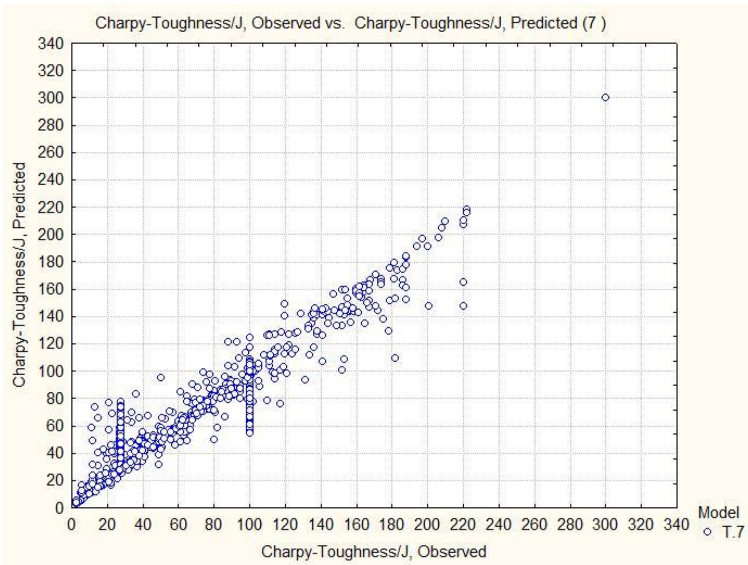



Figure a Training Data for GRNN model of Charpy Toughness

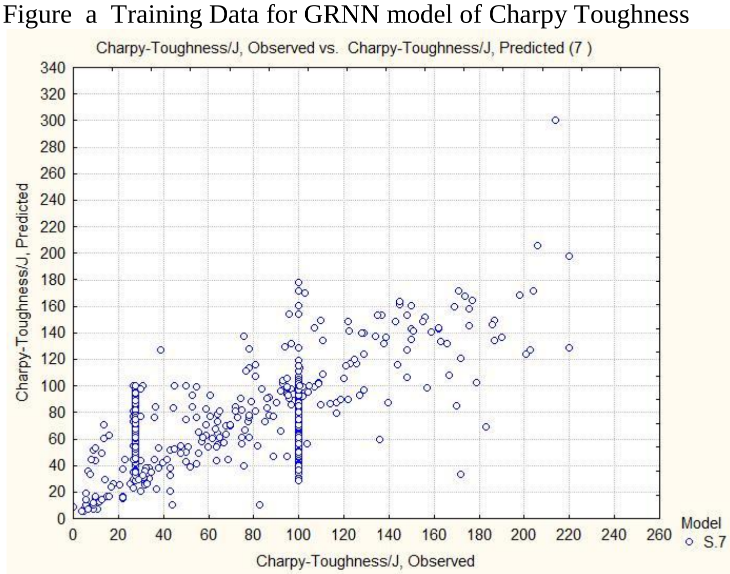



Fig b Validation Data for GRNN model of Charpy Toughness

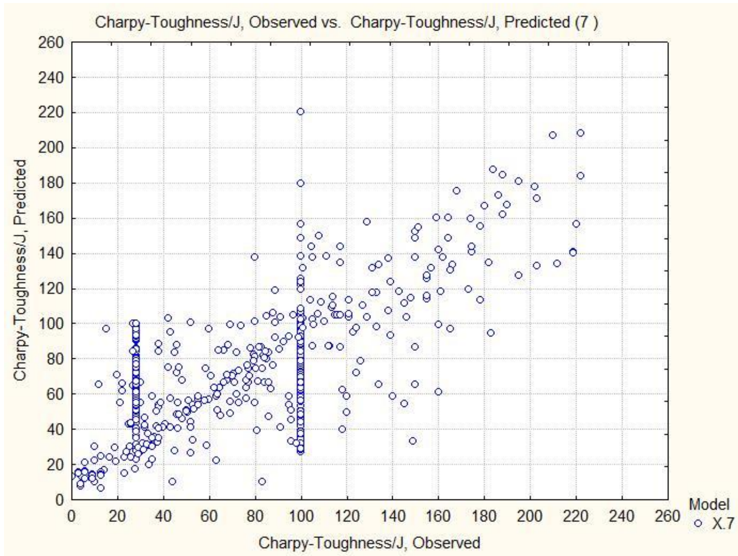



Fig c Test Data for GRNN model of Charpy Toughness

Figure 3.16 (a to c) Training data, validation data and test data of the Best GRNN model for Charpy Toughness.

The best model of GRNN has training error 0.011404, validation error (selection error) 0.018101, and testing error 0.018669. This model is used for getting the results in form of various response graphs to understand the trend between the input variables and output variable (Charpy Toughness).(Figure 4.11 (a to t)).

Table 3.12 Comparison of Significance of Best Trained Models of Elongation

| Input Variables | Significance GRNN Model | Significance BNN Model |
|---|---|---|
| Carbon(wt%) | 7 | 13 |
| Silicon(wt%) | 19 | 14 |
| Manganese(wt%) | 6 | 12 |
| Sulphur(wt%) | 18 | 7 |
| Phosphorus(wt%) | 16 | 2 |
| Nickel(wt%) | 4 | 6 |
| Chromium(wt%) | 10 | **5** |
| Molybdenum(wt%) | 5 | 16 |
| Vanadium(wt%) | 3 | 8 |
| Copper(wt%) | 15 | 20 |
| Oxygen(ppm) | 8 | 15 |
| Nitrogen(ppm) | 14 | 4 |
| Titanium(ppm) | 17 | 1 |
| Boron(ppm) | 20 | 11 |
| Niobium(ppm) | 12 | 10 |
| Heat_input(kJ.mm-1) | 11 | 17 |
| Interpass_temperature(C) | 9 | 19 |
| Postweld_heat_treatment_temperature(C) | 2 | **18** |
| Post-weld_heat_treatment_time(h) | 13 | 9 |
| Testing Temperature CT (K) | 1 | 3 |

Table 3.12 shows the comparison of Significance of the GRNN and BNN models. Number 1 indicates highest value of significance and Number 20 lowest value of significance. Most of the Input Variables are not closer in significance for both the models. All input variables considered are found to have a significant effect on the output indicating a good selection of inputs.

### 3.4.5 Neural Network and Genetic Algorithms Modelling for Charpy Toughness of Ferritic Steel Welds

### 3.4.5.1 Genetic Algorithms parameters and procedure

A genetic algorithm has been developed in language C considering the following parameters:

Number of populations = 3

Number of generations = 3000

Population size = 20 chromosomes

When a new generation is created, the following steps are followed: after ranking the 20 chromosomes according to their scores, the first chromosome is copied without change. The chromosomes 2 to 19 are recombined with each others. One gene of one of these chromosomes is mutated between ±0.2%. The chromosome 20, with the worst score, is killed and a new random chromosome is generated and incorporated in the new population.

This program can calculate the best set ( x1 ,x2, ... ,x j) of input parameters for a desired output y, which is in this study, the charpy toughness of ferritic steel welds, for which a Bayesian neural network model was developed[32].

The steps for Genetic Algorithms Modelling:

• First, all the files related to the neural network created for the charpy toughness were put in the folder "gacode" to optimise. These files were the following:

generate44.exe
norm_test.in
_w*f
*.lu
spec1.t1
outran.x
MINMAX

• Then, the labels of the inputs variables of the neural network were written in the "labels.tct" file
• Then the all inputs variables were define in the "values" file to vary.
• Then, the desired target value of charpy toughness was normalised and entered it in the "nninput" file, as well as the wanted accuracy.

- Finally, the C program "ga_code" was compiled and executed.

After execution of the Genetic Algorithms program, the output was the values of 20 input variables for given target value of the charpy toughness of Ferritic Steel Weld. The calculation time was in hours.

Three different target values of Ferritic Steel Weld’s charpy toughness were taken and Genetic Algorithms programs were run. The outputs were given in result and discussion Chapter 4.