Training Machine Learning Algorithms to Laboratory Triaxial Ultrasonic Velocity Data for Utah FORGE Well 16A(78)-32

Report documenting completion of Milestone 2.2.1 of Utah FORGE project 2439: A Multi-Component Approach to Characterizing In-Situ Stress at the U.S. DOE FORGE EGS Site: Laboratory, Modeling and Field Measurement

Ayyaz Mustafa[a], Mark Kelley[c], Andrew Bunger[a,b,*]

a Department of Civil and Environmental Engineering, University of Pittsburgh, Pittsburgh, PA, USA

b Department of Chemical and Petroleum Engineering, University of Pittsburgh, Pittsburgh, PA, USA

c Battelle Memorial Institute, Columbus, OH, USA

*Corresponding author: bunger@pitt.edu

Last Updated: 19 June 2023

Table of Contents

List of Figures

List of Tables

1. Abstract

Three machine learning (ML) predictive models were developed for the prediction of vertical and two orthogonally oriented horizontal stresses in the Utah FORGE well 16A(78)-32. The ML models were trained using laboratory-based triaxial ultrasonic wave velocity (labTUV) data wherein wave velocities were measured with various combinations of true triaxial applied stress. The ultrasonic velocities data include compressional, fast shear, and slow shear velocities in each of three directions for a total of nine velocities for each stress combination. However, because the ultimate goal is to deploy the trained model for interpretation of field sonic log data where only the vertically-propagating waves are measured, the work here focuses on just the wave velocities with vertical (z-direction) propagation. Also, because vertical (overburden) is often well constrained, one approach explored here is to take the vertical stress also as known and train the model to predict the two horizontal stresses.

Feed forward artificial neural network (FFNN) has been applied to build three data-driven ML predictive models for vertical and two horizontal stresses in the Granitoid formation from well 16A(78)-32. Prior to the execution of FFNN algorithm, the available data was thoroughly analyzed through exploratory data analysis (EDA) in order to understand the data distribution, relationship between input and output features, and to identify the relative importance of input features with respect to output. EDA was performed using different libraries of open-source software *Python (ver:3.9.13)* such as *Pandas, Seaborn, Matplotlib,* and *SciPy* executed on *Spyder IDE (ver:5.2.2)*. The FFNN algorithm was implemented using *Deep Learning Toolbox* of *MATLAB 2022* software and optimized by tuning different hyperparameters such as neurons count in hidden layers, training function, activation function, and realization count in order to obtain reliable and consistent prediction outcomes. The ML prediction model is converted to empirical mathematical correlation which can be used to estimate vertical and horizontal stresses provided that the same input variables and their ranges are used. All three developed ML models demonstrated reliable and consistent prediction performance for the vertical and two horizontal stresses in terms of low root means squared error (RMSE) and high coefficient of correlation (R). The first ML model for vertical stress (σ_z) exhibited the training (model building) and testing (validation) RMSE error of 3.1 and 3.5, respectively. The R value for training and testing was found to be 0.975 and 0.978 for the vertical stress (σ_z) model. In the case of the first horizontal stress (σ_x) model, the 'R' values were determined to be 0.965 and 0.923 for the testing and training results. The excellent model functioning was reflected by the low RMSE errors for testing (4.4) and training (6.3) prediction, respectively. Likewise, the second horizontal stress model (σ_y) demonstrated the 'R' value of 0.95 for testing and 0.934 for training prediction. The prediction RMSE errors were observed to be 4.2 and 4.8 for training and testing and training datasets, respectively. In summary, accuracy measures revealed excellent prediction performance of all three ML models for stress prediction. The proposed ML models were able to capture the stress variation trends for testing and validation dataset to provide reliable and robust stress prediction. An explicit mathematical correlation can be confidently applied for stress estimation using the same input variables and ranges for which the model was developed.

2. Task and Milestone Description

This report documents the task completion and technical accomplishments comprising achievement of Milestone 2.2.1, as per the project SOPO. The validation of milestone accomplishments is illustrated in Figures 14-18 demonstrating the key outcomes of the developed machine learning models for vertical and horizontal stresses.

Milestone 2.2.1 – the task encompasses three data-driven machine learning (ML) models developed for the prediction of vertical and two orthogonally oriented horizontal stresses (x, y, and z-directions) in Granitoid Formation using the experimental dataset. The experimental data was generated using the core samples of Granitoid Formation retrieved from the well 16A(78)-32. The detailed description about the experimental work was documented in Milestone report 2.1.1 of the project.

The first ML model was developed for vertical stress 'σ_z' (z-direction) using three velocity components in vertical direction as input features including one compressional (P) and two shear (S) wave slownesses. Note that slowness is adopted here for convenience because field sonic logs report slowness, which is just the inverse of the velocity. There are two vertically-propagating S-wave slownesses corresponding to waves polarized in x and y directions, respectively.

The second ML model was developed for horizontal stress in x-direction 'σ_x' using four input features including P-wave, fast S-wave, slow S-wave slowness, and stress in vertical direction. The third model was built for horizontal stress in y-direction 'σ_y' using the same four input features, i.e., P-wave, fast S-wave, slow S-wave slowness, and stress in vertical direction. By considering both x- and y-directional stresses the validation of the trained model shows it is capable, in principle, of predicting both minimum and maximum horizontal stress magnitudes using field sonic logging data and overburden stress as constrained by integrating the density log.

Three ML prediction models were developed using feed forward neural network (FFNN) technique. A total of 46 data points were utilized for the prediction modelling. Dataset was subjected to routine exploratory data analyses (EDA) techniques prior to the ML model execution such as statistical parameters, histograms, correlation coefficients using Pearson, Spearman, and Kendall criteria, violin plots, heat maps, and pair plots etc. While reflective of the chosen testing matrix rather than intrinsic rock properties, such a routine of quantifying descriptive statistics is nonetheless a part of good practice as it aids with selection of ML models and training strategies that are suited to the nature of the datasets.

The models were trained using seventy percent (70) of the data points while testing and validation of model was performed on the remaining thirty percent (30) of the data points. The prediction performances of ML models were evaluated using root mean squared error (RMSE) and correlation coefficient 'R'.

This report begins with a description of the machine learning technique FFNN used for the prediction modelling of stresses. The FFNN model was trained using the *Deep Learning Toolbox* of MATLAB software (MATLAB 2022). Exploratory data analysis was performed using open-source software *Python (ver:3.9.13)* (Python software foundation, 2023) and program codes were run on IDE *Spyder (ver:5.2.2)* (Cerezo et al. 2023). Different libraries of *Python* software were utilized such as *Pandas* (McKinney et al. 2023), *Seaborn* (Waskom and Seaborn, 2023), *Matplotlib* (Hunter and Droettboom, 2016), and *SciPy* (Jones et al. 2023) during the performance of EDA. Then, a comprehensive ML modelling workflow with each step is presented. The main findings of EDA are discussed for the given input and output features. A thorough illustration of the main features of FFNN model optimization and hyperparameter tuning are provided. Subsequently, models' accuracy and prediction performance are discussed. The last section demonstrates the mathematical correlation development for the three FFNN prediction models. Finally, the key findings and important postulates are provided in conclusions.

3. Artificial Neural Network (ANN)

Artificial neural network (ANN) is one of the extensively employed machine leaning methods for handling prediction issues, data mining, pattern recognition, and approximation (Mozaffari and Azad, 2014) 24, 25. Although various types of ANN techniques are used, backpropagation and feedforward ANN algorithms are the most implemented for prediction and training (Chau, 2007). ANN uses various learning functions, networks and activation functions to provide the required prediction solution for the specified problem (Mohaghegh et al. 1994).

ANN solves the underlying problem through replicating the components and operations of the nervous system of human beings. Artificial neurons are components of an explicit architectural network that are chosen based on the specifications of the given engineering challenge (Ali, 1994). In order to produce accurate, dependable, and consistent prediction outcomes, ANN establishes connections among the nonlinear variables using a massive collection of algorithms (Otchere, 2021). In general, feed-back neural networks (FBNN) and feed-forward neural networks (FFNN) are the two types of ANN that are widely used. FFNN is a basic and simplest ANN design that uses interconnected perceptron layers to build a forward-only, unidirectional process of transferring information. In order to identify a precise and pertinent output feature, the knowledge of input features is transmitted nonlinearly via activation function of hidden layers. The second variant, which has the same architectural qualities as FFNN with the extra feature of establishing a back loop, is equally accepted, and frequently used for ANN applications (Saggaf et al. 2003). The back loop feature of FBNN iteratively refines the predicted output feature by delivering error information to execute the algorithm again to modify the weights until accuracy measures (errors and correlation coefficients) can no longer be improved (Saikia, 2020).

Numerous interconnections were established through the specified routes for the interaction between the nodes of ANN structure. Each node of the hidden layer is assigned with specific weight to streamline the functionality of ANN structure (Hornik et al., 1989). During the training process, algorithms are fed with input features in the form of vectors. After the output errors are routed back, the weights between the nodes are finely adjusted using gradient descent. The process is kept repeating until the output errors are not further improved. The gradient descent uses the error function to update the weights between the nodes of ANN structure as shown in Eq. 1 (Avseth and Mukerji, 2002).

$$\Delta \mathrm{W(t)} = \Delta \mathrm{W(t-1)} * \alpha + \Delta \mathrm{E} * \eta * (\mathrm{t}) \qquad \text{(Eq. 1)}$$

Where η, E, Δw, and α, demonstrate adjusted weights, output error, learning components, and momentum, respectively.

There are various advantages of using ANN as compared to other supervised ML algorithms.
1. It has the ability to handle the non-linearity and complexity association between input and output features.
2. ANN functions with great efficiency while dealing with high dimensionality of the data.
3. ANN can demonstrate the complicated classification groups and non-limiting functions that contain output and input features.

4. Methodology – Modelling Approach

4.1. Machine Learning Modelling Workflow

The ML workflow started with data collection, data cleaning and exploratory data analysis of experimental dataset to explore the data distribution, relationships, and relative importance of input features with respect to output features. Exploratory data analysis was performed using different libraries of open-source software *Python (ver:3.9.13)* (Python software foundation, 2023) and program codes were executed on *Spyder (ver:5.2.2)* integrated development environment (Cerezo et al. 2023). The *Pandas (ver:1.4.4)* library of *Python* (McKinney, 2022) was used to extract the *Microsoft Excel* data and generate data frame on *Python* software. The *Seaborn (ver:0.11.2)* library was employed for generating the heat maps, pair plots, histogram, KDE, and violin plots (Waskom and Seaborn, 2023). All EDA plots were generated using the *Matplotlib.pyplot* module of *Matplotlib (ver:3.5.2)* library of *Python (ver:3.9.13)* software (Hunter and Droettboom, 2016). Statistical features and correlation coefficients (Pearson, Spearman, and Kendall) were determined using *Scipy (ver:1.9.1)* library (Jones et al. 2023). After EDA analysis, dataset is ready to be fed for ML modelling. All the Python libraries and IDE (Integrated Development Environment) collectively function under the *Anaconda (ver:22.9.0)* package (Anaconda Inc. 2023).

Subsequently, the FFNN type of ANN technique with the feature of back propagation was employed to develop prediction models for vertical and horizontal stresses in the Granitoid Formation. Training of model was performed using the *Deep Learning Toolbox* of MATLAB software (MATLAB, 2022). Grid search cross validation process was used for optimizing the hyperparameters in FFNN. The complete workflow adopted in this study is shown in Figure 1.

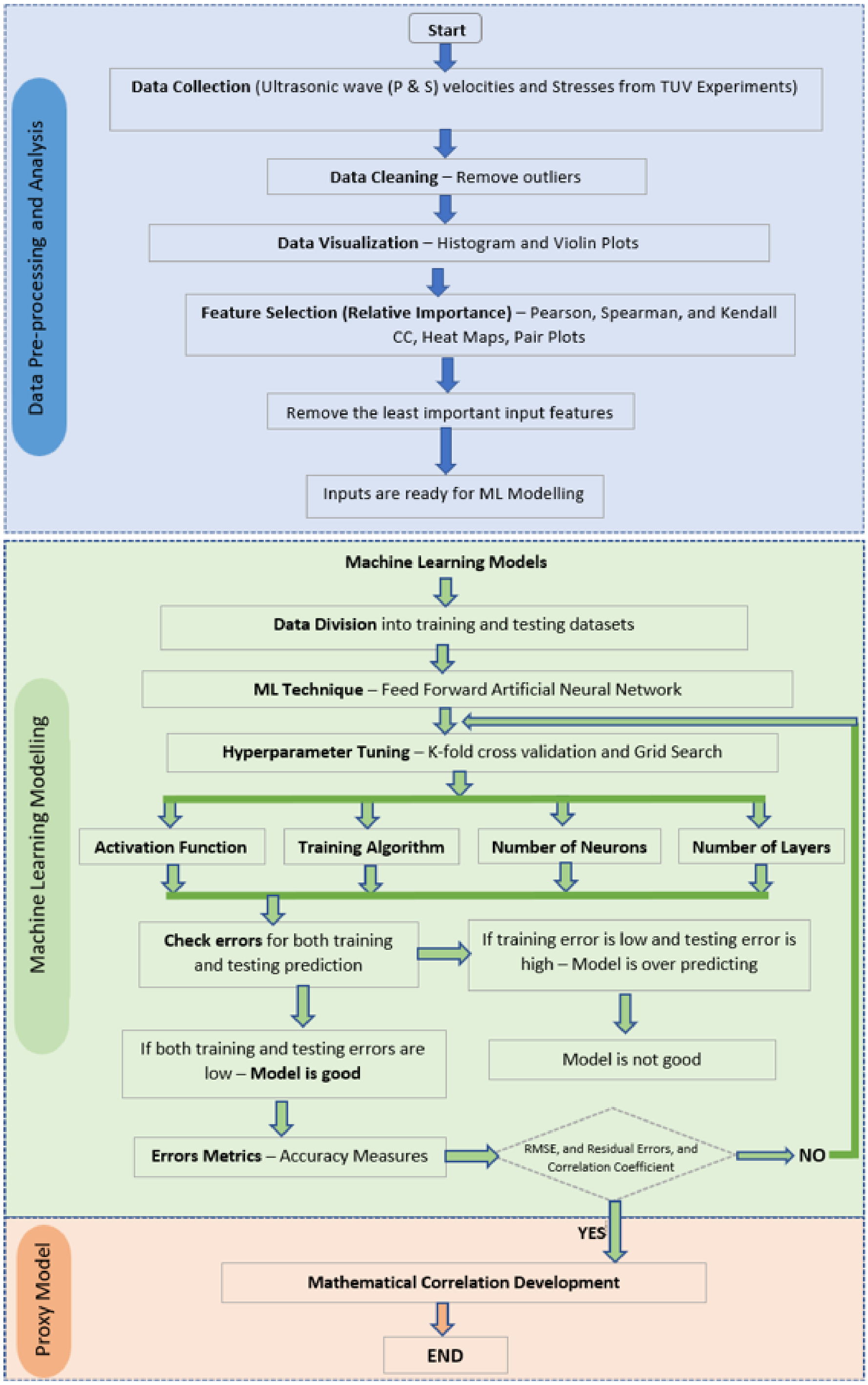
Data Pre-processing and Analysis
Start
Data Collection (Ultrasonic wave (P & S) velocities and Stresses from TUV Experiments)
Data Cleaning – Remove outliers
Data Visualization – Histogram and Violin Plots
Feature Selection (Relative Importance) – Pearson, Spearman, and Kendall CC, Heat Maps, Pair Plots
Remove the least important input features
Inputs are ready for ML Modelling
Machine Learning Modelling
Machine Learning Models
Data Division into training and testing datasets
ML Technique – Feed Forward Artificial Neural Network
Hyperparameter Tuning – K-fold cross validation and Grid Search
Activation Function
Training Algorithm
Number of Neurons
Number of Layers
Check errors for both training and testing prediction
If training error is low and testing error is high – Model is over predicting
Model is not good
If both training and testing errors are low – Model is good
Errors Metrics – Accuracy Measures
RMSE, and Residual Errors, and Correlation Coefficient
NO
YES
Proxy Model
Mathematical Correlation Development
END

Figure 1: Modelling Workflow for this study

4.2. Exploratory Data Analysis (EDA)

In this study, ultrasonic wave velocity (actually the inverse of the velocities, the P- and S-wave slownesses) and stress data were used to develop three machine learning (ML) prediction models for vertical stress (σ_z) and two orthogonally oriented horizontal stresses (σ_x and σ_y). The dataset was obtained from the true triaxial ultrasonic velocity (TUV) experiments on lower and upper granitoid formations with samples at 5474' MD and 5850' MD, respectively. See Bunger et al. (2023) for a detailed description of the labTUV experiments.

Based on this available dataset, a total of 46 TUV data points were utilized that contain P-wave (h_{zz}), and S-wave slowness (h_{zy} and h_{zx}), in z-direction under different combinations of applied stresses 'σ_z', 'σ_x', and 'σ_y' in z, x, and y directions respectively (Figure 2). Note that throughout this report, each loading combination is designated an identifier, its "dataset number". While the order of these does not have physical meaning, it is nonetheless sometimes useful for visualizing the span of the data to report data by this number.

The first ML model was developed to predict the vertical stress 'σ_z' using P and S-wave slowness 'h_{zz},', 'h_{zy}', and 'h_{zx}' as inputs features. Note that throughout this report, slownesses ('h') are given two subscripts, where the first gives the propagation direction and the second gives the direction of particle motion. Hence, 'h_{zz},', 'h_{zy}', and 'h_{zx}' all describe waves propagating in the z-direction and indicate the P-wave, the y-polarized S-wave, and the x-polarized S-wave, respectively. For the second ML model, four input features including 'h_{zz},', 'h_{zy}', 'h_{zx}' and 'σ_z' were utilized to predict the stress (σ_x) applied in x-direction. The third ML model was developed using the same four input features ('h_{zz},', 'h_{zy}', 'h_{zx}' and 'σ_z') for the prediction of stress 'σ_y' applied in y-direction. The total data points were split into two portions. The first portion (seventy percent) of the dataset was dedicated for model training while validation and testing of the trained model was performed using the second portion (remaining thirty percent) of the data points. Testing and training sets of data used for 'σ_z', 'σ_x', and 'σ_y' prediction models are presented in Figures 3-5, respectively.

Testing and training datasets were firstly subjected to exploratory data analysis (EDA) for the purpose of data cleaning (if necessary, although none was needed in this case), understanding the data distribution, relative importance, and relationships of input and output features. The EDA is essential for feature selection prior to the execution of ML algorithms to ultimately generate a robust prediction model. Important statistical parameters such as maximum, minimum, mode, mean, median, kurtosis, skewness, and standard deviation of input and output features are demonstrated in Table 1. The statistical indicators are reflective of the choices of testing matrix (load combinations) carried out in the lab, not of any intrinsic rock behavior. Visualization of data distribution for each feature is provided by the violin plots that expresses the data points in the form of kernel density estimation function (KDE). Violin plots are important to visualize the extreme values, inter quartile range (IQR), and arithmetic mean in the dataset as shown in Figure 5. Bimodal distribution of data can also be easily identified by visualizing violin plots as shown for 'h_{zy}' and 'σ_x' features (Figure 6).

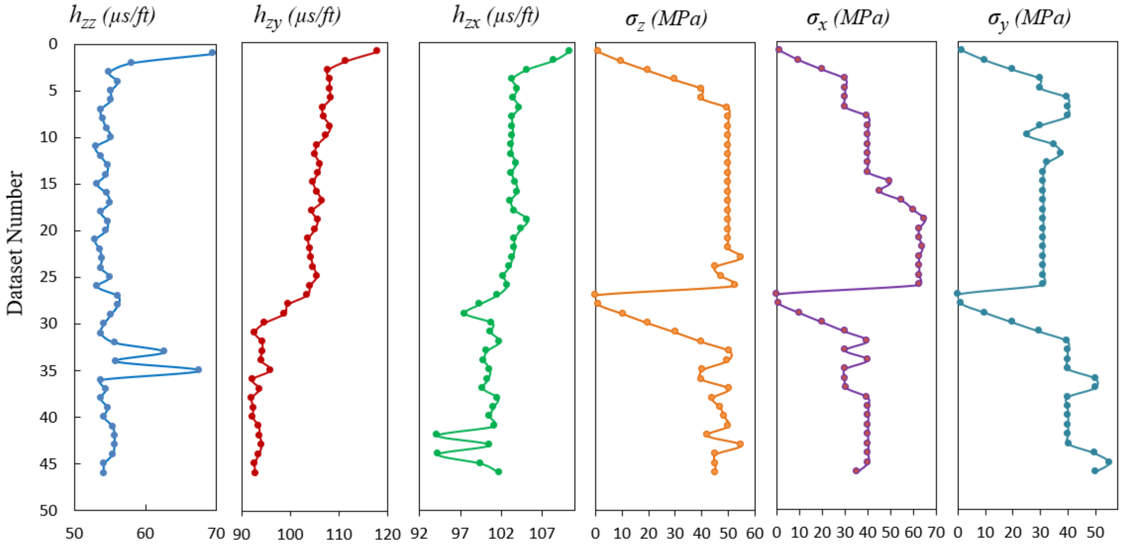

Figure 2: Total Dataset used for all three ML prediction models.

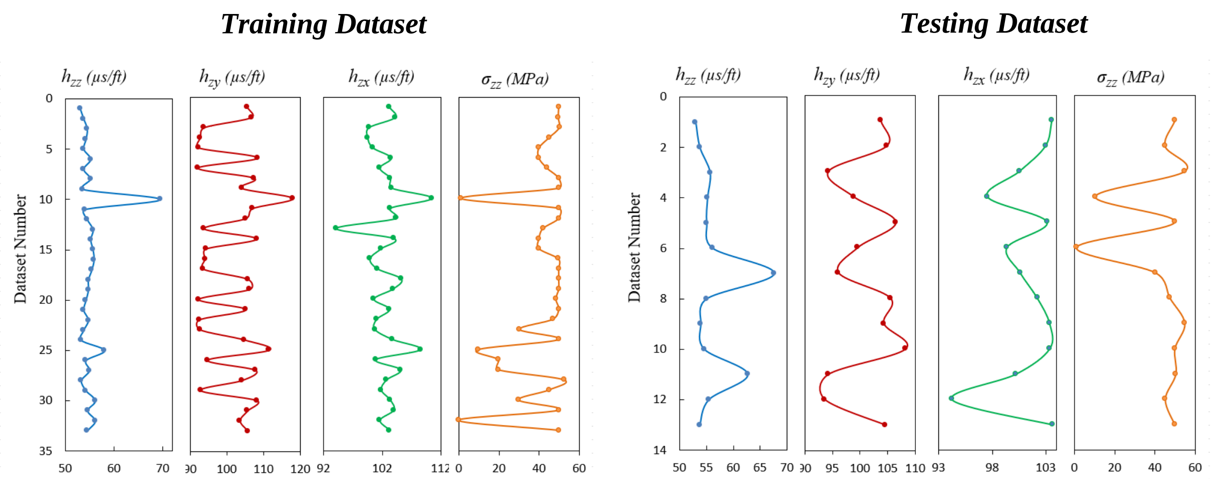

Figure 3: Training and testing dataset used for vertical stress '$\boldsymbol{\sigma_z}$' prediction model.

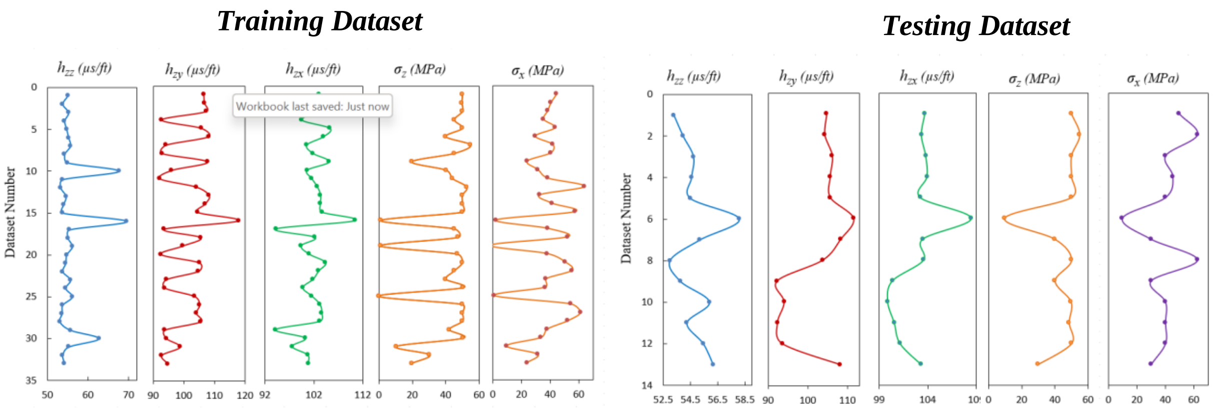

Figure 4: Training and testing dataset used for horizontal stress '$\boldsymbol{\sigma_x}$' prediction model.

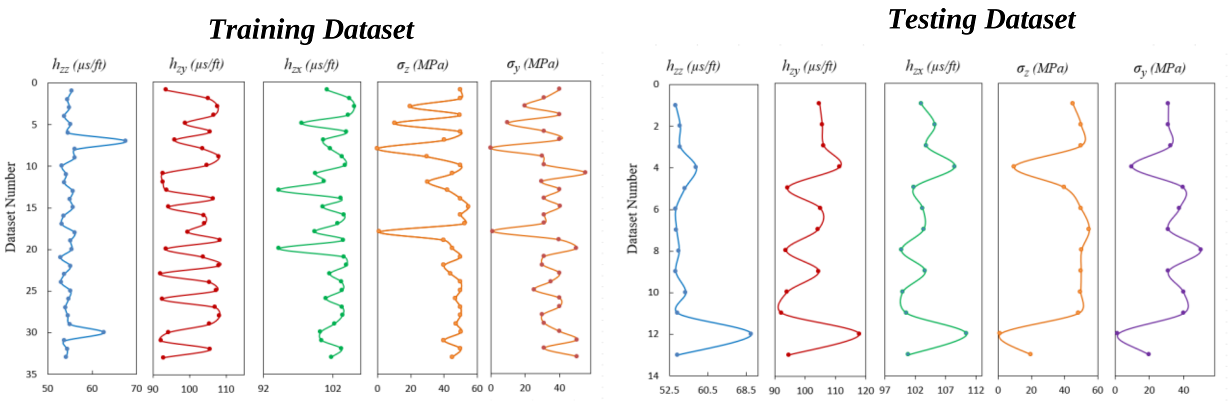

Figure 5: Training and testing dataset used for horizontal stress '$\boldsymbol{\sigma_y}$' prediction model.

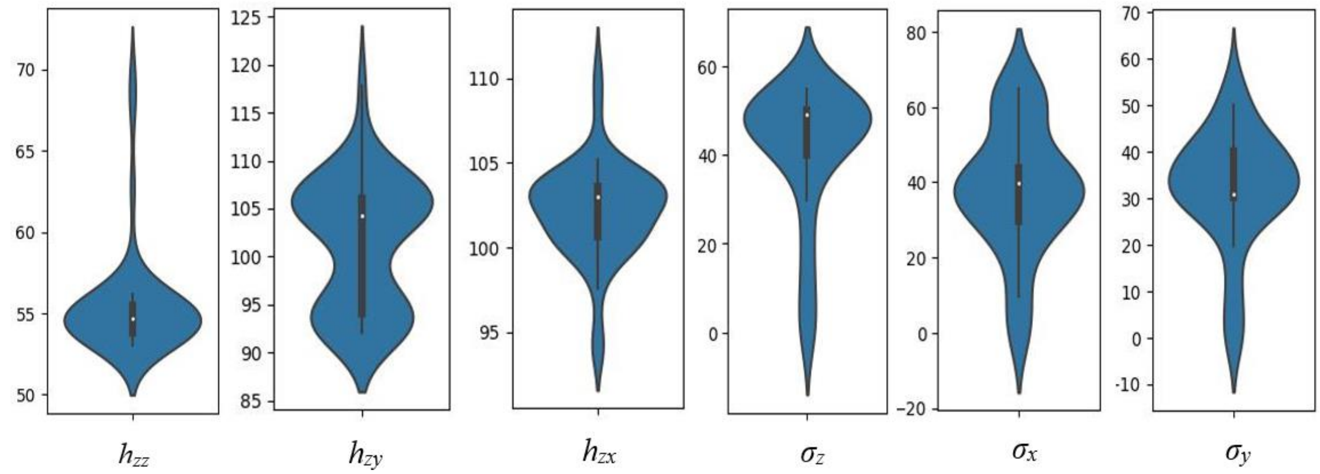

Figure 6: Violin plots for input and output variables

Data distribution of all input and output features in the form of histogram is shown in Figure 7. Histogram revealed that P-wave slowness ‘h_{zz}’ is positive (right) skewed indicating that most of the data points are clustered towards the left side of distribution reflecting non-symmetric distribution. A major portion of the ‘h_{zz}’ dataset is clustered towards to the left side of the plot (values are lower than the mean values) that leads to the longer tail on right side of the distribution. The S-wave slowness ‘h_{zx}’ and ‘h_{zy}’ is not skewed or slightly negatively skewed indicating fairly symmetric distribution of data on both sides of the mean value. The majority of data points exist around the mean values of these input features. On the contrary, the stresses ‘σ_z’, ‘σ_x’, and ‘σ_y’ data is negative (left) skewed exhibiting majority of the dataset is larger than mean value with longer tail on left side. The distribution revealed that mean value is smaller than mode and median. Kurtosis of P-wave slowness is leptokurtic with heavier tailed and peaked distribution as reflected by higher positive value. Most of the data points lie in the tail of distribution instead of close proximity of the mean value. Other five input features such as S-wave slowness (h_{zx} and h_{zy}) and stress (σ_z, σ_x, and σ_y) data distribution reflect platykurtic distribution with lower and broader peaks. The tail of data distribution curve is lighter than normal distribution indicating minimum chances of outlier existence in the dataset. These features are lacking extreme values with flatter distribution around the mean value that reflects good data quality of data.

The relative importance of all the input features were explored using three criteria namely Pearson, Spearman, and Kendall correlation coefficients (R) as shown in Figure 8. For the first ML model, the input feature ‘h_{zz}’ exhibited strong inverse correlation with output ‘σ_z’ while relatively weaker correlations were observed for the other two input features ‘h_{zy} ’and ‘h_{zx}’. On the other hand, horizontal stress ‘σ_x’ exhibited strong correlation with P-wave slowness ‘h_{zz}’ and vertical stress ‘σ_z’. S-wave slowness ‘h_{zy}’ and ‘h_{zx}’ demonstrated weaker correlation with ‘σ_x’. For the case of other horizontal stress ‘σ_y’, three input features ‘σ_z’, ‘h_{zy}’ and ‘h_{zx}’ demonstrated strong relationship and P-wave slowness ‘h_{zz}’ reflected weaker relationship with ‘σ_y’.

The heat maps revealed the collinearity between the pairs of input and output variables as shown in Figure 9. Heat maps were generated using the Spearman, Pearson, and Kendall correlation criteria. Furthermore, pair plots among all the input and output variables demonstrated the inter-relationships between all the pairs of output and input features in a single plot (Figure 10). The velocity variations with stresses are more prominent for the Lower Granitoid Formation. It could be observed in the pair plots that P- and S-wave slowness have inverse relationship with vertical and horizontal stresses however, the impact is more pronounced for Lower Granitoid Formation compared to Upper Granitoid Formation. The diagonal of pair plot represents the KDEs distribution of the variables. Pairs of slowness exhibited direct relationship between them. Pearson, Spearman, and Kendall's correlation criterion are provided in Eqs. 2–4, respectively.

$$\rho_{pearson} = \frac{\mathrm{k}\sum \mathrm{xy} - (\sum \mathrm{x})(\sum \mathrm{y})}{\sqrt{\mathrm{k}(\sum \mathrm{x}^2) - (\sum \mathrm{y})^2}\sqrt{\mathrm{k}(\sum \mathrm{b}^2) - (\sum \mathrm{b})^2}} \qquad \text{(Eq. 2)}$$

Where y and x represent the respective variables, and number of samples are represented by k.

$$\rho_{spearman} = \rho_{pearson}\frac{\mathrm{cov}(x,y)}{\gamma_x \gamma_y} \qquad \text{(Eq. 3)}$$

$$\tau_{kendall} = \frac{n_c - n_d}{n(n-1)/_{2}} \qquad \text{(Eq. 4)}$$

Where n_c and n_d represent the values of number of concordant and discordant pairs and total number of samples are represented by ‘n’.

Table 1: Statistical indicators of dataset used in ML Modelling

Statistical Parameters	*h_{zz} (μs/ft)*	*h_{zy} (μs/ft)*	*h_{zx} (μs/ft)*	*σ_z (MPa)*	*σ_x (MPa)*	*σ_y (MPa)*
Minimum	53.0	92.1	94.2	0.0	0.0	0.0
Maximum	69.6	117.8	110.4	54.9	65.0	55.1
Mean	55.4	101.4	102.2	41.2	38.3	32.4
Mode	54.7	93.6	103.2	49.9	40.0	31.1
Median	54.6	104.2	103.0	49.1	40.0	31.1
St. Dev.	3.2	6.7	2.8	15.1	17.0	12.4
Kurtosis	11.9	-1.0	2.8	2.0	0.1	1.3
Skewness	3.4	0.0	-0.3	-1.8	-0.4	-1.0

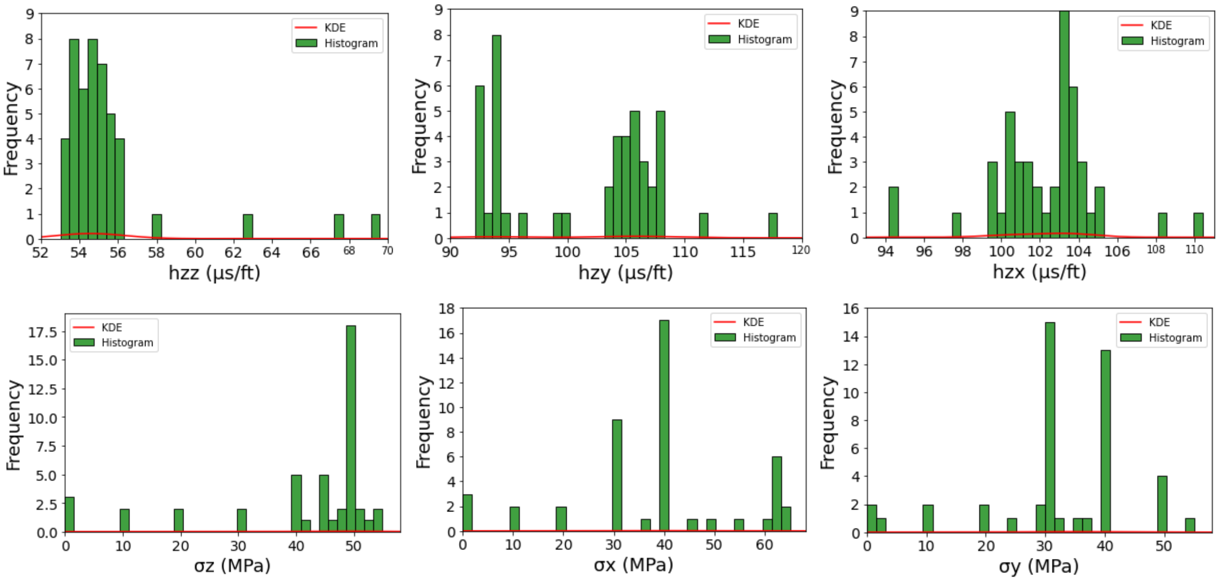

Figure 7: Histogram and KDE plots of input and output datasets

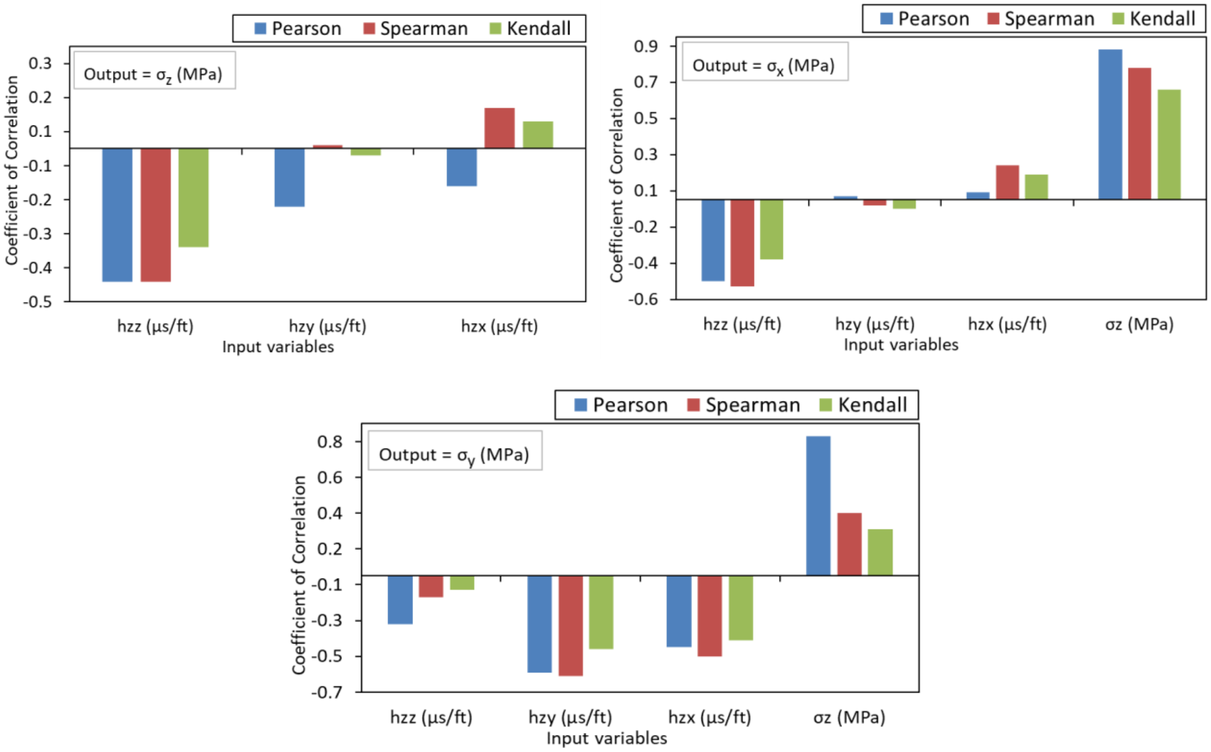

Figure 8: Coefficient of correlations of input variables with output showing relative importance.

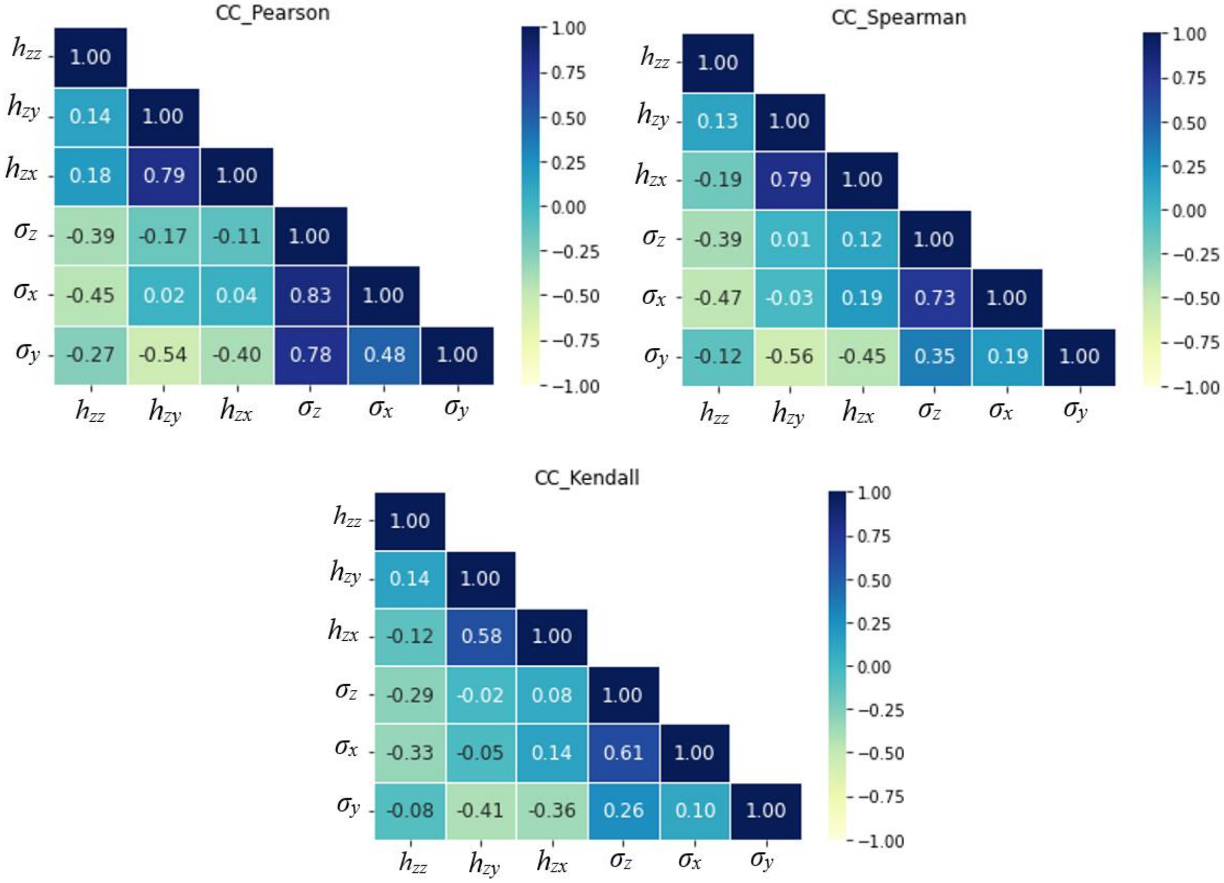

Figure 9: Heat maps showing the Pearson, Spearman and Kendall coefficient of correlations between input variables.

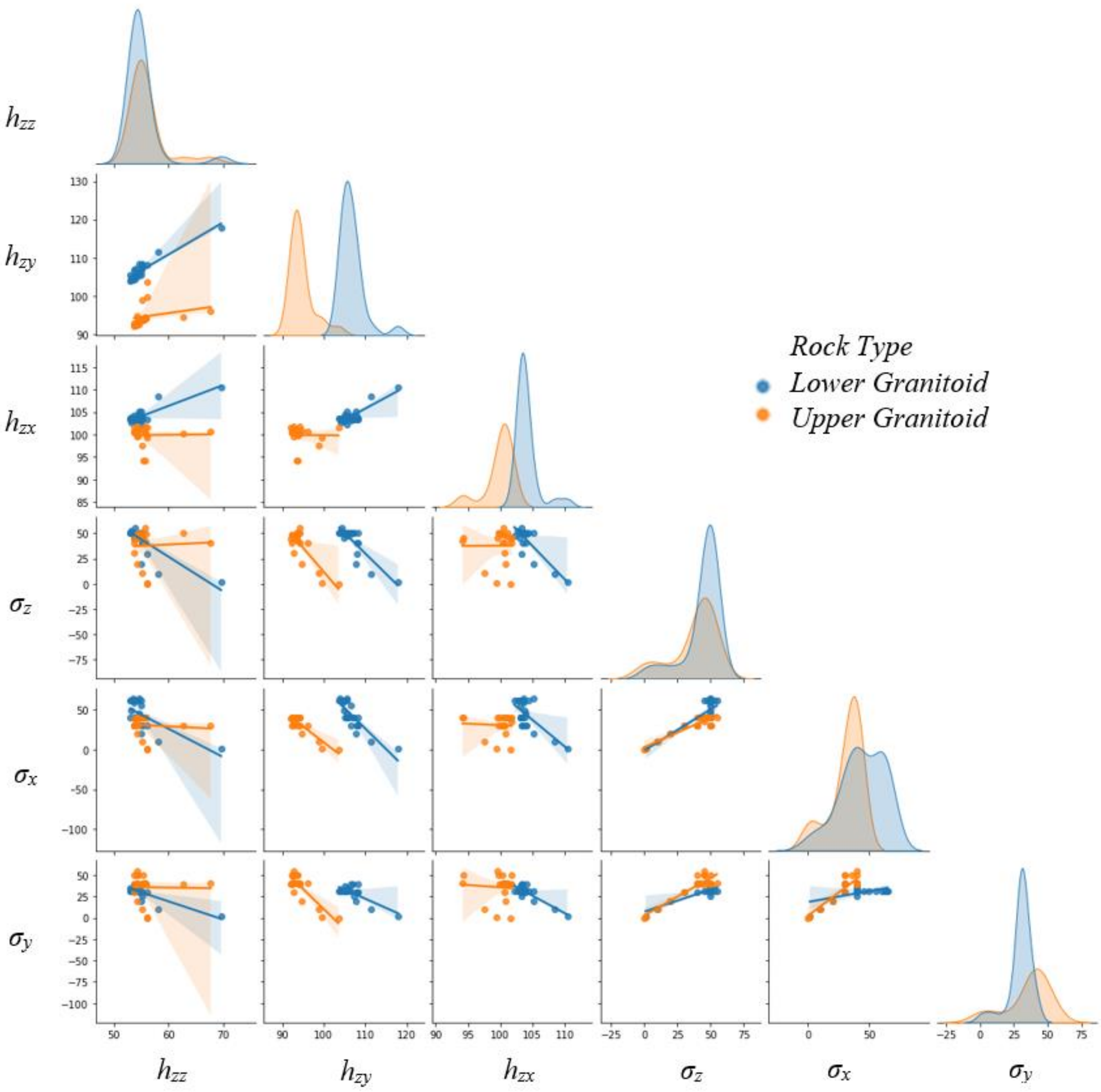

Figure 10: Pair plots showing cross plots between input and output features.

4.3. Hyperparameter Tuning and Models Optimization

This study presented three robust machine learning models developed to predict the stresses in vertical (σ_z) and two horizontal directions (σ_x and σ_y) for the Granitoid Formation using fully connected Feed Forward Neural Network (FFNN) technique. The ML model was trained using ultrasonic wave velocities in z directions which were obtained from the TUV laboratory experiments.

The FFNN prediction model for stresses was optimized by improving the coefficient of correlation (R) and minimizing the prediction errors through different strategies of hyperparameter tuning. The data points were split in to two halves. The train test split function in the *MATLAB* software (MATLAB, 2022) were used to divide the data. The data splitting was an arbitrary occurrence. Data splitting was done in such a way that testing data contained the range of the input variables present in the training dataset. Seventy percent of the dataset was set aside for the model training and the remaining thirty percent was set aside for testing and validation. Seed function was used to regulate the random number generation for each run. ANN algorithm was run several times using numerous selected hyperparameters such as input features, neurons count in hidden layers, training functions, activation functions, and realizations count. The k-fold strategy was used to cross-validate the training process in order to avoid the overfitting of data.

For all three stress prediction models, the FFNN algorithm was executed for different training functions such as resilient propagation, Levenberg-Marquardt, BFGS Quasi-Newton, Scaled Conjugate Gradient, and Fletcher-Powell Conjugate Gradient to optimize the prediction accuracy ((MATLAB, 2022)). Best prediction results were obtained with Levenberg-Marquardt training function for stresses σ_z, σ_x, and σ_y. The best activation function was found to be Tangent sigmoidal and linear functions that connect the hidden and input layers, and the output and hidden layers, respectively after executing the model using different activation functions such as tangent sigmoidal, logarithmic sigmoidal, soft max, linear, triangular basis activation functions (MATLAB, 2022). Optimization of neurons count in hidden layers was performed by executing the model algorithm at various neurons counts from 5 to 40 starts from lower numbers to higher (Opper and Haussler, 1996). A comparison of model performance at different number of neurons is demonstrated in Figure 11. The model accuracy was further improved by executing the model for 1000 realizations for the selected neurons count in the central hidden layer to catch the non-distinctiveness of the data points (Zhang, 2022). The optimum prediction results were achieved at 9, 7, and 9 number of neurons and 375, 678, and 260 realizations for stresses σ_z, σ_x, and σ_y, respectively exhibiting minimum errors and maximum coefficient of correlation (R). The best realization for the optimized neurons count is shown in Figure 13.

4.4. Performance Measures

Two performance measures such as correlation coefficient (R), and root mean squared error (RMSE) were selected for evaluation of accuracy and consistency of the predicted output. The mathematical expression of RMSE is demonstrated in Eq. 5. The extent of accuracy was assessed from ‘R’ value between predicted and actual values.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}[\beta_a - \beta_p]^2)}{n}} \qquad \text{(Eq. 5)}$$

Where; β_a and β_p are the predicted and actual values and total data counts are represented by ‘n’.

5. Modelling Results

All three stress prediction models exhibited excellent performance in terms of low errors and high correlation coefficient (R). The topology of neuron structures for the three models are shown in Figure 12.

The developed FFNN models provided reliable and robust predicted vertical and horizontal stress values. The prediction performance of all three models FFNN ML model in terms of cross plots between predicted and experimental stress values are demonstrated in Figure 14. For the vertical stress ‘σ_z’ prediction model, the correlation coefficient (R) between predicted and experimentally determined stresses was observed as 0.978 and 0.975 for testing (unseen data) and training (model building) datasets, respectively. The model demonstrated excellent outcome with prediction error (RMSE) of 3.5 and 3.1 for testing and training prediction, respectively. The FFNN training and testing predictions of all three prediction models (‘σ_z’, ‘σ_x’, and ‘σ_y’) are also compared with experimental stress values demonstrating good agreement between them. For the horizontal stress ‘σ_x’ prediction model, the ‘R’ value between predicted and experimental data points was determined to be 0.965 and 0.923 for testing and training data points, respectively. The excellent model functioning was reflected by the low RMSE errors for testing (4.4) and training (6.3) prediction, respectively.

The second horizontal stress ‘σ_y’ prediction model resulted in reliable and steadfast predicted values of stress. The model demonstrated the correlation coefficient (R) of 0.95 for testing and 0.934 for training prediction results. The prediction error RMSE for the horizontal stress ‘σ_y’ was observed to be 4.8 and 4.2 for testing and training data points, respectively, demonstrating the accuracy of model prediction. Hence, all three FFNN models for vertical and horizontal stresses prediction demonstrated accurate, reliable, and consistent prediction abilities as reflected by the excellent accuracy measures.

Consequently, accuracy indicators revealed that developed three FFNN models can provide the reliable, robust and consistent prediction solution for vertical stress ‘σ_z’ as a function of ‘h_{zz}’, ‘h_{zx}’ and ‘h_{zy}’ and two horizontal stresses (mutually perpendicular) as a function of ‘h_{zz}’, ‘h_{zx}’, ‘h_{zy}’ and ‘σ_z’ for the given input and output features and respective ranges for which the model is trained.

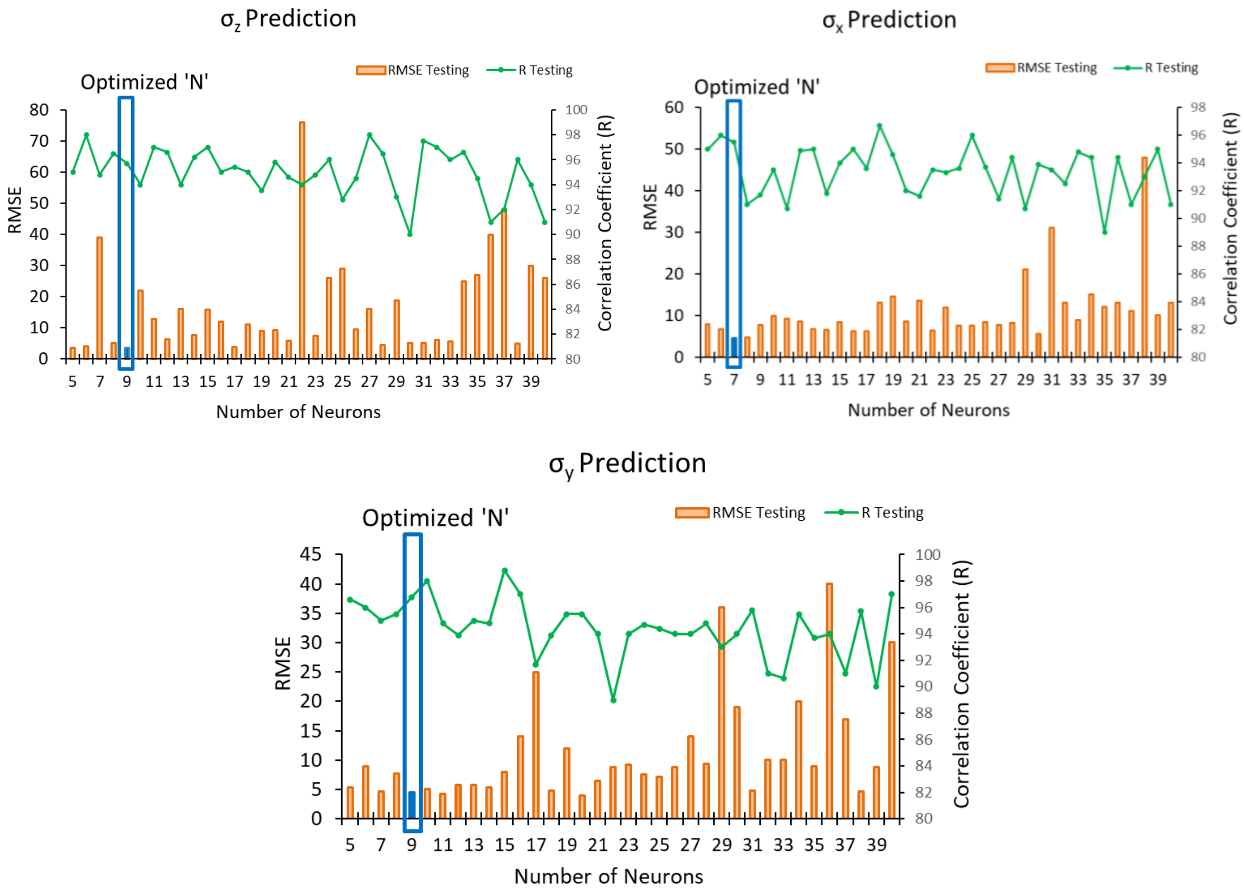

Figure 11: Models’ performance comparison for different number of neurons

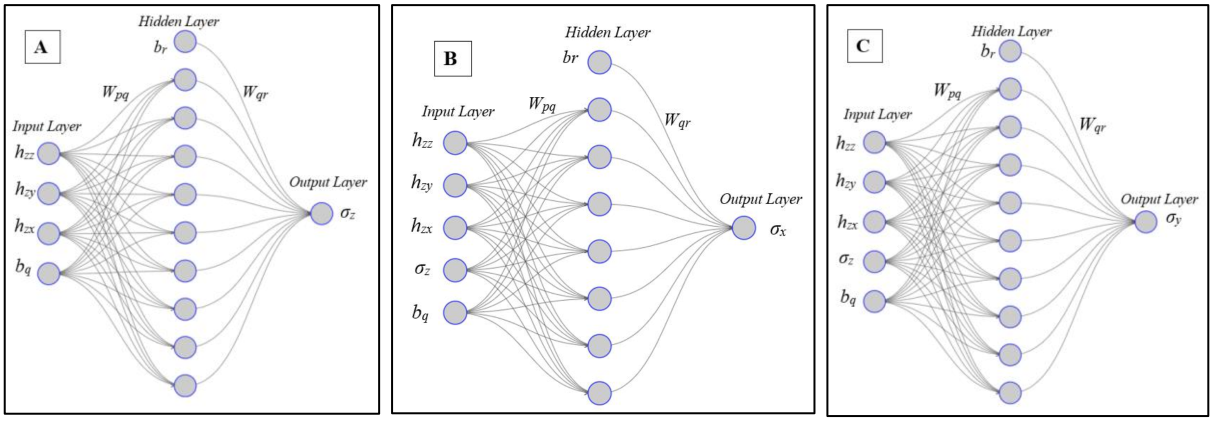

Figure 12: ANN topology showing neurons structure for: (A) ‘σ_z’, (B) ‘σ_x’, and (C) ‘σ_y’ prediction models.

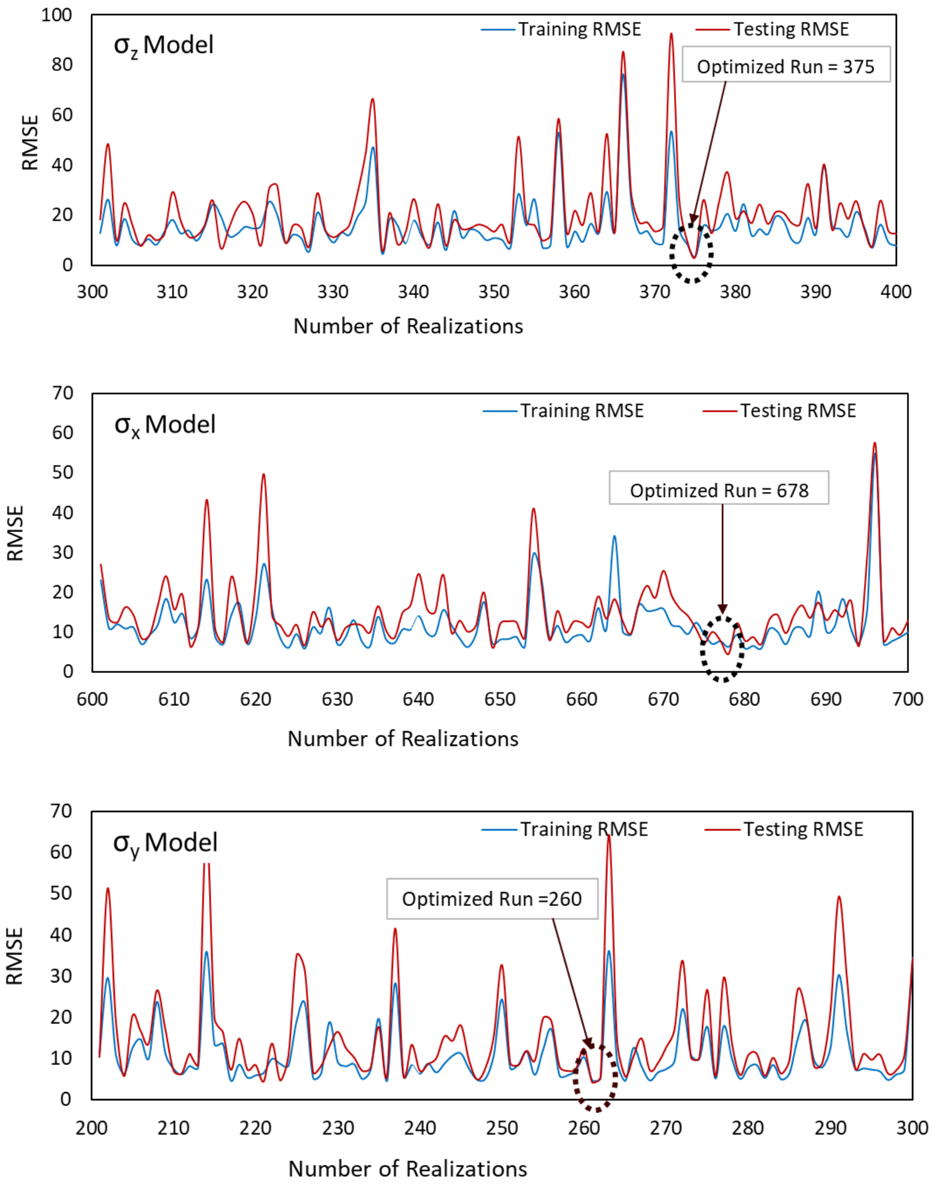

Figure 13: RMSE comparison for selected 100 realizations at optimized number of neurons for 'σ_z', 'σ_x', 'σ_y' models.

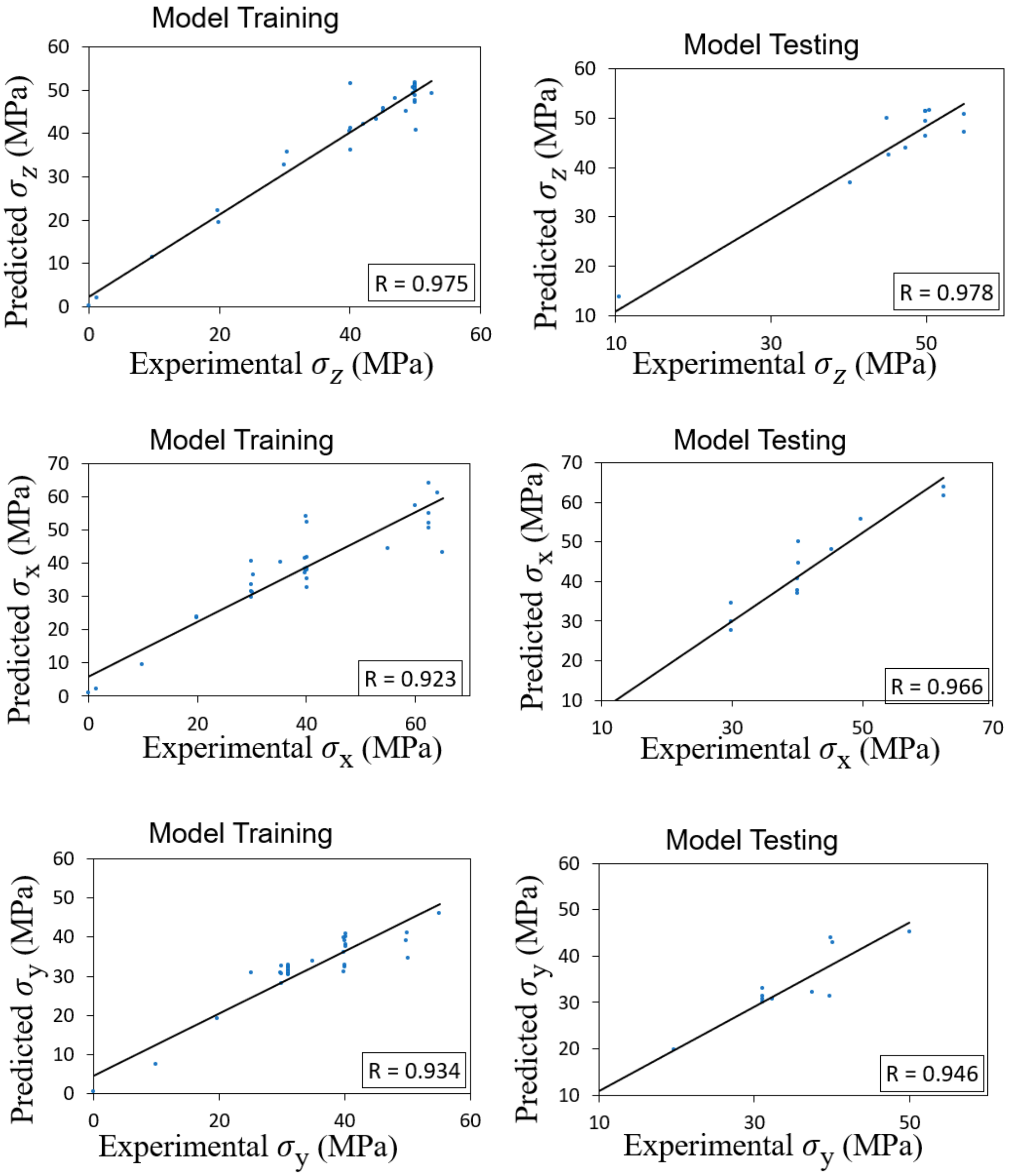

Figure 14: Training and testing prediction performance for ‘σ_z’, ‘σ_x’, and ‘σ_y’ for the proposed FFNN models

The comparison between the predicted and experimental stress ‘σ_z’, ‘σ_x’, and ‘σ_y’ values revealed good harmony between them as shown in Figures 15-17. The models’ accuracy measures are compared for the proposed three stress ‘σ_z’, ‘σ_x’, and ‘σ_y’ prediction models and demonstrated in Figure 18. All three ML model exhibited excellent prediction performance as reflected by evaluation metrics. However, the highest accuracy was revealed by the vertical stress ‘σ_z’ prediction model with testing R and RMSE of 0.978 and 3.5, respectively.

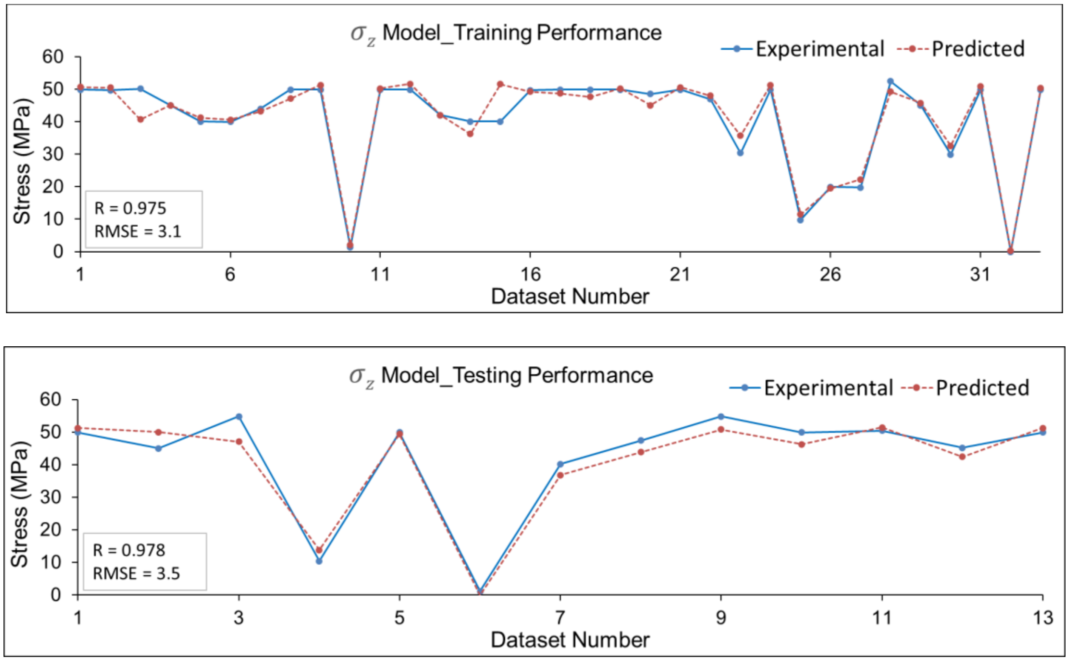

Figure 15: Comparison of testing and training predictions with experimental ‘σ_z’.

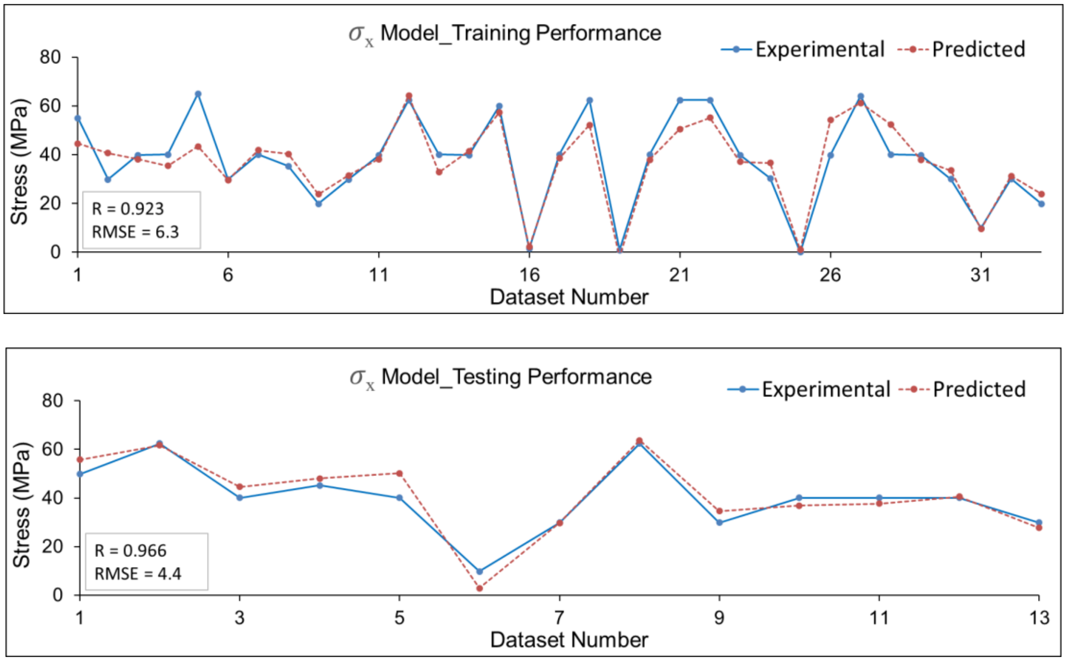

Figure 16: Comparison of testing and training predictions with experimental ‘σ_x’.

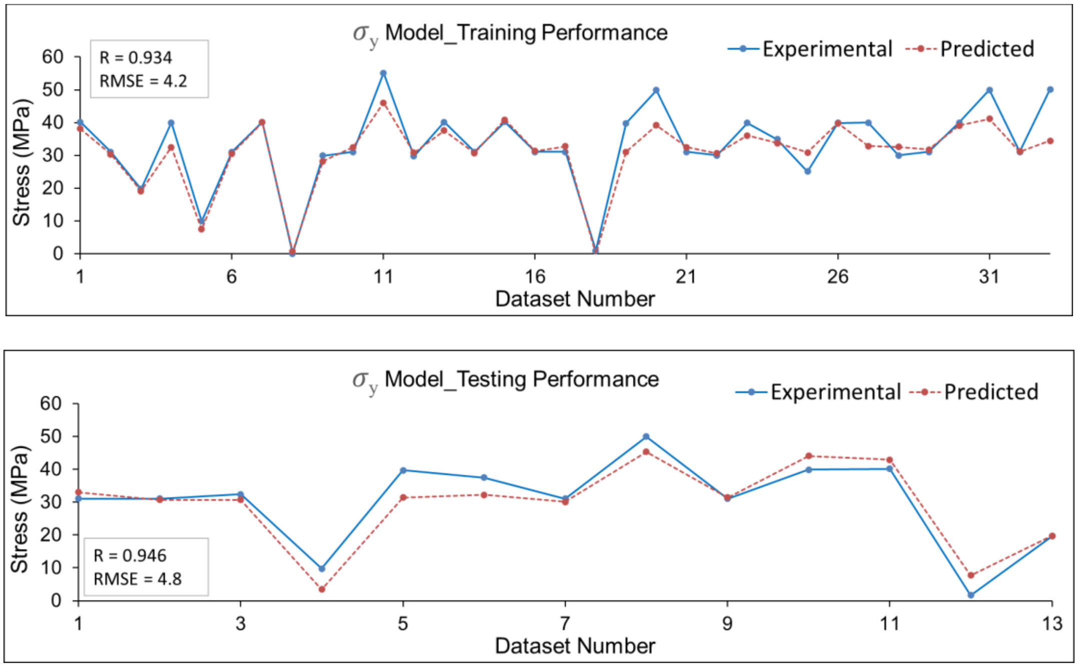

Figure 17: Comparison of testing and training predictions with experimental 'σ_y'.

6. Mathematical Correlation Development

All three optimized FFNN prediction models were converted into empirical mathematical correlations for stress estimations in vertical and two horizontal directions. The mathematical model can be conveniently used for stress estimation without running ML codes. The mathematical correlation functions well with the extracted weights and biases of optimized FFNN model. Weights and biases for each optimized model are provided in Tables 2-4.

6.1. Vertical Stress 'σ_z' Prediction Model

The mathematical expression of ML prediction model is given as below.

$$(\sigma_z)_N = \sum_{q=1}^{H_n} W_{qr} n_{hq} + b_r \qquad \text{(Eq. 6)}$$

$$n_{hq} = f(\sum_{p=1}^{N_p} W_{pq} i_p + b_q) \qquad \text{(Eq. 7)}$$

$$n_{hq} = f\Big((W_{1q}(h_{zz})_N + W_{2q}\big(h_{zy}\big)_N + W_{3q}(h_{zx})_N + b_q\Big) \qquad \text{(Eq. 8)}$$

f (x) = $\frac{2}{1+e^{-2x}} - 1$ = tanh(x) (Hyperbolic Tangent Sigmoidal Activation Function) (Eq. 9)

However, input features were normalized before the FFNN simulations (using Eq. 10).

$$Inputs\ (i_p) = \frac{(\psi_{max}-\psi_{min})(i-i_{min})}{(i_{max}-i_{min})} + \psi_{min} \qquad \text{(Eq. 10)}$$

Where, ψ_{max} and ψ_{min} are 1 and -1 respectively. The maximum and minimum values of input features are provided in Table 1. After normalization process, input features are shown in Eq. 11-13.

$$(h_{zz})_N = 0.118\left((h_{zz}) - 53.0\right) - 1 \qquad \text{(Eq. 11)}$$

$$\left(h_{zy}\right)_N = 0.078\left(\left(h_{zy}\right) - 92.1\right) - 1 \qquad \text{(Eq. 12)}$$

$$(h_{zx})_N = 0.123\left((h_{zx}) - 94.2\right) - 1 \qquad \text{(Eq. 13)}$$

The final output values of model were undergone de-normalization using Eq. 14.

$$Output = \frac{(\psi_{max}-\psi_{min})(i-i_{min})}{(i_{max}-i_{min})} + \psi_{min} \qquad \text{(Eq. 14)}$$

During the de-normalization, the values of i_{min} and i_{max} are taken as -1 and 1 respectively. The ψ_{max} and ψ_{min} correspond to the maximum and minimum values of the output feature. The de-normalized output feature for the proposed model is shown in Eq. 15.

$$(\sigma_z) = 27.45\ ((\sigma_z)_N + 1) - 0.00 \qquad \text{(Eq. 15)}$$

The model biases and weights for hidden, input, and output layers of FFNN predictive model are provided in the Table 2.

Table 2: Biases and weights connecting the input, output, and hidden layers for FFNN '$\boldsymbol{\sigma_z}$' models.

	Weights connecting Hidden and Input Layers (w_{pq})			**Weights between Output and Hidden Layers (w_{qr})**	**Hidden Layer Neurons bias values**
Hidden Layer Neurons (q)	Input Layer Neurons (p)			Output Neuron (r)	Bias (b_q)
	1	2	3		
1	2.32	0.76	1.52	-0.71	-2.94
2	4.23	-4.09	-2.51	3.25	0.50
3	2.08	2.04	0.28	0.63	-0.86
4	2.50	1.03	-1.22	-0.52	-0.89
5	2.58	-2.61	-6.05	-3.47	0.45
6	2.23	-2.35	0.25	0.30	0.38
7	0.46	0.95	-2.81	-0.59	-2.13
8	-0.03	0.25	-2.61	-1.13	2.46
9	-4.77	-6.33	-5.83	1.19	-0.27
Bias Values for Output Layer (b_r)				-0.25	

6.2. First Horizontal Stress 'σ_x' Prediction Model

The mathematical expression of ML prediction model is given as below.

$$(\sigma_x)_N = \sum_{q=1}^{H_n} W_{qr} n_{hq} + b_r \qquad \text{(Eq. 16)}$$

$$n_{hq} = f\left(\sum_{p=1}^{N_p} W_{pq} i_p + b_q\right) \qquad \text{(Eq. 17)}$$

$$n_{hq} = f\left((W_{1q}(h_{zz})_N + W_{2q}\left(h_{zy}\right)_N + W_{3q}(h_{zx})_N + W_{4q}(\sigma_z)_N + b_q\right) \qquad \text{(Eq. 18)}$$

$$\text{f (x)} = \frac{2}{1+e^{-2x}} - 1 = \tanh(x) \quad \text{(Hyperbolic Tangent Sigmoidal Activation Function)} \qquad \text{(Eq. 19)}$$

The normalized input features are shown in Eq. 20 to Eq. 23.

$$(h_{zz})_N = 0.118\left((h_{zz}) - 53.0\right) - 1 \qquad \text{(Eq. 20)}$$

$$\left(h_{zy}\right)_N = 0.078\ \left(\left(h_{zy}\right) - 92.1\right) - 1 \qquad \text{(Eq. 21)}$$

$$(h_{zx})_N = 0.123\left((h_{zx}) - 94.2\right) - 1 \qquad \text{(Eq. 22)}$$

$$(\sigma_z)_N = 0.036\left((\sigma_z) - 0.0\right) - 1 \qquad \text{(Eq. 23)}$$

For the proposed model, the de-normalized model output is demonstrated in Eq. 24.

$$(\sigma_x) = 32.5\,((\sigma_x)_N + 1) - 0.00 \qquad \text{Eq. 24}$$

The model biases and weights for hidden, input, and output layers of FFNN predictive model are provided in the Table 3.

Table 3: Biases and weights connecting the input, output, and hidden layers for FFNN '$\boldsymbol{\sigma_x}$' models.

	Weights connecting Hidden and Input Layers (w_{pq})				**Weights between Output and Hidden Layers (w_{qr})**	**Hidden Layer Neurons bias values**
Hidden Layer Neurons (q)	Input Layer Neurons (p)				Output Neuron (r)	Bias(b_q)
	1	2	3	4		
1	-1.37	-1.33	1.29	-0.52	-0.15	2.46
2	1.50	0.48	1.19	-0.39	-0.60	-2.24
3	-0.46	2.24	1.86	0.52	2.03	0.62
4	3.56	-0.78	4.65	0.67	0.66	-3.08
5	0.04	1.29	0.47	-0.07	-2.76	0.29
6	-3.71	3.28	3.31	0.72	-0.12	0.70
7	-0.87	-0.31	0.96	0.16	0.20	-3.93
	Bias Values for Output Layer (b_r)				0.05	

6.3. Second Horizontal Stress 'σ_y' Prediction Model

The mathematical expression of ML prediction model is given as below.

$$(\sigma_y)_N = \sum_{q=1}^{H_n} W_{qr} n_{hq} + b_r \qquad \text{(Eq. 25)}$$

$$n_{hq} = f\left(\sum_{p=1}^{N_p} W_{pq} i_p + b_q\right) \qquad \text{(Eq. 26)}$$

$$n_{hq} = f\Big((W_{1q}(h_{zz})_N + W_{2q}(h_{zy})_N + W_{3q}(h_{zx})_N + W_{4q}(\sigma_z)_N + b_q\Big) \qquad \text{(Eq. 27)}$$

f (x) = $\frac{2}{1+e^{-2x}} - 1$ = tanh(x) (Hyperbolic Tangent Sigmoidal Activation Function) (Eq. 28)

The normalized input features are shown in Eq. 29 to Eq. 32.

$$(h_{zz})_N = 0.118\left((h_{zz}) - 53.0\right) - 1 \qquad \text{(Eq. 29)}$$

$$(h_{zy})_N = 0.078\left((h_{zy}) - 92.1\right) - 1 \qquad \text{(Eq. 30)}$$

$$(h_{zx})_N = 0.123\left((h_{zx}) - 94.2\right) - 1 \qquad \text{(Eq. 31)}$$

$$(\sigma_z)_N = 0.036\left((\sigma_{zz}) - 0.0\right) - 1 \qquad \text{(Eq. 32)}$$

For the proposed model, the de-normalized model output is demonstrated in Eq. 33.

$$(\sigma_y) = 27.55\left(\left((\sigma_y)_N + 1\right)\right) - 0.00 \qquad \text{Eq. 33}$$

The model biases and weights for hidden, input, and output layers of FFNN predictive model are provided in the Table 4.

Table 4: Biases and weights connecting the input, output, and hidden layers for FFNN 'σ_y' models.

	Weights connecting Hidden and Input Layers (w_{pq})				**Weights between Output and Hidden Layers (w_{qr})**	**Hidden Layer Neurons bias values**
Hidden Layer Neurons (q)	Input Layer Neurons (p)				Output Neuron	Bias(b_q)
	1	2	3	4		
1	-1.54	-0.18	-0.46	-1.37	-0.50	2.82
2	0.01	-1.23	0.91	-1.99	-0.62	-1.29
3	0.05	0.45	0.60	-0.96	-0.46	-1.21
4	0.96	0.80	1.13	1.10	-0.17	0.02
5	2.56	0.47	0.13	-1.57	0.10	-1.41
6	-0.74	1.55	1.75	-0.23	-0.16	0.20
7	1.98	-1.19	-1.16	1.06	-0.27	2.16
8	0.72	-0.24	3.33	0.10	-0.22	-0.83
9	1.58	-0.84	1.76	0.85	0.90	2.10
	Bias Values for Output Layer (b_r)				-0.76	

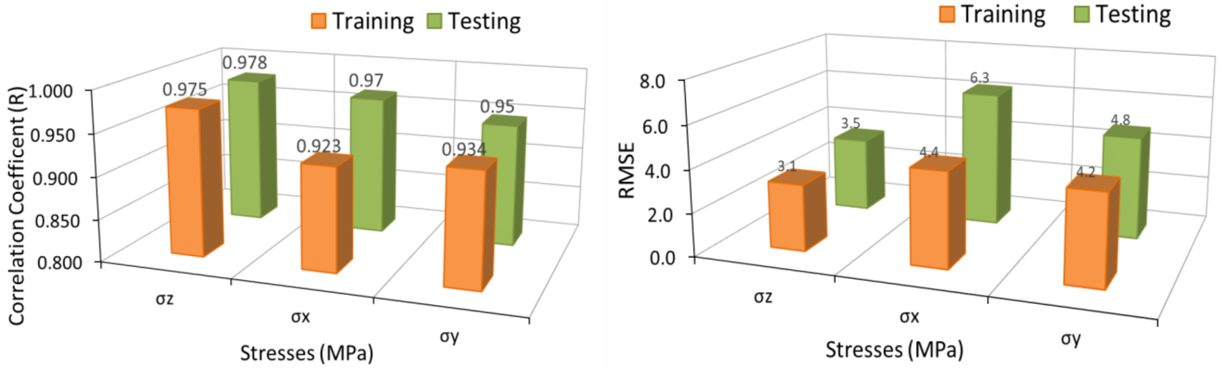

Figure 18: Comparison of accuracy measures for the proposed three stress 'σ_z', 'σ_x', and 'σ_y' predictive models.

7. Conclusions

The project Milestone comprised of training an ML model on laboratory triaxial ultrasonic velocity data for Utah FORGE well 16A(78)-32 (Granitoid formation, samples from 5474'-5850' MD) has been completed. In completing this milestone, there have been developed three machine learning models intended to eventually be deployed on field sonic log data for estimation of all three principal stress magnitudes as they distribute along well 16A(78)-32. The present Milestone marks a prerequisite that enables the work on this next part of the work. The main finding of this task is that the ML models are capable of estimating the vertical stress as a function of ultrasonic wave slowness and horizontal stresses as a function ultrasonic wave slowness and vertical stress. This conclusion is developed based on promising accuracy indicators of both training and testing outcomes reflected the reliability and robustness of the predictive models in terms of low RMSE and high correlation coefficient. Hence, the developed mathematical correlation can be employed with reasonable confidence for stress estimation across the samples zone provided that the same input and output features and ranges are used. The mathematical model has to be recalibrated if prediction is required outside the mentioned ranges of input and output features.

Overall, the best prediction performance was observed for vertical stress as compared to two horizontal stresses. This is expected because the wave velocity data was limited to waves propagating in the vertical direction in order to match the limitations of the field sonic log data. Nonetheless, the predictions of the horizontal stresses using only vertically-propagating wave velocities (or equivalently, slownesses) and the well-constrained vertical stress are of acceptable quality and provide a promising path forward for stress predictions as a part of the next tasks in this work.

8. Acknowledgement

This work was performed at the University of Pittsburgh with support via Subcontract No. 845391 to Battelle Memorial Institute for Utah FORGE Project 2439, Prime Contract No. DE_EE0007080. Additional support for APB is provided by the RK Mellon Faculty Fellowship in Energy. All of these contributions are gratefully acknowledged.

9. Nomenclature

p	=	Neurons count for input layer
q	=	Neurons count for hidden layer
i_p	=	Input features after normalization
W_{pq}	=	Weight connecting input and hidden layer neurons
W_{qr}	=	Weight connecting hidden and output layer neurons
f	=	Activation function
b_r	=	bias values for output layer
b_q	=	bias values for hidden layer
N	=	Feature value after normalization
H_n	=	Total count of hidden layer neurons
N_p	=	Total count of input features
n_{hq}	=	Neuron at qth position in central hidden layer
ψ_{min}	=	Factor used to normalize the features
ψ_{max}	=	Factor used to normalize the features
h_{zz}	=	P-wave slowness in z-direction (vertical)
h_{zx}	=	S-wave slowness in z-direction (vertical) polarized in x-direction
h_{zy}	=	S-wave slowness in z-direction (vertical) polarized in y-direction
$(h_{zz})_N$	=	Normalized P-wave slowness
$(h_{zx})_N$	=	Normalized S-wave slowness in z-direction polarized in x-direction
$(h_{zy})_N$	=	Normalized S-wave slowness polarized in y-direction
$(\sigma_z)_N$	=	Normalized vertical stress output
$(\sigma_x)_N$	=	Normalized 1st horizontal stress output
$(\sigma_y)_N$	=	Normalized 2nd horizontal stress output
i	=	Input features
i_{min}	=	Input feature’s minimum value
i_{max}	=	Input feature’s maximum value
β_a	=	Original value
β_p	=	Predicted value
n	=	Count of data points

10.References

Ali, J.K. (1994). Neural networks: a new tool for the petroleum industry?. Society of Petroleum Engineers - European Petroleum Computer Conference, EPCC 1994. Society of Petroleum Engineers, pp. 233–242. https://doi.org/10.2523/27561-ms.

Anaconda, Inc. (2023). Anaconda distribution (Version 22.9.0). Retrieved from https://www.anaconda.com/products/distribution/

Avseth, P., Mukerji, T. (2002). Seismic lithofacies classification from well logs using statistical rock physics. Petrophysics 43, 70–81.

Bunger AP, Higgins J, Huang Y, Kelley M. 2023. Core-Based In-Situ Stress Estimation for Utah FORGE Well 16A(78)-32 using Triaxial Ultrasonic Velocity and Deformation Rate Analysis. Milestone Report 2.1.1 for Utah FORGE Project 2439. https://gdr.openei.org/submissions/1438

Cerezo, J., & other developers of the Spyder project. (2023). Spyder: Scientific Python Development Environment. Retrieved from https://github.com/spyder-ide/spyder

Chau, K.W. (2007). Application of a PSO-based neural network in analysis of outcomes of construction claims. Automation in construction 16 (5), 642-646.

Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Network. 2, 359–366. https://doi.org/10.1016/ 0893-6080(89)90020-8.

Hunter, J. D., & Droettboom, M. (2016). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95. doi:10.1109/MCSE.2007.58

Jones, E., Oliphant, T. E., Peterson, P., & others. (2023). SciPy: Open source scientific tools for Python. Retrieved from https://scipy.org/

MATLAB User Guide, 2022. https://matlab.mathworks.com/

McKinney, W. (2022). pandas-dev/pandas: Pandas: Powerful data analysis and manipulation tool. Retrieved from https://github.com/pandas-dev/pandas

Mohaghegh, S., Arefi, R., Ameri, S., Rose, D. (1994). Design and development of an artificial neural network for estimation of formation permeability. In: Proceedings Petroleum Computer Conference. Society of Petroleum Engineers (SPE), pp. 147–154. https://doi.org/10.2118/28237-pa.

Mozaffari, A., Azad. N.L. (2014). Optimally pruned extreme learning machine with ensemble of regularization techniques and negative correlation penalty applied to automotive engine coldstart hydrocarbon. Neurocomputing 131, 143-156.

Opper, M., & Haussler, D. (1996). Neural networks: Foundations of machine learning. Boston, MA: MIT Press.

Otchere, D.A., Ganat, T.O.A., Gholami, R., Ridha, S. (2021). Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ANN and SVM models. J. of Pet. Sci. and Eng. Vol. 200, 108182.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Python Software Foundation. (2023, May 17). Python 3.9.13. Retrieved from https://www.python.org/downloads/release/python-3913/

Saggaf, M., Toksoz, M.N., Mustafa, H.M. (2003). Estimation of reservoir properties from seismic data by smooth neural networks. Geophysics 68, 1969–1983. https://doi. org/10.1190/1.1635051

Saikia, P., Baruah, R.D., Singh, S.K., Chaudhuri, P.K. (2020). Artificial Neural Networks in the domain of reservoir characterization: A review from shallow to deep models. Computers and Geosciences, vol. 135 104357.

Waskom, M., & Seaborn, W. (2023). seaborn: statistical data visualization. Retrieved from https://seaborn.pydata.org/

Zhang, Z., & Ye, J. (2022). On the existence of infinitely many realization functions of non-global local minima in the training of artificial neural networks with ReLU activation. arXiv preprint arXiv:2202.11481.