Estimate stress for 16B(78)-32 based on sonic logging data and laboratory data, enabled by the validated ML algorithm

Report documenting completion of Milestone 2.3.2 of Utah FORGE project 2439v2: 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], Guanyi Lu[a], 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

*Corresponding author: bunger@pitt.edu

Last Updated: 20 May 2025

Table of Contents

List of Figures

List of Tables

1. Technical Summary

In this report, the deep/machine learning (DL/ML) based workflow is illustrated for the prediction of in-situ stresses i.e. minimum, intermediate, and maximum principal stresses in the subsurface geological formations in the Utah FORGE well 16B(78)-32. The workflow was completed in a sequence of four stages. In the first stage, a total of four hundred sixty-five '465' true triaxial ultrasonic velocity (TUV) experiments were performed in laboratory on five subsurface core samples (93 on each core) obtained from the geothermal production well 16B(78)-32 drilled at the Utah FORGE site. The TUV experiments involve the measurements of ultrasonic wave including compressional (P-) and two shear (S_{fast} and S_{slow}) wave velocities are measured under various combinations of true triaxial applied stress. It is important to note that ultrasonic wave velocities were measured in core samples in such direction that velocities propagate along the wellbore axis in order to imitate the actual field scenario.

In the second stage, DL/ML predictive models were developed for three principal stresses i.e vertical and two mutually orthogonal horizontal stresses by training and validation of five DL/ML algorithms using TUV experimental data. Further, the prediction performance of DL/ML models were evaluated for different portions of datasets in order to determine the minimum number of TUV experiments to be performed that are sufficient to develop reliable prediction models of three principal stresses. The analysis allows us to obtain the minimum size of TUV dataset that can be utilized to develop reliable prediction models of principal stresses in order to reduce the experimental time, labor and resources to ultimately improve the overall efficiency of the process. As illustrated, TUV experiments were performed on five different core samples, therefore distinctive DL/ML models of three principal stress were developed for each core sample. A total of three DL and two ML algorithms such as deep neural network (DNN), convolutional neural network (CNN), and gated recurrent units (GRU), random forest (RF) and extreme gradient boosting (XGB) were employed for the model development. For the training and validation of predictive model of vertical stress (σ_y), three input features including P-, S_{fast}- and S_{slow}- wave velocities i.e. V_{zz}, V_{zx}, V_{zy} were utilized as input features. For the development of predictive models of two orthogonally oriented horizontal stresses (σ_x and σ_z), vertical stress (σ_y) was also used as input feature in addition to the three ultrasonic velocities (V_{zz}, V_{zx}, V_{zy}). Then, the best DL/ML algorithm with highest prediction accuracy was selected for the in-situ stress prediction in the well 16B(78)-32. The training of DL/ML models was performed using 70% of the TUV dataset and 30% dataset was assigned for model testing/validation. The prediction accuracy of DL/ML models were evaluated using graphical and statistical metrics such as coefficient of determination (R^2), residual error (RE), root mean squared error (RMSE), and average absolute percentage error (AAPE). It is important to note that prior to the development of the DL/ML models, the TUV datasets of five core samples were thoroughly discovered and evaluated through various exploratory data analysis (EDA) techniques in order to comprehend the data distribution patterns, trends, features interrelationships, and to determine the relative importance of input features with respect to target output. Optimization of the DL/ML models was performed by tuning hyperparameters associated with each of the DL/ML algorithms. Hyperparameters associated with DNN models include number of hidden layers, neuron counts in hidden layers, optimizer, activation function, learning rate, batch size, and dropout rate that were precisely tuned to ultimately achieve the optimized prediction outcome. After the optimization process, generalization capabilities of the DNN models of three principal stresses (σ_y, σ_z, and σ_x) were also evaluated by parametric/sensitivity analysis. Parametric analysis revealed the excellent generalization capabilities of the DNN predictive models reflecting the underlying physical phenomenon through demonstrating constitutive relation between each of the input features with predicted output.

The proposed DL/ML models exhibited excellent and reliable prediction performance for the three principal stresses (σ_y, σ_z and σ_x), however best prediction accuracy was demonstrated by DNN models reflecting low prediction errors (RMSE and AAPE) and high R^2. The DNN model for stress σ_y exhibited the AAPE and RMSE of 3.43% and 2.05 MPa for training and 4.93% and 2.88 MPa for validation subsets of data. The predictions of validation and training phases reflected the R^2 of 0.98 and 0.984. For the DNN model of σ_x, validation and training results demonstrated the AAPE of 3.66% and 3.08%, RMSE of 1.89

and 1.73 MPa, and R^2 of 0.977 and 0.979, respectively. Further, DNN model performance was observed to be superior for σ_z stress with RMSE of 1.83 and 2.01, and AAPE of 4.13, and 4.97, for the training ad validation phases, respectively.

At the third stage, learning curve was evaluated to explore the performance of ML/DL models for different portions/sizes of TUV dataset starting from small portion (20%) to the full range of dataset (100%). The purpose of learning curve evaluation is to identify the data adequacy (minimum size) of TUV dataset required for reliable and generalized ML/DL models. Models' performance was observed to be improved with gradual increase in the size of dataset up to 80% of dataset (75 data points). After that, no significant improvement in the prediction accuracy was observed with further increase in the dataset size. Further, models' explainability and interpretability was evaluated in the fourth stage to comprehend the inner working of the complex ML/DL models. Interpretability of the models enhanced the scientific validation of the proposed ML/DL models by explaining the rationale behind the models' prediction.

At fifth stage, unsupervised K-means algorithm was implemented to classify the subsurface rock formations into plurality of petrofacies/rock facies (PF) based on well logging data such as gamma ray (GR), bulk density (ρ), photoelectric factor (PEF), and neutron porosity (ϕ). K-means clustering algorithm was iteratively executed for a number of petrofacies (clusters) and the optimum number of petrofacies was selected based on silhouette index (SI) and inertia values. Optimally six (06) petrofacies were identified in subsurface rock formations. The purpose of the subsurface rock classification is to identify the representative petrofacies that correspond to the subsurface core depths/locations, along the entire well section.

At the final stage, DNN models (trained on TUV data) with optimum hyperparameters setting were selected to implement for the prediction of in-situ principal stresses in the subsurface rock formations in well 16B(78)-32 using field sonic and bulk density logs as input data. It is worth mentioning that the respective predictive models of σ_y, σ_z and σ_x stresses were employed for the prediction of vertical, minimum horizontal and maximum horizontal stresses in the field. However, the performance of DNN models was observed to be optimum to predict subsurface in-situ stresses only for the representative PF as they encapsulate the petrophysical and formations characteristics similar to core sampling locations/depths. In contrast, the performances of DL/ML models for non-representative PF were examined not to be as good as for representative PF due to attributes different from the subsurface core locations/depths. It was observed that DNN predicted in-situ stresses match closely the in-situ stresses obtained from field-based elastic geomechanical model (FB-EGM) for the representative PF as they are illustrative of the constitutive behavior of subsurface core sample. In contrast, DNN predictions of stresses were observed to be deviated from the FB-EGM stresses estimation for the non-representative PF.

The entire work was completed using different computational softwares embedded with Anaconda package. Open-source software, namely *Python (ver:3.12.5),* was used for the EDA analyses and DL/ML models development. Various libraries of *Python (ver:3.12.5)* such as *Pandas, Seaborn, Matplotlib,* and *SciPy, numpy, openpyxl,* and *Scikit-learn*. The codes and algorithms were executed using two on two different integrated development environments (IDE) such as *Spyder (ver:5.4.3)* and *Visual Studio Code (ver: 1.98.2)*.

In summary, evaluation metric demonstrated excellent performance of DL/ML predictive models of three principal stresses for both laboratory-based and log-based stress predictions. Parametric/Sensitivity analysis revealed excellent generalization capabilities of the proposed DNN models by capturing the variations of ultrasonic wave velocities under the influence of stress. Thus, the DNN models successfully predicted the in-situ principal stresses in the geothermal well 16B(78)-32 through interpreting sonic log data. The approach can be extended to other wells for the reliable estimation of in-situ stresses in the subsurface rock formations.

2. Task and Milestone Description

This report documents the task completion and technical accomplishments comprising achievement of Milestone 2.3.2, as per the project SOPO. The validation of milestone accomplishments is illustrated in Figures 18-21 demonstrating the key outcomes of the developed deep/machine learning models for in-situ vertical, minimum horizontal and maximum horizontal stresses.

Milestone 2.3.2 – the task incorporates the development and implementation of deep/machine learning (DL/ML) models for predicting the in-situ vertical (S_v), minimum horizontal (S_{hmin}) and maximum horizontal (S_{hmax}) stresses in the well 16B(78)-32. The detailed description about the experimental work was documented in Milestone report 2.1.1 of the project.

This Milestone task 2.3.2 follows the Milestone report 2.3.1 that elaborates the ML model development, validation and field implementation strategy comprehensively for the well 16A(78)-32. In this Milestone task, DL/ML predictive models were developed and implemented for the prediction of in-situ principal stresses i.e. S_v, S_{hmin}, and S_{hmax} in the well 16B(78)-32. The DL/ML predicted in-situ stresses were found to be in good agreement with stresses obtained from field-based elastic geomechanical model. The accomplishment of the Milestone task 2.3.2 is illustrated in figures 18-21 reflecting harmony between DL/ML-based and field-based estimation of in-situ stresses.

Milestone task was accomplished in a sequence of several stages including performance of TUV experiments, exploratory data analyses (EDA), training and validation of DL/ML models, parametric analysis of models, and field implementation for the in-situ stress prediction. This report embark with a illustration of experimental procedure and DL/ML techniques employed for the development of predictive models of stresses. Then, an integrated DL/ML based workflow is presented. The major outcomes of EDA analysis are briefly discussed for the TUV experimental datasets. Then, a comprehensive illustration of DL/ML models building, optimization, and hyperparameter tuning is provided. Subsequently, comparison of evaluation metrics of five DL/ML predictive models is presented. The parametric/sensitivity analysis is presented to evaluate the generalization capabilities of the proposed DNN models. Then, the performance of DNN models is shown for different sizes of TUV datasets. The last section demonstrates the implementation of DL/ML models on field sonic log data to predict the in-situ principal stresses in the well 16B(78)-32. Finally, the key findings and important postulates are provided in conclusions.

3. Deep/Machine Algorithms

3.1. Deep Neural Network

The DNN, an artificial neural network (ANN) with multiple hidden layers, is extensively adopted in machine learning for various tasks such as approximation, data mining, pattern recognition, and prediction (Mozaffari and Azad, 2014). DNN leverages various learning algorithms, activation functions, and network architectures to address complex engineering challenges (Mohaghegh et al., 1995). By emulating the structure and functionality of the human nervous system, DNN tackles specific problems effectively. Artificial neurons, the fundamental units of architectural networks, are configured based on the requirements of the particular engineering task (Ali, 1994). DNN uses numerous techniques to adeptly make connections between nonlinear variables in order to generate robust and reliable predictions results. DNN use a broad spectrum of techniques to link nonlinear variables, producing predictions that are accurate, reliable, and consistent (Otchere, 2021).

The two predominant types of DNN are feedback neural networks (FBNN) and feedforward neural networks (FNN) (Chau, 2007). FNN consist of interconnected layers of perceptron that facilitate unidirectional information flow from input to output, without cycles or loops. This straightforward architecture enables the transformation of input features through hidden layers using activation functions, allowing the network to learn complex patterns and generate precise outputs. On the other hand, FBNN, which share a similar structural design with FFN, incorporate a feedback mechanism (Saggaf et al., 2003).

This mechanism loops back error information, enabling adjustments to neuron weights through an iterative process. The feedback loop continues until prediction errors are minimized, enhancing the predictive accuracy of the model's target output (Saikia, 2020).

3.2. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN) were initially designed for computer vision tasks like image classification and object detection (LeCun et al., 1998). Over time, their application has extended to regression problems, particularly in domains where spatial or structural patterns in data play a crucial role. CNN excel in processing grid-like data structures, including not only images but also structured numerical or tabular data when reformulated appropriately (Goodfellow et al., 2016). The core architecture of CNN includes convolutional layers, pooling layers, and fully connected layers. The convolutional layers extract spatial and hierarchical features from input data using filters or kernels. This operation enables CNN to capture local patterns and aggregate them into higher-level abstractions. Pooling layers, such as max pooling, reduce the spatial dimensions of feature maps, improving computational efficiency and minimizing overfitting. Finally, fully connected layers integrate these features to make predictions (Goodfellow et al., 2016).

For regression tasks, CNN utilize convolutional layers to extract features that capture underlying spatial relationships. These features are then fed into fully connected layers to map the high-dimensional representations to continuous output variables (Zhang et al., 2017). This ability makes CNN highly effective for tasks such as predicting physical properties, analyzing time-series data reformulated as matrices, and estimating real-world measurements like temperature or stress (Hosseini et al., 2021). CNN's ability to automatically learn spatial hierarchies makes them a robust alternative to traditional machine learning models for regression problems (Zhu et al., 2021; Guo et al., 2020).

3.3. Gated Recurrent Units (GRU)

The Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture introduced by Cho et al. (2014) as a simpler and computationally efficient alternative to the Long Short-Term Memory (LSTM) network. Like LSTM, GRU is designed to address the vanishing gradient problem in traditional RNNs, making it suitable for time series prediction and regression tasks. GRUs utilize two primary gates: the reset gate and the update gate. The reset gate determines how much of the previous information is forgotten, while the update gate controls the amount of new information added to the hidden state. Unlike LSTM, GRU does not have a separate memory cell; instead, it merges the memory and hidden states into a single vector. This simplified architecture of GRU is particularly well-suited for applications requiring less computational resources without compromising on accuracy (Cho et al., 2014). The dual-gate mechanism provides GRUs with the ability to improve the model's performance in tasks such as regression (Chung et al., 2014).

In regression tasks, particularly those involving sequential data, GRUs have been found to perform well by capturing temporal dependencies and providing accurate continuous predictions. Since GRUs are computationally more efficient and faster than their counterpart, LSTM networks, and are often preferred when the task involves less complex data or when computational resources are constrained (Yin et al., 2017). Additionally, GRUs are widely used in domains like speech recognition, weather forecasting, and natural language processing for regression analysis, due to their ability to learn both long-range and short-range dependencies efficiently (Chung et al., 2014). Thus, GRU represents a powerful tool for modeling tasks, offering a balance between simplicity, efficiency, and accuracy, which makes it a widely adopted model in machine learning research and applications (Zhou et al., 2016).

3.4. Random Forest (RF)

Random Forest (RF) is a robust supervised machine learning algorithm that employs ensemble learning techniques to address both regression and classification tasks. Ensemble learning combines predictions from multiple machine learning algorithms to achieve superior predictive performance compared to individual models. In the case of RF, it constructs an ensemble of unpruned classification or

regression trees by using bootstrap samples of the training data and random feature selection during tree induction. RF derives its predictions by averaging the class outcomes from all these trees. This approach enhances the model's accuracy and robustness, making it a valuable tool in various predictive modeling applications (Palmer et al., 2007). For a comprehensive understanding of the RF methodology, refer to the work by Svetnik et al. (2003).

3.5. Extreme Gradient Boosting (XGB)

The XGB is also an ensemble supervised ML technique designed for both classification and regression tasks. It represents an advanced and scalable implementation of the gradient boosting framework, initially developed by Friedman et al. (2000). Introduced by Chen et al. (2015), XGB aims to enhance the prediction and generalization capabilities of traditional boosting methods. Generally, gradient-based approaches rely solely on the first-order partial derivative of the loss function to determine error direction, however, XGB incorporates both first and second-order partial derivatives, providing a more comprehensive understanding of the gradient's direction. This inclusion of second-order derivatives allows for more precise adjustments during the boosting process. Additionally, XGB integrates L1 (lasso) and L2 (ridge) regularization techniques, which help in preventing overfitting and contribute to the development of a more generalized model.

The algorithm's efficiency is further enhanced by its ability to handle missing values inherently, streamlining the data preparation process. Moreover, XGB's architecture supports parallel and distributed computing, offering significant improvements in computational speed over other gradient-based algorithms. These features collectively contribute to XGB's exceptional predictive performance, as reported by several researchers (Wang et al., 2020; Yang et al., 2017; Zhao et al., 2018).

3.6. K-Means Clustering (K-means)

In the 1950s and 1960s, an unsupervised ML method namely K-means clustering algorithm was introduced by various researchers (Steinhaus, 1956; MacQueen, 1967; Jancey, 1966). This algorithm partitions data into predetermined number of clusters. Normally, arbitrary clusters count is required as an input parameter which is used to select the centers of arbitrary clusters of the dataset (Jain, 2010).

The K-means clustering algorithm initiates by randomly selecting a predetermined number of centroids within the dataset. Each data point is then assigned to the nearest centroid based on calculated distances, typically using Euclidean distance. Following this assignment, centroids are recalculated as the mean of all data points within each cluster. This iterative process continues until cluster memberships stabilize, meaning that assignments no longer change between iterations. K-means is widely utilized across various fields due to its robustness and straightforward implementation (Jain, 2010). The primary objective of K-means is to minimize the sum of squared errors within clusters, enhancing intra-cluster similarity (Jain, 2010; Drineas et al., 1999). Notably, increasing the number of clusters generally leads to a decrease in the sum of squared errors, as each cluster can more precisely represent a subset of the data (Ahmad & Khan, 2019).

4. Methodology

4.1. Modelling Workflow

This report presents an integrated DL/ML-based approach that combines supervised and unsupervised DL/ML algorithms for estimating in-situ stresses in the subsurface rock formations in well 16B(78)-32. Four stages are involved in the accomplishment of this work. At the first stage, true triaxial ultrasonic velocity experiments (TUV) were performed in the laboratory on subsurface core samples, retrieved from five different locations of the well 16B(78)-32 drilled at Utah FORGE geothermal site. Subsurface cores represent the rock types including gneiss, granite, granodiorite, and quartz gneiss. A total of 465 TUV experiments were performed on five core samples with 93 experiments on each core. In each TUV

experiment, compressional (P) and two shear (S_{fast} and S_{slow}) wave velocities under various combinations of true triaxial stresses were measured along the wellbore axis.

The second stage of the workflow comprised of exploratory data analysis (EDA), developing optimized DL/ML predictive models, and parametric study of the models using the TUV experimental data. For EDA, various visualization and numerical techniques were used to explore and present the data distribution, pattens, relation between the features, and relative importance of features. Further, TUV experimental data was used to construct (training and validation) predictive models for the vertical (σ_z) and two mutually orthogonal horizontal stresses (σ_x and σ_y) using five versatile DL/ML algorithms with three algorithms belong to DL class and two belong to ML class of AI. Algorithms include deep neural network (DNN), gated recurrent units (GRU), convolutional neural networks (CNN), random forest (RF), and extreme gradient boosting (XGB). Individual predictive models were constructed and optimized for each subsurface core-based TUV data. Extensive numerical experimentation was performed for tuning the hyperparameters of each DL/ML predictive models. A powerful computing tool namely Python was utilized for constructing the DL/ML models.

At the third stage, learning curve was evaluated to assess the performance of ML/DL models for different sizes of TUV dataset starting from low portion (20%) to the full range of dataset (100%). The purpose of learning curve evaluation is to identify the data adequacy (minimum size) of TUV dataset required for reliable and generalized ML/DL models. Further, models' explainability and interpretability was evaluated in the fourth stage to comprehend the inner working of the complex ML/DL models.

At the fifth stage, petrophysical attributes obtained from well log data of the geothermal well 16B(78)-32 were used to classify the subsurface rock formations into plurality of petrofacies through the unsupervised ML algorithm namely K-means clustering. Further, the representative petrofacies were recognized that correspond to the subsurface core locations for the entire continuous interval of the well with measured depth 'MD' ranging from 4835 to 10872 ft. Thus, this stage incorporates the steps of acquiring of petrophysical attributes from the well logs, classifying petrofacies, and recognizing the representative petrofacies.

Subsequently, in the last stage, in-situ horizontal and vertical stresses in the subsurface rock formations were predicted by employing the selected predictive DL/ML models with best evaluation metrics using the field-based acoustic log data (P- and S-wave velocities) of the same geothermal well (16B(78)-32). All stages of the workflow adopted in this study is presented in form of workflow diagram as illustrated in Figure 1. Experimental procedure and a quick overview of the DL/ML techniques used in this work is provided before the detailed discussion of these stages.

The workflow was completed using different libraries of open-source software *Python (ver:3.12.1)* (Python software foundation, 2023) and program codes were executed on two integrated development environments (IDE) including *Spyder (ver:6.0.1)* (Cerezo et al. 2023) and *Visual Studio Code (ver:1.87)* (Microsoft,2015). The *Pandas (ver:2.2.3)* library of *Python* (McKinney, 2022) was used to extract the *Microsoft Excel* data and generate data frame on *Python* software. The *Seaborn (ver:0.13.2)* 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:3.10.1)* library of *Python (ver:3.12.1)* software (Hunter and Droettboom, 2016). Statistical features and correlation coefficients (Pearson, Spearman, and Kendall) were determined using *Scipy (ver:1.15.2)* library (Jones et al. 2023). After EDA analysis, dataset is ready to be fed for ML modelling. Training and validation of DL/ML models were performed using scikit-learn *(ver:24.10-1)* (Pedregosa et al. 2011) library of Python software. All the Python libraries and IDE (Integrated Development Environment) collectively function under the *Anaconda (ver:24.10-1)* package (Anaconda Inc. 2023).

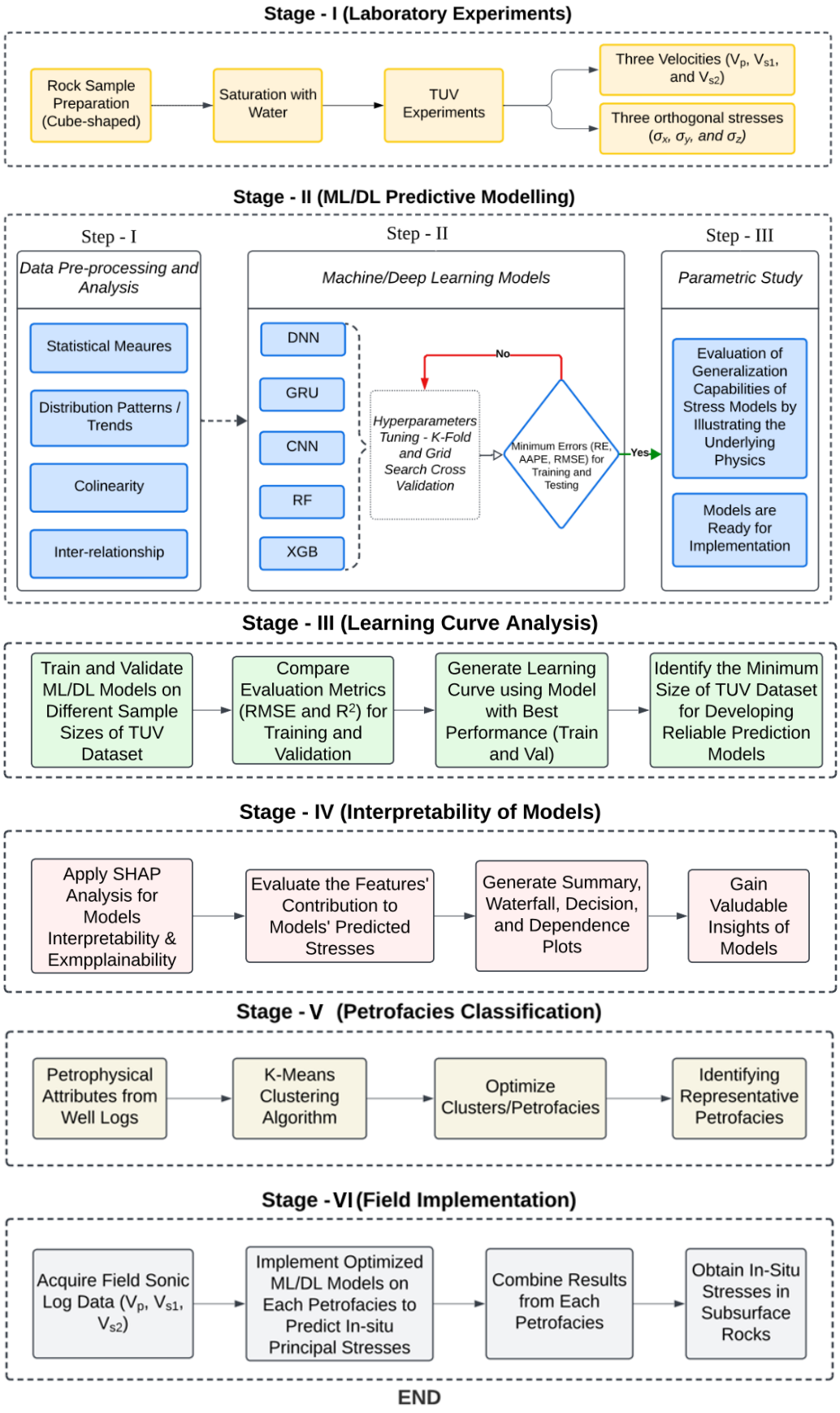


Figure 1: Workflow diagram of the tasks completed in this Milestone 2.3.2.

4.2. Laboratory Experiments – True Triaxial Ultrasonic Velocity (TUV) Experiments

A plurality of true triaxial ultrasonic velocity (TUV) experiments were performed on subsurface core samples in order to generate adequate dataset for each core sample to develop robust ML/DL predictive models for vertical and two horizontal principal stresses. Subsurface core samples were acquired from five different locations of the well 16B(78)-32 drilled at Utah FORGE geothermal site.

Laboratory experiments were completed in different steps such as sample preparation, saturation, and performance of TUV experiments. Core samples were carefully prepared through cutting and end-face grinding of core samples to finally prepare cube-shaped samples with dimensions of 2.6 inches cube. Final samples were prepared ensuring even and smooth sides with 0.001 inches tolerance and perpendicularity of sides. The core samples were then saturated with water under the vacuum pressure of -100 kPa for 24 hours. Subsequently, TUV tests were performed on saturated core samples under to imitate the subsurface reservoir conditions. TUV experiments include the measurements of travel times of compressional (P-) and two orthogonally polarized shear (S-) waves under various configurations of true triaxial stresses. Travel times were recorded using acoustic transducers that were installed in steel platens on each side of the core samples for transmitting and receiving the ultrasonic pulses. P- and S-waves velocities were computed using the measured travel times and dimensions of each cube-shape sample.

The P- and S-waves velocities propagating in z-direction were utilized in this study. Three independently controlled triaxial compressive stresses (referred to as σ_z, σ_x and σ_y) were employed along three mutually orthogonal axes (i.e. z, x, and y directions) of cube-shaped samples, respectively. The σ_y, σ_z and σ_x represent three orthogonally oriented principal stresses in TUV experiments. Here, P-wave velocity is denoted by V_{zz} and two S-waves velocities are represented by V_{zx} and V_{zy} that are polarized in x and y orientations, all propagating in z-direction. Generally, velocities V_{zx} and V_{zy} are referred to fast and slow-shear wave velocities. The triaxial stresses orientations in TUV experiments are shown in Figure 2. Further details about the experimental work was documented in Milestone report 2.1.1 of the project.

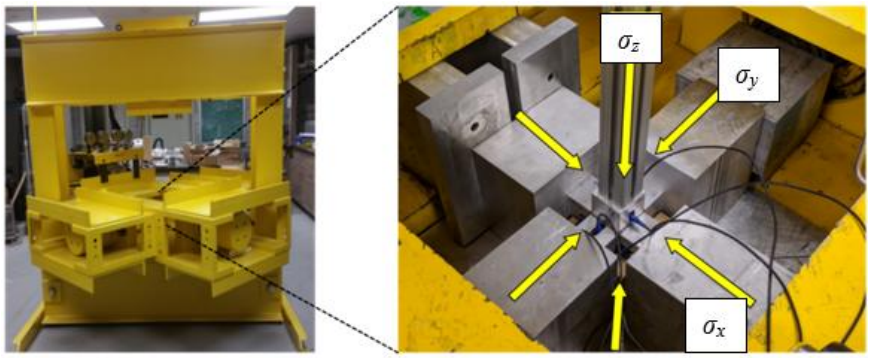


Figure 2: Experimental setup for TUV test; a) Triaxial loading system; b) Rock sample installed in loading frame (after Bunger et al. (2024))

Note that throughout this report, velocities 'V' are given two subscripts, where the first gives the propagation direction and the second gives the direction of particle motion. Hence V_{zz}, V_{zy}, and V_{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 velocities, respectively. The illustration of waves propagation direction and particle movement is provided in Figure 3.

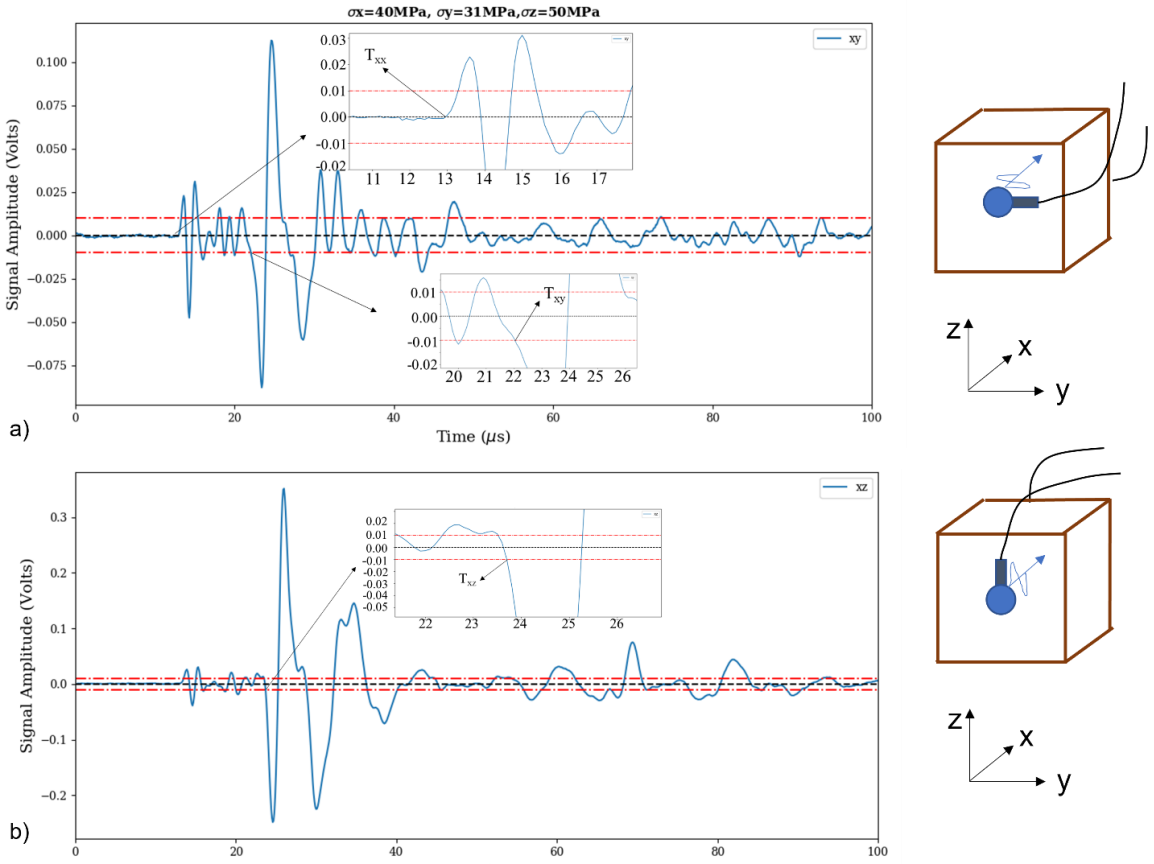


Figure 3: Example of waveforms from Lower Granitoid, x-direction. Waveforms result from two separate experiments with transducers oriented to capture the y-polarization (top) and z-polarization (bottom). Sketches show sample and transducer orientation to illustrate propagation direction and shear wave polarity for each of these cases. (Bunger et al. 2023).

5. Data Description

Two different kinds of datasets were utilized to complete this work such as laboratory-based experimental data and field-based log data. The experimental data was utilized to train and validate/test the DL/ML models. This data contains compressional (P-) and fast and slow shear (S_{slow}- and S_{fast}-) waves velocities i.e. V_{zz}, V_{zx}, and V_{zy}, respectively, measured on saturated subsurface core samples under true triaxial stress configurations. Experimental data was generated using five subsurface core samples collected from five depth locations of well 16B(78)-32 at the FORGE geothermal site. The core samples represent different rock formations such as Gneiss, Granite, Granodiorite, and Quartz gneiss. The measured depths of the core samples are provided in Table 1. A suite of data was generated from 465 TUV experiments performed on five core samples (93 on each core) which was used to develop (train and validate) DL/ML models.

In reference to field scenario, core samples are oriented in such a way that σ_y in TUV experiments represents vertical whereas σ_z and σ_x represent two orthogonal horizontal stresses. The DL/ML predictive models for stress σ_y were generated (trained and validated) using three input features including V_{zz}, V_{zx}, and V_{zy}. Whereas four input features such as V_{zz}, V_{zx}, V_{zy}, and σ_y were selected to train and validate DL/ML predictive models for horizontal stresses (σ_x and σ_z). It is important to notice that an additional input feature i.e. vertical stress σ_y was selected for the prediction of horizontal stress. The significance of selecting vertical stress (gravitational stress) as input feature for predicting horizontal stress is quite obvious because the gravitational stress (overburden) is one of the major contributors to generate horizontal stresses in subsurface rock formations. It is important to note that distinct ML models were generated for vertical and horizontal stresses of each core sample. The experimental dataset of core-A used in this work is provided

in Figure 4. TUV datasets of core-B, -C, -D, and -E are provided in Appendix-A.

Further, third and fourth stages of this study was completed using the field logging measurements acquired for the measured depth interval of 4835-10872 ft of the well 16B(78)-32. In third phase, formation and petrophysical attributes obtained from logging data such as bulk density (ρ), neutron porosity (ϕ), photoelectric factor (PEF), and gamma ray (GR) was used to classify the subsurface rocks into rock facies. Subsequently, field sonic and bulk density logs of the same well were utilized to ultimately execute the field implementation of the generated DL/ML models.

Table 1: Depths and photos of subsurface core samples used for TUV experiments

Rock Type	Core ID	Measured Depth 'ft'	Density 'g/cc'	Porosity '%'	Dynamic Young's Modulus 'GPa'	Dynamic Poisson's ratio	Photo cube sample
Gneiss	A	10,438	2.65 – 2.68	2.0	73.5	0.29	
Granite	B	10,253	2.56	2.0	66.4	0.28	
Granodiorite	C	10,264	2.55 – 2.57	3.0	66.3	0.27	
Quartz Gneiss	D	9,839	2.68	2.0	79.7	0.30	
Quartz Gneiss	E	9842.3	2.68	2.0	80.4	0.29	

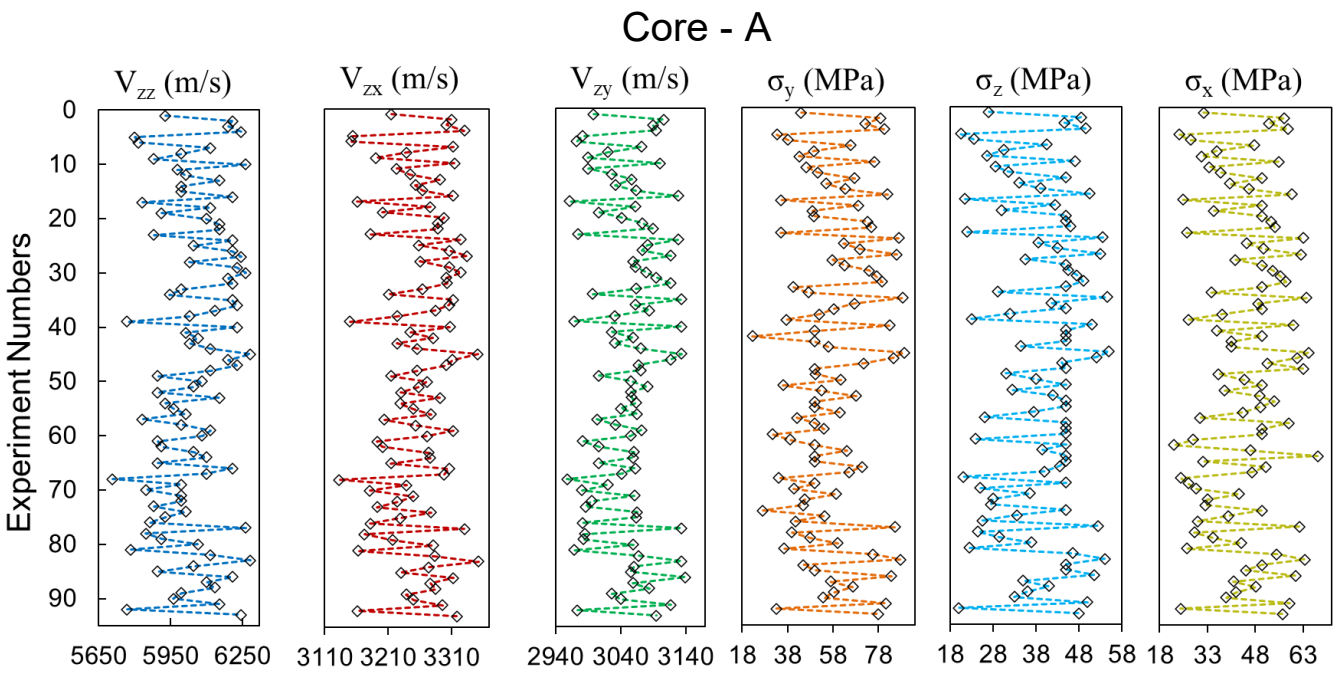


Figure 4: A suite of TUV data used for DL/ML models development.

5.1. Exploratory Data Analysis (EDA)

A comprehensive exploratory data analysis (EDA) was performed on the dataset before constructing DL/ML models. EDA offered a thorough insight and intuition about the dataset in terms of statistical features, data distribution patterns and trends, inter-correlation between the features, and relative significance input features. Statistical parameters such as median, mode, mean, range, standard deviation, kurtosis, and skewness are illustrated in Table 2. Further, histogram plots of input and output features illustrating the data distribution and patterns of dataset of core-A are shown in Figure 5. Histogram plots of dataset of cores-B, -C, -D, and -E are provided in the Appendix-B. Additionally, violin plots are provided to visualize the data distribution in terms of kernel density estimation function in order to demonstrate the extreme values, inter-quartile range, arithmetic means, and distribution type (uni-, bi-, or multi-modal). Figure 6 presents the violin plots of core-A. Violin plots of cores-B, -C, -D, and -E are provided in the Appendix-C. The heat maps are presented to exhibit the collinearity between each pair of the features of dataset of core-A as shown in Figure 7. Violin plots of the datasets of cores-B, -C, -D, and -E are provided in the Appendix-D. Heat maps were produced using three different criteria such as Kendall, Pearson, and Spearman. Most of the features exhibited positive correlation with each other. Mathematical expressions of Kendall, Pearson, and Spearman criteria are provided as

$$\rho_{pearson} = \frac{k\sum xy - (\sum x)(\sum y)}{\sqrt{k(\sum x^2) - (\sum y)^2}\sqrt{k(\sum y^2) - (\sum y)^2}} \qquad \text{(Eq. 1)}$$

$$\rho_{spearman} = \rho_{pearson}\frac{\mathrm{cov}(x,y)}{\gamma_x \gamma_y} \qquad \text{(Eq. 2)}$$

$$\tau_{kendall} = \frac{n_c - n_d}{n(n-1)/_2} \qquad \text{(Eq. 3)}$$

Here, x and y represents the respective variables, and the number of samples is shown by k. Standard deviations of the x and y variables are denoted as γ_x and γ_y, and covariance as $\mathrm{cov}(x, y)$. Numbers of discordant and concordant pairs of variables are denoted by n_d and n_c, number of data samples as n, respectively.

Table 2: Statistical measures of the dataset of core - A

Statistical Indicators	σ_x (MPa)	σ_y (MPa)	σ_z (MPa)	V_{zz} (m/s)	V_{zx} (m/s)	V_{zy} (m/s)
Maximum	22.5	22.5	20.0	5708.6	3132.8	2958.1
Minimum	67.5	89.4	55.0	6282.9	3352.5	3139.5
Median	47.0	54.4	43.0	6046.7	3264.8	3059.4
Mode	50.0	50.0	45.0	6207.7	3250.3	3133.9
Mean	44.9	57.8	39.3	6046.1	3258.0	3050.6
St. Dev.	11.8	16.5	9.6	143.5	52.0	47.7
Kurtosis	-1.0	-0.9	-0.9	-0.9	-0.6	-0.7
Skewness	-0.1	0.2	-0.5	-0.2	-0.5	-0.1

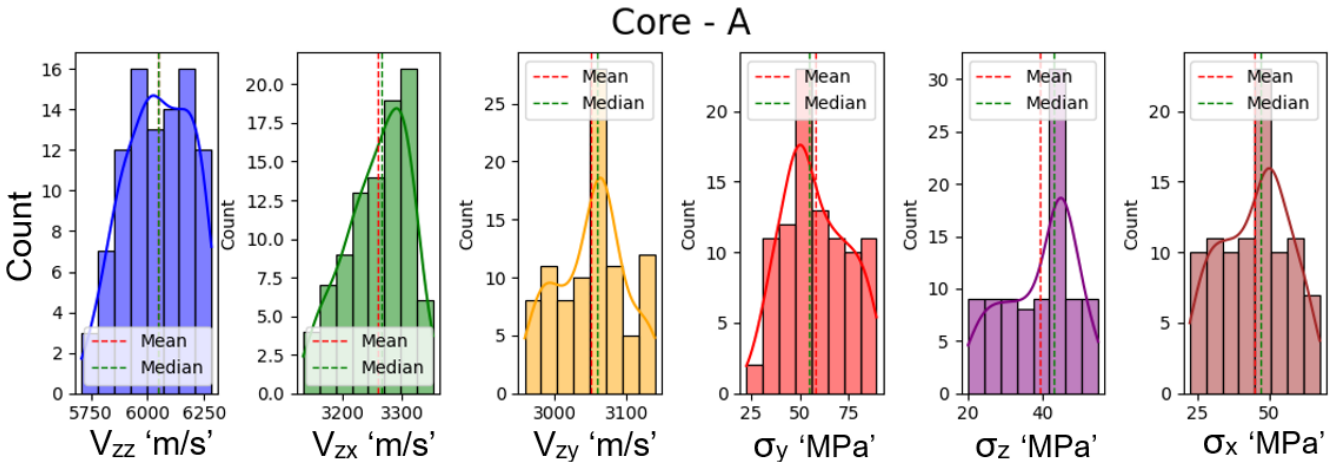


Figure 5: Histogram along with mean and median of input and output features of TUV data of core-A showing the data distribution.

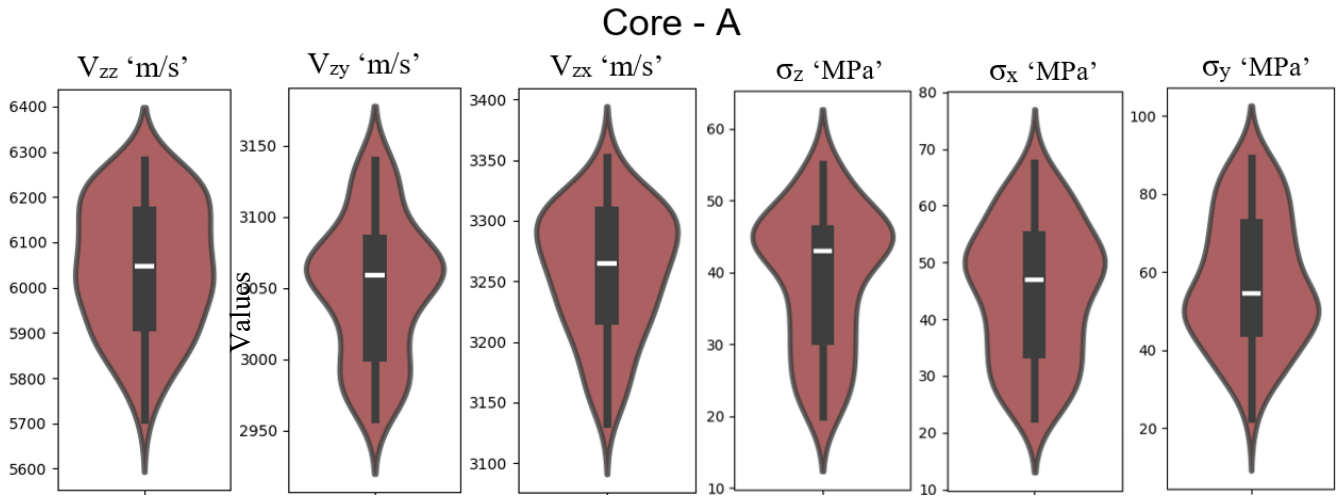


Figure 6: Violin plots for input and output data representation of core-A.

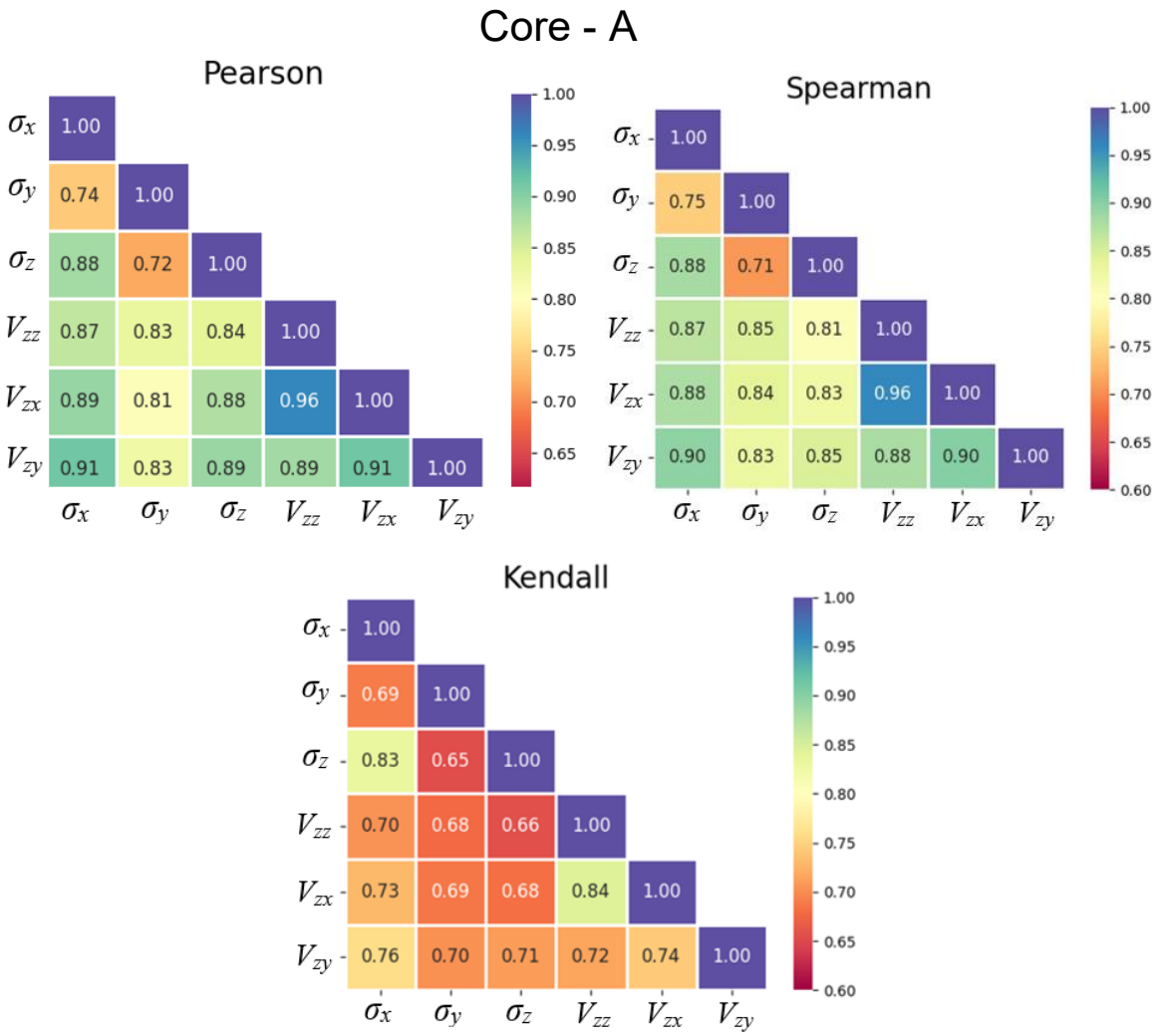


Figure 7: Heat maps showing collinearity between the features of dataset of core-A using Kendall, Spearman and Pearson criteria.

The weights of relative importance of input features with respect to output features for the σ_z, σ_x, and σ_y models of core-A are presented in Figure 8. Relative importance weights of input features in the datasets of cores-B, -C, -D, and -E are provided in the Appendix-E. The criteria for determining relative importance are provided in Eqs. 1-3. The performance of ML models' prediction is greatly affected by the score of relative importance. The input features V_{zz}, V_{zx}, and V_{zy}, were observed to be positively correlated with σ_y stress for Core-D. The features V_{zz} and V_{zx} exhibited a relatively stronger positive correlation with σ_y stress as compared to V_{zy}. Likewise, feature V_{zy} also showed a relatively weaker positive correlation with σ_x and σ_z stresses compared to V_{zz} and V_{zx} velocities. Further, feature σ_y were found to have very strong correlation with σ_z and σ_x stresses. Features V_{zx} and σ_y exhibited stronger correlation with both σ_z and σ_x stresses compared to other two features V_{zz} and V_{zy}. It is important to note that the direct positive correlation of ultrasonic wave velocities with stresses applied in the directions of wave propagation is well-known fact.

Subsequently, cross plots between each pair of input and output features are presented in a single plot termed as pair plot. Cross plots reflect that the V_{zz}, V_{zx}, and V_{zy} velocities are substantially influenced by the applied stresses. Cross plots between the features are shown in the lower triangle whereas KDE distribution of features is demonstrated along the diagonal of the pair plot as shown in Figure 9.

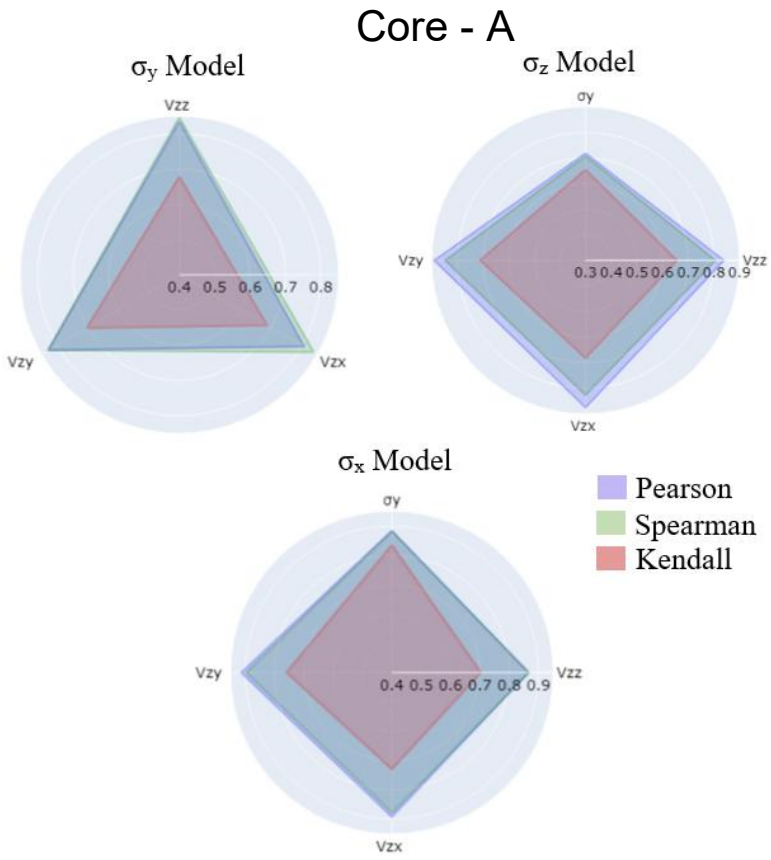


Figure 8: Relative importance of input features from the correlation between input and output features.

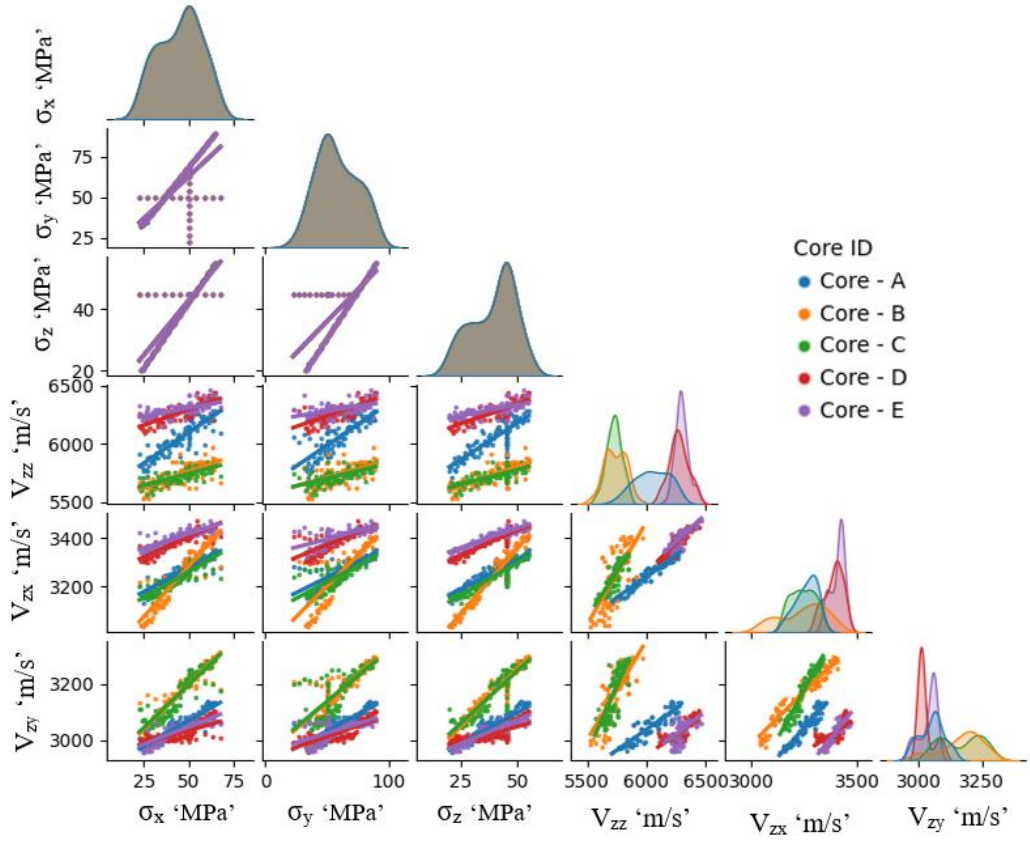


Figure 9: Pair plots exhibiting relationship between each pair of input and output features.

6. Evaluation Metrics

Graphical and statistical metrics were used to evaluate the ML prediction performance. Evaluation metrics include coefficient of determination (R^2), residual error (RE), root mean squared error (RMSE), and average absolute percentage error (AAPE) were employed for assessing the models' performances. Evaluation metrics can be demonstrated as

$$R^2 = \left[\frac{k\sum ab - (\sum a)(\sum b)}{\sqrt{(k(\sum a^2) - (\sum a)^2)}\sqrt{k(\sum b^2) - (\sum b)^2}}\right]^2 \quad \text{(Eq. 4)}$$

Here, total number of samples are denoted by k, and variables by a and b, respectively.

$$\text{Residual Error} = \left(\sigma_{\text{measured}} - \sigma_{\text{predicted}}\right) \quad \text{(Eq. 5)}$$

$$\text{AAPE (\%)} = \frac{\sum\left|\left(\sigma_{\text{measured}} - \sigma_{\text{predicted}}\right)/\frac{100}{\sigma_{\text{measured}}}\right|}{\text{total number of data points}} \quad \text{(Eq. 6)}$$

$$\text{RMSE} = \sqrt{\frac{\sum\left(\sigma_{measured} - \sigma_{predicted}\right)^2}{total\ number\ of\ data\ points}} \quad \text{(Eq. 7)}$$

Here, predicted and experimental values of stress are represented by σ_{measured} and $\sigma_{\text{predicted}}$, respectively.

Further, performance of clustering model was evaluated by the Silhouette Index (SI), a widely used metric that quantifies how well each data point fits within its assigned cluster compared to other clusters (Rousseeuw in 1987). The SI considers both intra-cluster compactness (cohesion) and inter-cluster separation to determine the appropriateness of the clustering configuration. Mathematically, SI can be expressed as

$$\text{SI} = s(j) = \frac{(a(i) - b(i))}{Max\{b(i), a(i)\}} \quad \text{(Eq. 8)}$$

Here, a(i) represents the mean separation distance between a data point 'j' and the other points within its own cluster 'X_i'. Whereas b(i) shows the shortest mean distance between point 'j' and any point in a different cluster 'X_k' is denoted by 'b(i)'. The mathematical definitions of the evaluation metrics are provided by Eqs. 4-8.

7. DL/ML Models Training, Validation/Testing and Optimization

The DL/ML predictive models were trained and validated for vertical (σ_y) and two orthogonal horizontal stresses (σ_x and σ_z) using the TUV experimental dataset of corresponding core samples. It is important to note that individual DL/ML model was developed for each stress of all core samples. As mentioned earlier, three input features including V_{zz}, V_{zx}, and V_{zy} were used for the training and validation of the σ_y stress predictive model. Whereas four input features such as V_{zz}, V_{zx}, V_{zy}, and σ_y were used for the development (train and validate) of DL/ML predictive models for σ_x and σ_z. Initially, pre-processing of the dataset was performed through the Min-Max scalar to bring the features on the same scale. The pre-processed data was then divided into two subsets i.e. training and validation subsets. A stratified random sampling technique was implemented for splitting the data into two halves in order to minimize the variability and reduce the risk of biased performance of models. A train-test split feature of scikit-learn library in Python software was employed to generate the training and validation subsets of data. The data splitting was done in such a way that validation and training subsets contain 30% and 70% of the total dataset, respectively. Graphical plots were drawn for each feature against experiment number and set of numerical values of all features against experiment number is the representation of one TUV experiment.

The Grid search cross-validation strategy was adopted for tuning the hyperparameters of DL/ML models. Further explanation of Grid search cross-validation strategy can be found in Liashchynskyi and Liashchynskyi (2019). DL/ML prediction performance was optimized during the tuning process of hyperparameters through numerous training executions on the data. The reproducibility of the same prediction results for the optimized hyperparameters was ensured using the seed function. For avoiding the overfitting/underfitting during the process of model training and for further improvement in the prediction performance of the DL/ML models, k-fold cross validation technique was employed. In CV technique, training dataset was further randomly partitioned into 'k' portions or subsets termed as k-folds. Belyadi and Haghighat (2021), observed that best results are obtained for the folds values between 5 and 10. In this study, training dataset was further divided into eight random equal sized portions (folds) i.e. k=8. Seven out of eight folds were allocated for training, and one leftover fold is used for validation of the models during the training process. Each of the eight folds underwent through validation process. Then, average of prediction errors from all folds was computed for the training process.

In DNN deep learning models, foundation of the neural network is built on two fundamental blocks such as number of hidden layers of neurons, and neurons count in each layer of neurons. A typical neurons framework employed in this study is illustrated in Figure 10. Further, an important element of neural network is the activation function. Subsequently, it is imperative to implement and elaborate the sensitivity scheme to determine the optimum combination of hyperparameters. Then, the tuning of hyperparameters were performed in an order of stages to organize a logical sequence of events in the formation process of DNN model. In this work, the neuron sensitivity scheme for hidden layers was carried out with a range from 5 to 40 neurons in order to choose optimized value of neuron count. The respective RMSE and R^2 of training and validation with different number of neurons for all three DNN models of stress σ_y, σ_z and σ_x for core-A are presented in Figure 11. The most favorable neurons count was observed to be 21, 5, and 7 with least prediction RMSE and R^2. The RMSE and R^2 of validation phases with optimum number of neurons were observed to be 2.88. and 0.980 for stress σ_y, 2.01 and 0.973 for σ_z, and 1.89 and 0.977 for σ_x, respectively. Neuron sensitivity analysis of DNN models of stress σ_y, σ_z and σ_x for core-B, -C, -D, and -E are provided in the Appendix-F.

The optimization of hidden layers count, assessment of neurons count, and evaluation of activation function were performed at the model building stage at which input, hidden and output layers are defined. An additional dropout layer was embedded in between input and hidden layers. Then, the compilation of the DNN model was performed by carefully selecting the suitable optimizers, learning rate, loss function and accuracy metric for optimization and performance evaluation of the DNN models. Subsequently, models training were simulated with a different set of hyperparameters for numerous epochs and different batch sizes. Ultimately, optimum sets of hyperparameters were selected for training of DNN models as illustrated in Table 3. Further, to avoid the performance degradation on validation dataset during the training process, early stopping function was applied.

Additionally, a crucial phase in the construction of optimized models is to run several realizations in order to select best seeding functions that shows a distinctive identity affiliated to the DNN code of this study. A total of 100 realizations were performed for the DNN models of stresses σ_y, σ_x and σ_z. Integration of optimum values of hyperparameters complemented with the best realizations ultimately led to the construction of fully optimized DNN models of stresses σ_y, σ_x and σ_z. The mismatch between the experimental and DNN predicted stresses was reflected by the selected loss function i.e. mean squared error.

In addition to DNN, four predictive models including two DL models i.e. GRU and CNN and two ML models i.e. RF and XGB were also trained and validated on the same TUV dataset. The optimum values of associated hyperparameters were achieved during the training of the predictive models. Optimization of hyperparameters of each DL/ML model was conducted by Grid Search CV technique.

The GRU architecture was also constructed through a systematic procedure. After the data pre-

processing and data splitting, GRU architecture was devised, initiating with a first GRU layer (input layer), followed by another GRU layer, then by a dropout layer to avoid overfitting of data and finally the output layer to generate the predictions. Further, model compilation, training and validation process is more or less similar to DNN model. For optimization of hyperparameters, a range of values was explored such as GRU units (10-200), dropout rate (0.1-0.5), batch size (32-64), and learning rates (0.001-0.1) through grid search CV approach. After selecting the optimum combination of hyperparameters based on minimum loss function, resulting in a final architecture. Subsequently, optimized model was employed to predict the validation/test subset of data. The optimized values of hyperparameters are illustrated in Table 3.

In this work, the design of CNN architecture was developed to explore its effectiveness in resolving regression problems using tabular dataset. The architecture is composed of input layer, followed by the two 1D convolution layers (Conv1D) with number of filters, kernel size, and ReLU activation function. The output of Conv1D layers passed through the flatten layer, and then fully connected (dense) layer with ReLU activation function. Subsequently, the final output layer with single neuron and linear activation function was applied to obtain the continuous predictions. The optimum values of hyperparameters was obtained after exploring the range of values such as kernel sizes (2-10), number of filters (8-64), dense units (8-64), learning rate (0.001-0.01), batch sizes (8-64), optimizer (adam, nadam, sgd) through grid search CV approach. The optimum values are selected based on minimum loss function. The optimized CNN model was then employed for predicting validation dataset reflecting its generalization capabilities on unseen dataset.

Further, two ML predictive models such as RF and XGB were developed. Likewise, grid search CV strategy was applied to optimize the hyperparameters for each of the ML models. The prediction performances of all five DL/ML models were then compared to choose the best predictive model for field implementation. The optimized values of hyperparameters of all DL/ML models are presented in Table 3.

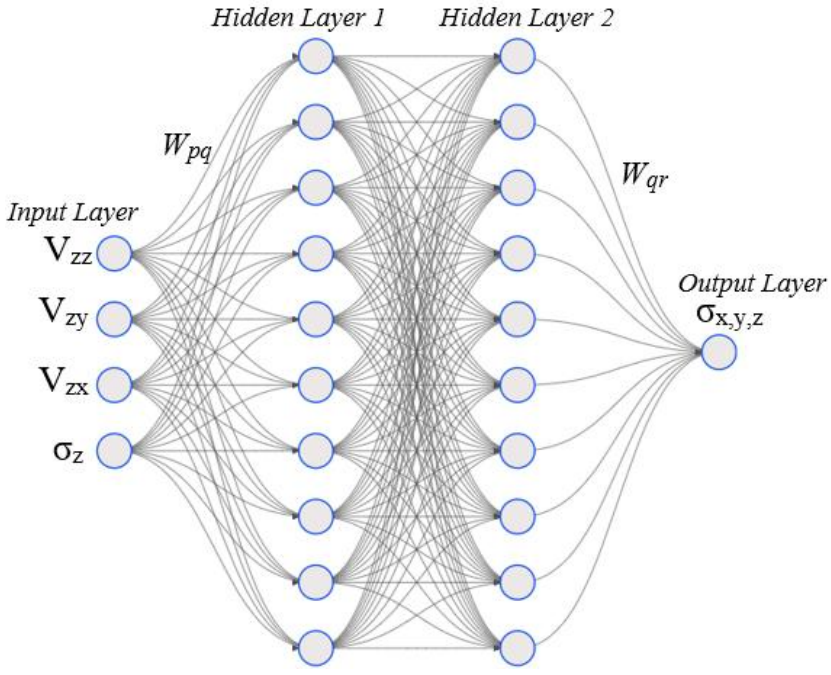


Figure 10: DNN topology showing neurons structure.

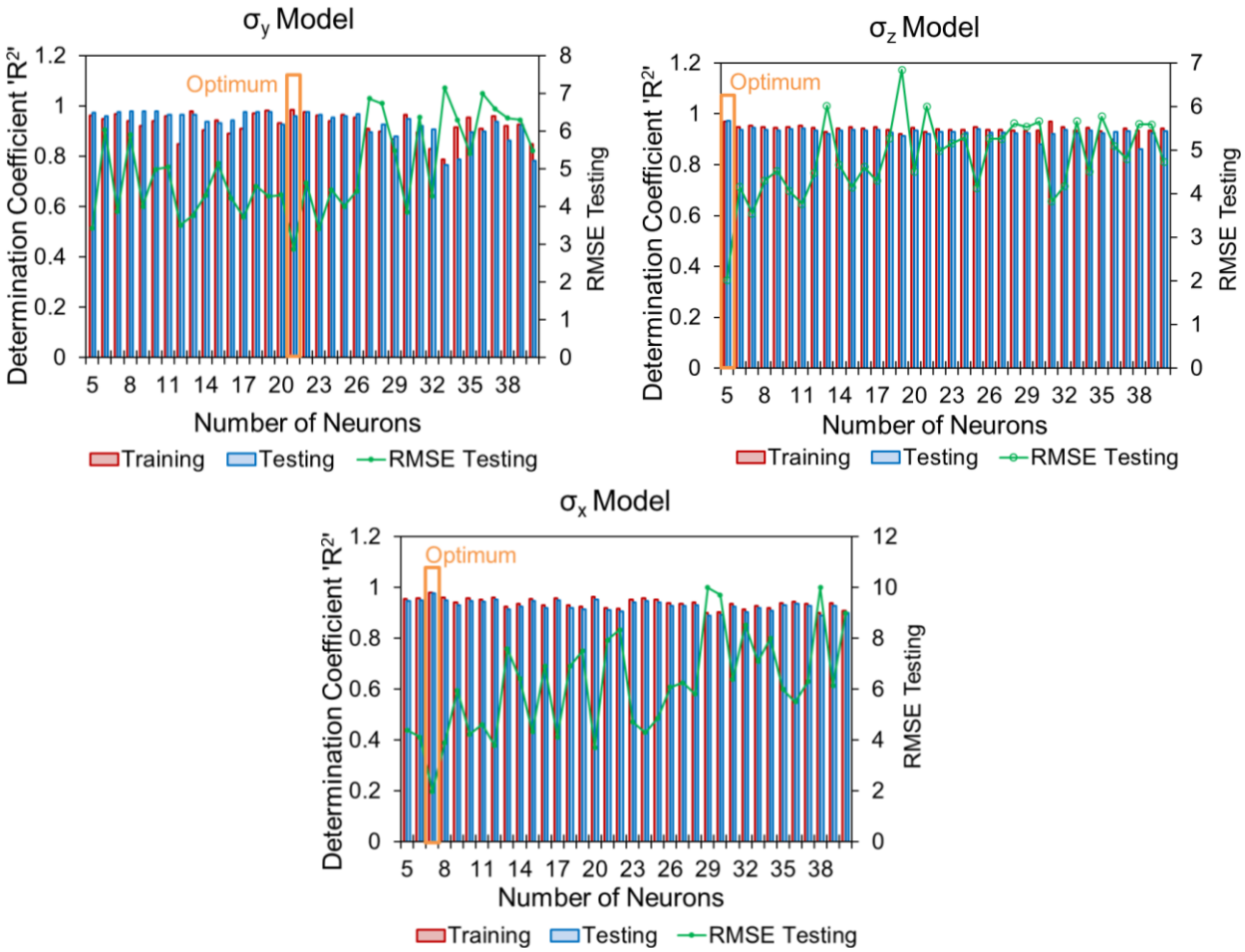


Figure 11: Neurons sensitivity analysis and selection of best realization using evaluation metrics of σ_z, σ_x, σ_y models.

Table 3: Optimum values of hyperparameters of the proposed DL/ML models for Core-A.

Model Type	Hyperparameter	Tested Values	Selected value for 'σ_z'	Selected value for 'σ_x'	Selected value for 'σ_x'
DNN	Number of hidden layers	1-3	2	2	2
	Number of neurons	5-40	21	5	7
	Number of realizations	1-100	50	58	24
	Optimizer	Adam, nadam, sgd	Adam	Adam	Adam
	Activation function	ReLU, LeakyReLU, Tanh	ReLU	ReLU	ReLU
	Learning rate	$1x10^{-4}$ – $1x10^{-1}$	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-3}$
	Batch size	8, 16, 32, 64	16	16	16
	Dropout Rate	0.1-0.5	0.2	0.2	0.2
	Loss Function	MSE	MSE	MSE	MSE
GRU	GRU Units	10-200	50	70	60
	Activation Function	ReLU, LeakyReLU, Tanh	Tanh	Tanh	Tanh
	Dropout rate	0.1-0.5	0.2	0.2	0.2
	Batch Size	8, 16, 32, 64	32	32	32
	Learning rate	$1x10^{-3}$ – $1x10^{-1}$	$1x10^{-2}$	$1x10^{-2}$	$1x10^{-2}$
	Optimizer	Adam, nadam, sgd	Adam	Adam	Adam
	Loss Function	MSE	MSE	MSE	MSE
CNN	Number of Filters	8-64	8	16	16
	Kernel Size	2-10	3	3	3
	Activation Function for Convolution layer	ReLU, LeakyReLU, Tanh	ReLU	ReLU	ReLU
	Dense Units	8-64	32	32	32
	Batch Size	8, 16, 32, 64	16	16	32
	Optimizer	Adam, nadam, sgd	nadam	nadam	nadam
	Loss Function	MSE	MSE	MSE	MSE
RF	Sample count necessary to split the internal node	2-20	10	12	10
	Maximum depth of the trees	3-10	7	6	7
	Total count of trees in the forest	50-2500	1200	1100	1400
	Sample count at the leaf node	1-5	2	2	2
XGB	Criterion	-	friedman mse	friedman mse	friedman mse
	Count of boosting stages to be performed	100-1000	200	220	200
	Learning rate	0.001-0.1	0.01	0.01	0.01
	Alpha	-	0.03	0.03	0.02
	Minimum sample split	1-5	2	2	2
	Max. depth	3-10	7	8	7
	Minimum sample leaf	1-5	2	2	2
	Loss	MSE, RMSE	MSE	MSE	MSE

8. Predictive Modelling Results

All five proposed models DNN, GRU, CNN, RF, and XGB demonstrated excellent prediction performances with low errors and high R^2 values. However, comparison of accuracy measures revealed that DNN models outperformed other four predictive models for stresses σ_z, σ_x, and σ_y.

The DNN model for stress σ_y exhibited the AAPE and RMSE of 3.43% and 2.05 MPa for training and 4.93% and 2.88 MPa for validation subsets of data. The predictions of validation and training phases reflected the R^2 of 0.98 and 0.984. For the DNN model of σ_x, validation and training results demonstrated the AAPE of 3.66% and 3.08%, RMSE of 1.89 and 1.73 MPa, and R^2 of 0.977 and 0.979, respectively. Further, DNN model performance was observed to be superior for σ_z stress with RMSE of 1.83 and 2.01, and AAPE of 4.13, and 4.97, for the training ad validation phases, respectively. The cross plots between the predicted and experimental values of stresses demonstrated the prediction performances of training and validation phases of the proposed DL/ML models of stresses σ_z, σ_x, and σ_y as illustrated in Figure 12. Evaluation metrics of the DL/ML models are also illustrated and compared on the cross plots. The predicted values of stresses values were also compared with experimental values as expressed in Figure 13-15. A good harmony was observed between the experimental and predicted values of stresses σ_z, σ_x, and σ_y. The histograms and KDE of the corresponding RE of the proposed DNN models demonstrated excellent prediction capabilities of the proposed DNN models as presented in Figure 16. The bell-shaped histogram and KDEs showed that lowest RE of the DL/ML models correspond to the highest frequencies in histograms plots that reflected their outstanding prediction performance.

A comparison of evaluation metrics of proposed DL/ML models of stress σ_y revealed the superior prediction performance of DNN with lowest AAPE (4.93%) and RMSE (2.88 MPa) and highest R^2 (0.98) values observed for validation phase. The prediction performances of other DL/ML models such as GRU, CNN, XGB, and RF are also excellent on validation dataset that reflected their generation capabilities on unseen dataset. However, metrics of GRU, CNN, XGB, and RF are slightly lower compared to DNN models. For the σ_x and σ_z stress models, the best validation performances were observed for DNN model with minimum RMSE (1.89 and 2.01 MPa) and AAPE (3.66 and 4.97%) and highest R^2 (0.977 and 0.973), respectively. Although all predictive models exhibited good performance during the training phase with marginal differences in the evaluation metrics. Hence, the overall performance of DNN was outstanding and slightly better than other DL/ML models. The relatively higher prediction errors RMSE (4.17, 2.63, and 3.78 MPa) and AAPE (4.17, 5.42, and 6.29%) were observed for XGB models of stresses σ_y, σ_z, and σ_x.

Consequently, the proposed predictive models are quite reliable with robust prediction performances. Table 4 summarizes the evaluation metrics of the proposed models. A comparison of validation and training prediction errors of the σ_z, σ_x, and σ_y stress models are presented in Figure 17.

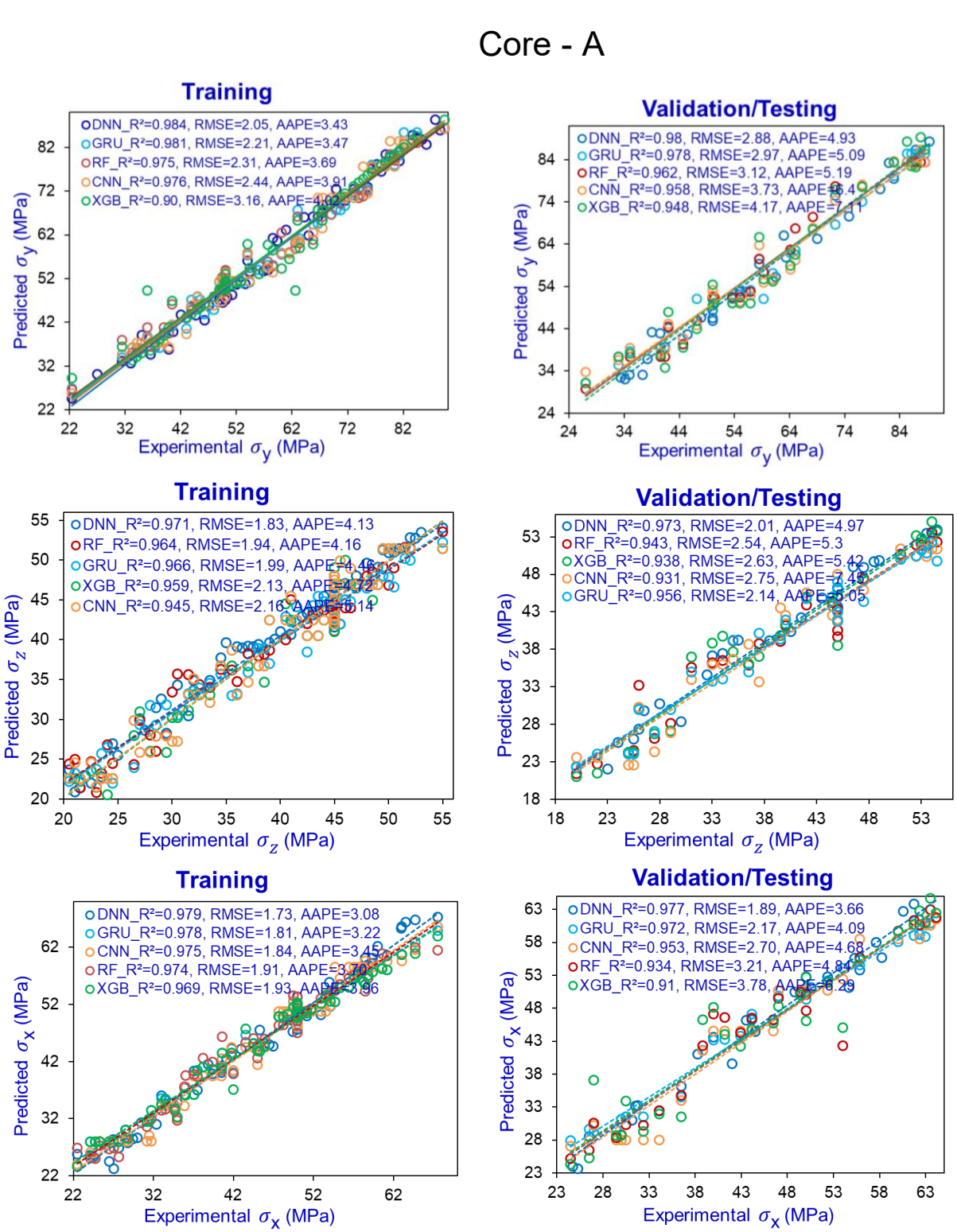


Figure 12: Cross plots between predicted and experimental stresses for the proposed DNN, GRU, CNN, RF, XGB models of σ_z, σ_x, and σ_y, stresses for Core-A.

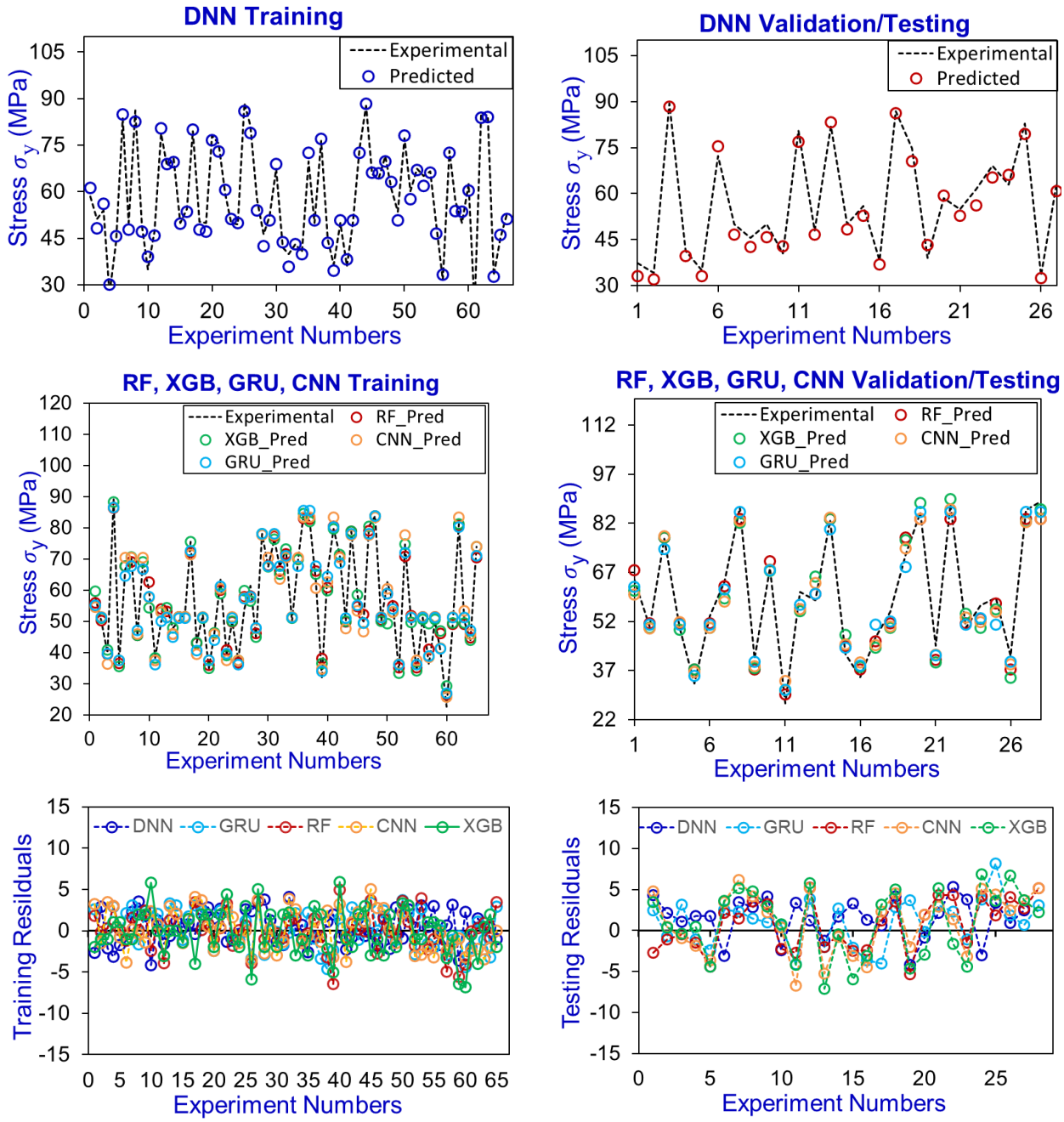


Figure 13: Comparison between predicted and experimental values of σ_y stress for the DNN, GRU, CNN, RF, XGB models developed for Core-A. Bottom two plots demonstrated the RE for the training and testing/validation phases.

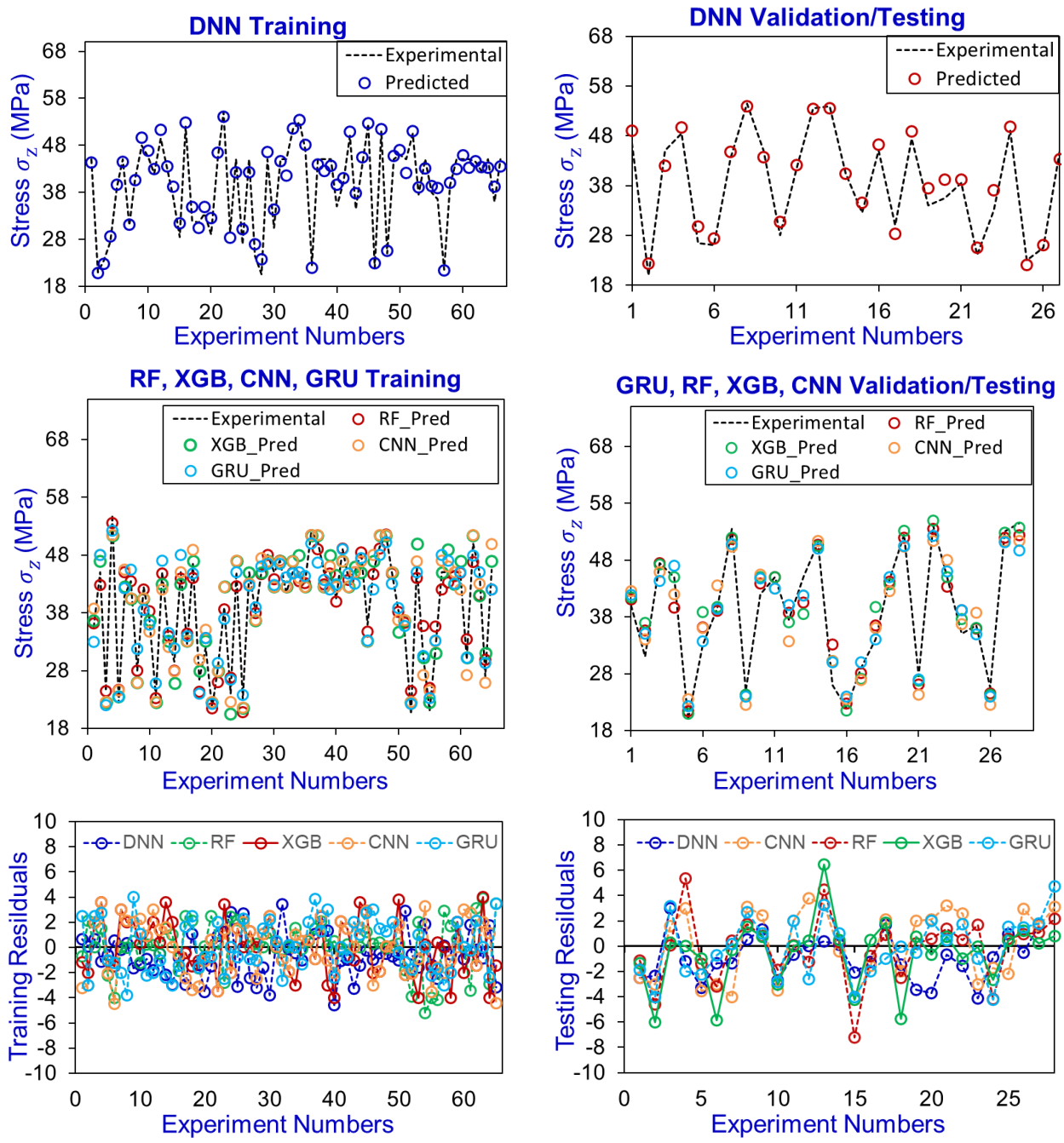


Figure 14: Comparison between predicted and experimental values of σ_z stress for the DNN, GRU, CNN, RF, XGB models developed for Core-A. Bottom two plots demonstrated the RE for training and validation/testing phases.

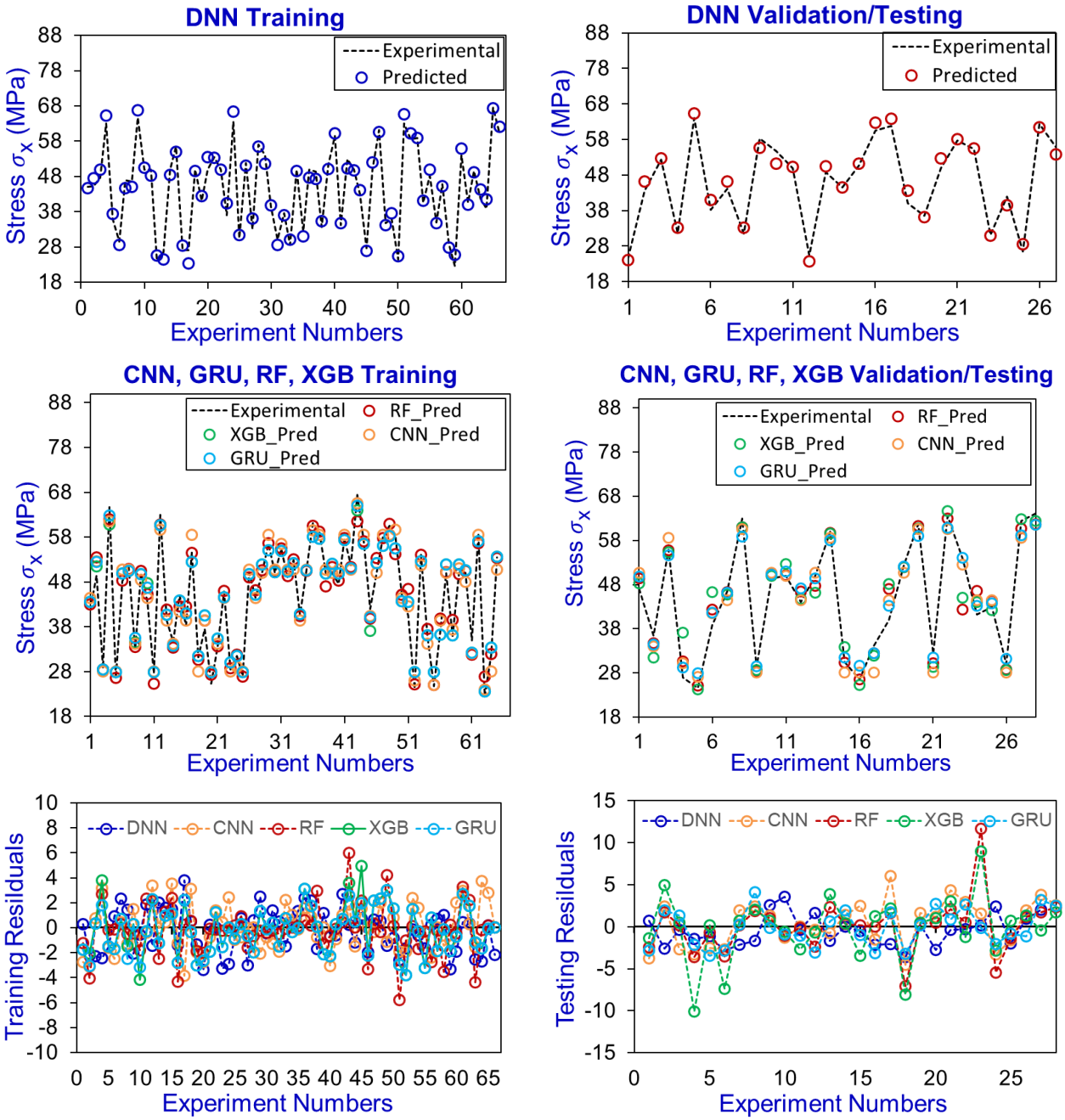


Figure 15: Comparison between predicted and experimental values of σ_x stress for the DNN, GRU, CNN, RF, XGB models developed for Core-A. Bottom two plots demonstrated the RE for training and testing/validation phases.

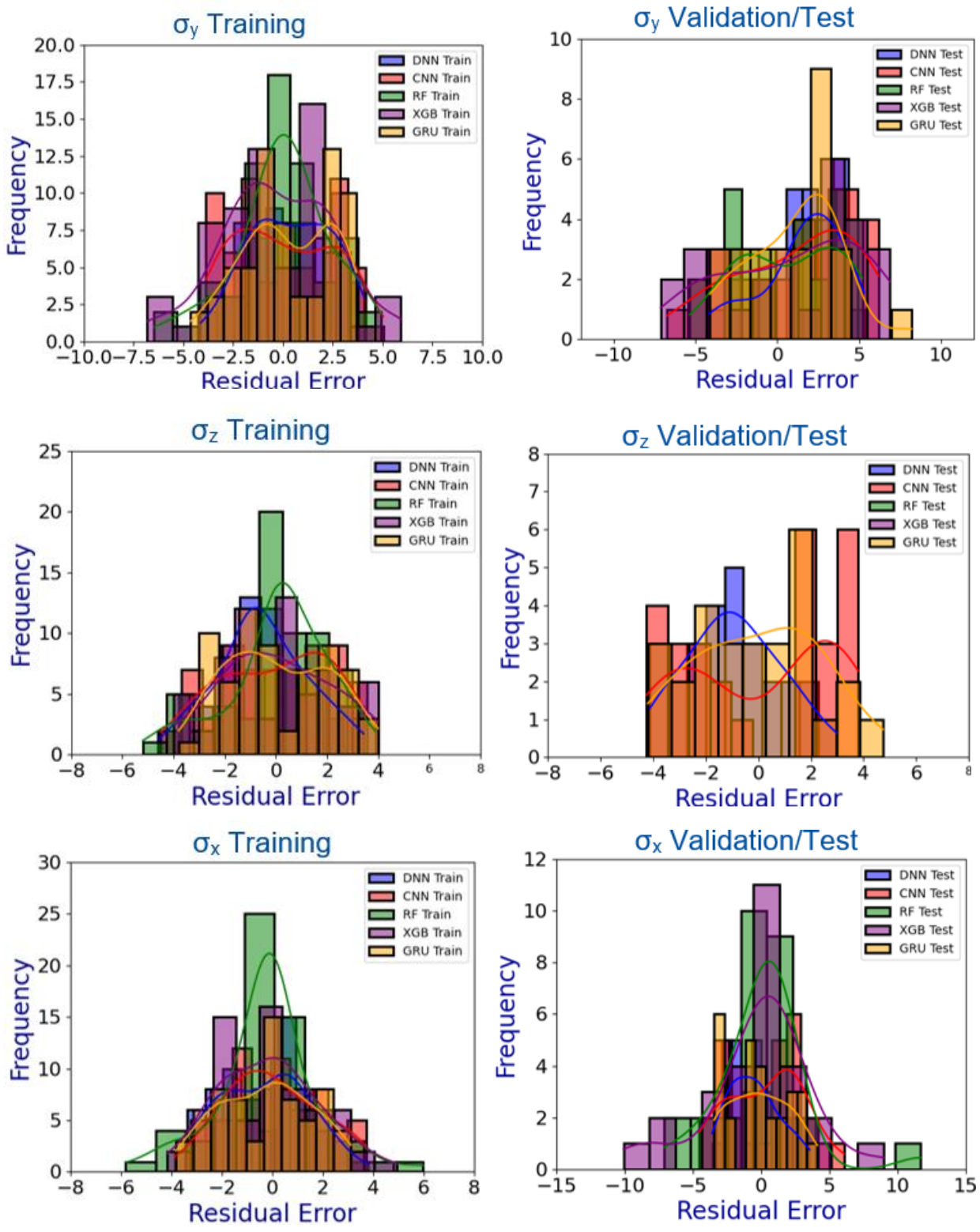


Figure 16: Residual errors of the stresses σ_z, σ_x, and σ_y, predicted by DNN, DT, KNN, RF, ADB, XGB models for training and testing/validation phases.

Table 4: Evaluation metrics for the training and testing/validation phases of the proposed DT, DNN, KNN, RF, ADB, XGB models for 'σ_z', 'σ_y', and 'σ_x' stresses.

Parameter	Evaluation Metrics	Data Category	ML Models				
			DNN	**GRU**	**CNN**	**RF**	**XGB**
σ_y	RMSE (MPa)	Training	2.05	2.21	2.44	2.31	3.16
		Test. & Val.	2.88	2.97	3.73	3.12	4.17
	AAPE (%)	Training	3.43	3.47	3.91	3.69	4.02
		Test. & Val.	4.93	5.09	6.40	5.19	7.11
	R^2	Training	0.984	0.981	0.976	0.975	0.901
		Test. & Val.	0.980	0.978	0.958	0.962	0.948
σ_z	RMSE (MPa)	Training	1.83	1.99	2.16	1.94	2.13
		Test. & Val.	2.01	2.14	2.75	2.54	2.63
	AAPE (%)	Training	4.13	4.46	5.14	4.16	4.72
		Test. & Val.	4.97	5.05	7.45	5.30	5.42
	R^2	Training	0.971	0.966	0.945	0.964	0.959
		Test. & Val.	0.973	0.956	0.931	0.943	0.938
σ_x	RMSE (MPa)	Training	1.73	1.81	1.84	1.91	1.93
		Test/Val.	1.89	2.17	2.70	3.21	3.78
	AAPE (%)	Training	3.08	3.22	3.45	3.70	3.96
		Test/Val.	3.66	4.09	4.68	4.84	6.29
	R^2	Training	0.979	0.978	0.975	0.974	0.969
		Test/Val.	0.977	0.972	0.953	0.934	0.910

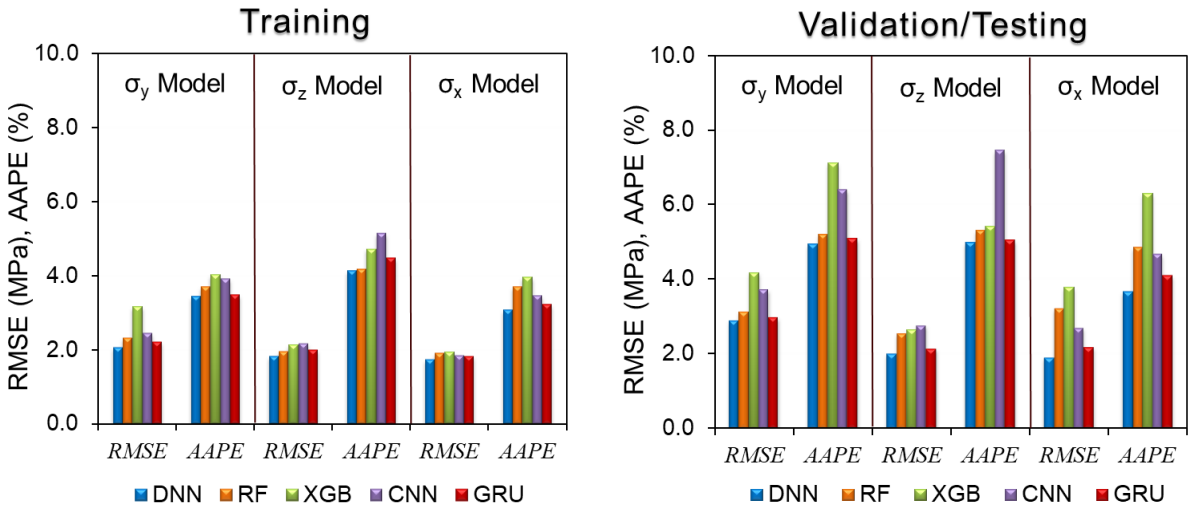


Figure 17: Comparison of prediction RMSE and AAPE for the proposed DNN, GRU, CNN, RF, XGB models of σ_y, σ_z, and σ_x stresses for Core-A.

9. Parametric/Sensitivity Analysis (Model Generalization)

Generalization capabilities of the proposed DNN predictive models were evaluated through the performance of parametric or sensitivity analysis. Parametric analysis illustrates the significance of input features for the target output variable. The parametric analysis was conducted on a synthetic dataset to evaluate the impact of each input feature on the target variable illustrating the underlying physics of the predictive models. Synthetic datasets are generated in a fashion that only one feature is kept changing while values of all other input features are constant. Thus, a distinct dataset is generated for each input feature and prediction was obtained using that generated dataset. In this work, parametric analysis is demonstrated for three models of σ_y, σ_z, and σ_x stresses developed. The influence of three input features i.e. V_{zz}, V_{zx}, and V_{zy} was examined for the target output σ_y. The impact of V_{zz} on σ_y was evaluated by changing the values of V_{zx} and keeping the values of V_{zx} and V_{zy} constant. Similarly, the impact of other input features V_{zx} and V_{zy} on σ_y was also evaluated by following the same strategy. Further, parametric analysis was performed for the other two predictive models of σ_x and σ_z as well using similar schemes. The parametric analysis demonstrated a specific pattern between each input feature and predicted output demonstrating their physical relationship. A unique pattern of curve was observed for each of the input features with output feature revealing good generalizability of the proposed DL/ML models of σ_y, σ_z and σ_x and σ_z stresses. The results obtained from the parametric analysis of σ_y, σ_z, and σ_x models are presented in Figures 18-20, respectively.

In this work, a unique pattern of curves represents the constitutive relation between each input feature and predicted output. Parametric analysis revealed that the relationship between the ultrasonic wave velocities (V_{zz}, V_{zx}, and V_{zy}) and stress (σ_y, σ_z, and σ_x) is not straightforward rather following a non-monotonic pattern which may not be the reflection of intuition. The non-monotonic patterns of constitutive relationships of velocities with vertical and horizontal principal stresses were previously observed and reported by Bunger et al. (2023), and Mustafa et al. 2024. It is inferred that vertical and horizontal stresses significantly influence the compressional and shear wave velocities (V_{zz}, V_{zx}, and V_{zy}) with rollover(s) after a certain level of stress developing a unique pattern of increasing and decreasing velocities.

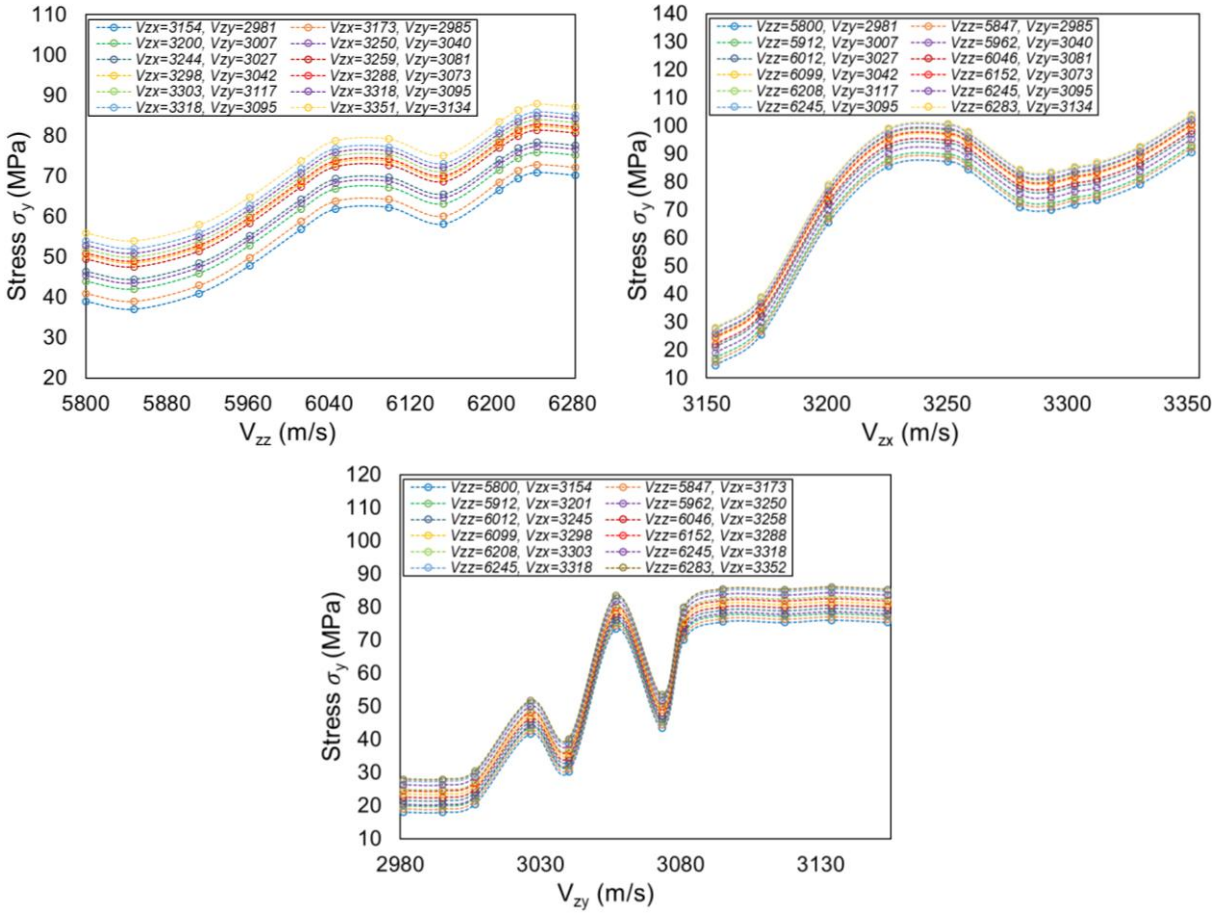
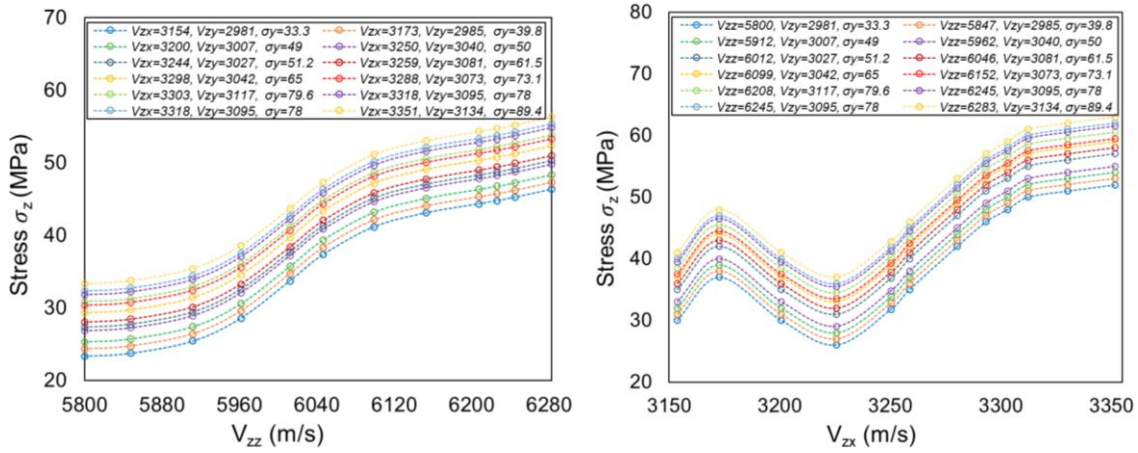


Figure 18: Parametric/Sensitivity study exhibiting impact of each input feature on predicted σ_y stress.



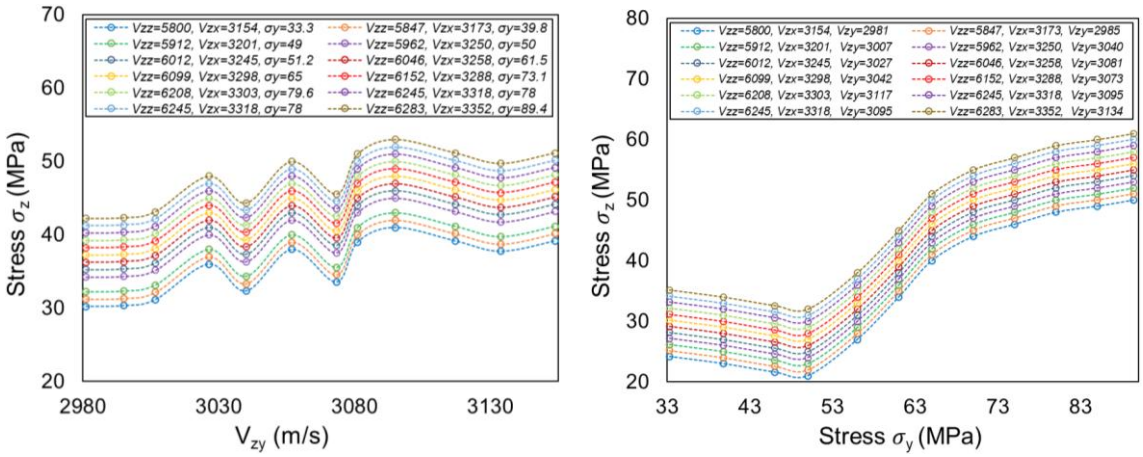


Figure 19: Parametric/Sensitivity study exhibiting impact of each input feature on predicted σ_z stress.

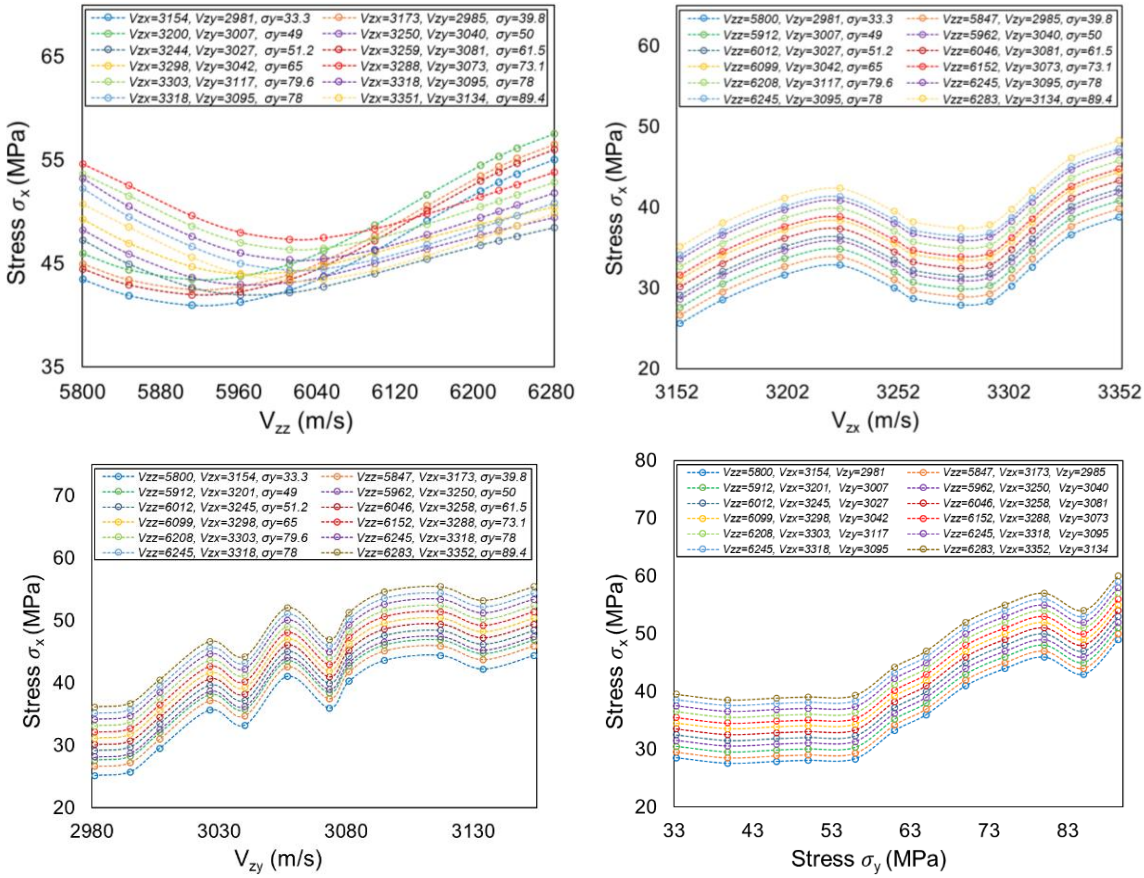


Figure 20: Parametric/Sensitivity study exhibiting impact of each input feature on predicted σ_x stress.

10.Dataset Size Effectiveness

To evaluate the effectiveness of TUV dataset sizes, learning curves of the proposed DNN, CNN, GRU, RF, and XGB models were analyzed for different sizes of TUV dataset. The purpose of evaluating the models' performances on different portions of datasets is to determine the minimum number of TUV experiments to be performed that are sufficient to develop reliable prediction models of three principal stresses. Additionally, ML/DL algorithms with the best and consistent predictive performance on all portions of datasets could be identified. Initially, the predictive performance of training and validation phases of the proposed models including DNN, CNN, GRU, RF, and XGB for three principal stresses (σ_y, σ_z, and σ_x) was evaluated for different sizes of dataset, ranging from a small portion (20%) to the full dataset (100%) with an interval of 20%. A comparison of RMSE and R^2 of training and validation phases of the

proposed predictive models confirmed the superior and consistent performance of the DNN models on all sizes of dataset sizes of three principal stresses. The comparison of predictive performances of the proposed models is presented in Figure 21.

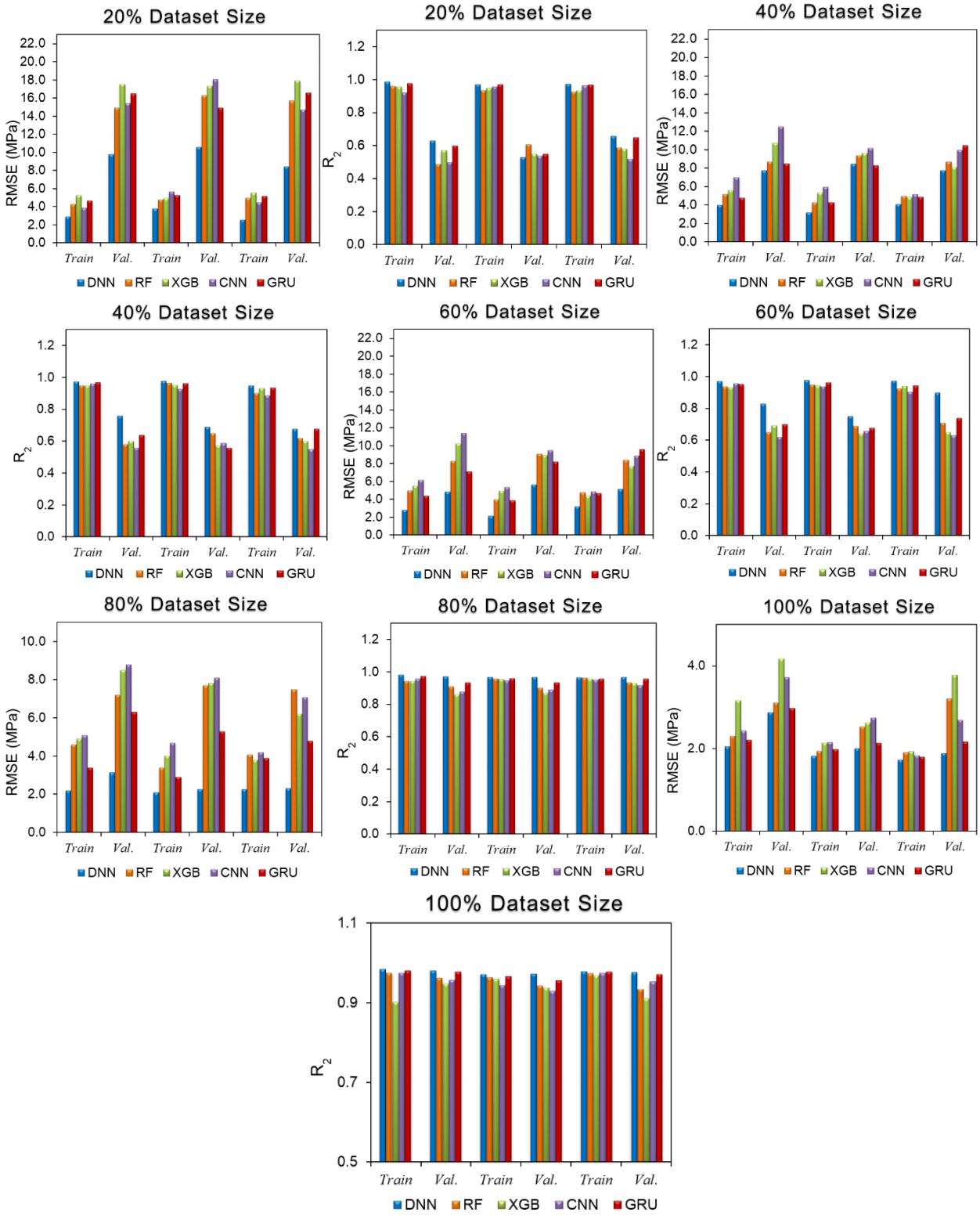


Figure 21: A comparison of RMSE and R^2 of the training and validation phases of DNN, CNN, GRU, RF, and XGB predictive models for σ_z, σ_x, and σ_y stress models.

For further analysis, DNN models are selected due to consistency, robustness and generalization capability. The analysis allows us to obtain the minimum size of TUV dataset that can be utilized to develop reliable prediction models of principal stresses in order to reduce the experimental time, labor and resources to ultimately improve the overall efficiency of the process. The learning curves were developed for all three principal stresses (σ_y, σ_z, and σ_x) by evaluating DNN models multiple times for even smaller sizes of TUV datasets, ranging from a small portion (20%) to the full dataset (100%) with an interval of 10%. Each of the dataset portions was split into training and testing subsets. For each selected portion, training and validation of DNN models were performed and corresponding evaluation metrics (R^2 and RMSE) were recorded. The iterative execution of models resulted in identifying the minimum TUV dataset size yielding no further significant improvement in R^2 and RMSE of predictive models with additional TUV data points. An optimal TUV dataset yields a balance between each model's prediction accuracy and experimental efficiency (time and labor, and other resources). Therefore, experimental resources, time and labor could be minimized without compromising the model accuracy.

The evolution of the model's performance with increasing size of TUV dataset is illustrated in Figure 22. For smaller TUV datasets, training performance (low RMSE and high R^2) is excellent, however validation performance is poor with high RMSE and low R^2 that indicates the overfitting of the model with poor generalization capability. With the increase in dataset size, training RMSE slightly increases and validation RMSE reduces significantly, leading to improved generalization. Out of the total data points, optimal performances (RMSE and R^2) for models' training and validation were achieved with 80% (75 data points) of the TUV dataset, confirming a balance between performance and generalization. Hence, reliable and generalized ML/DL models can be constructed for three principal stresses (σ_y, σ_z, and σ_x) using the dataset rendered from 75 TUV experiments per core sample, as long as velocities are measured precisely and the degree of nonlinearity in the relationships between the velocities and stresses are similar to the rocks considered in this study. Beyond this point, no significant improvements in model's accuracy was observed by further increasing the size of dataset, ensuring optimal size of TUV dataset is achieved. The corresponding RMSE and R^2 of the training and validation phases for different sizes of TUV dataset are demonstrated in Figure 23.

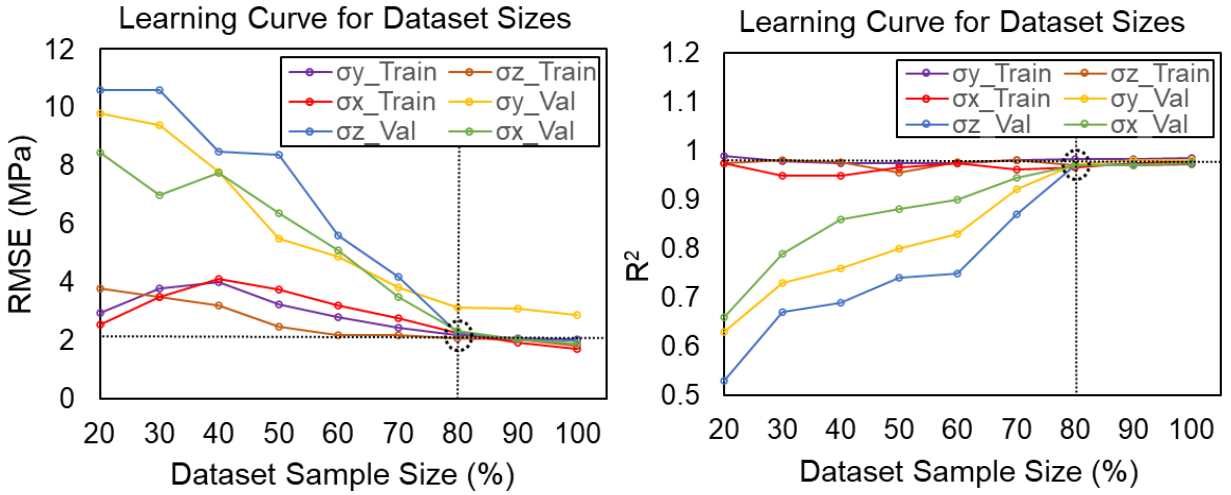


Figure 22: Learning curves of DNN models demonstrating the training and validation RMSE scores as a function of dataset size for σ_z, σ_x, and σ_y stress models.

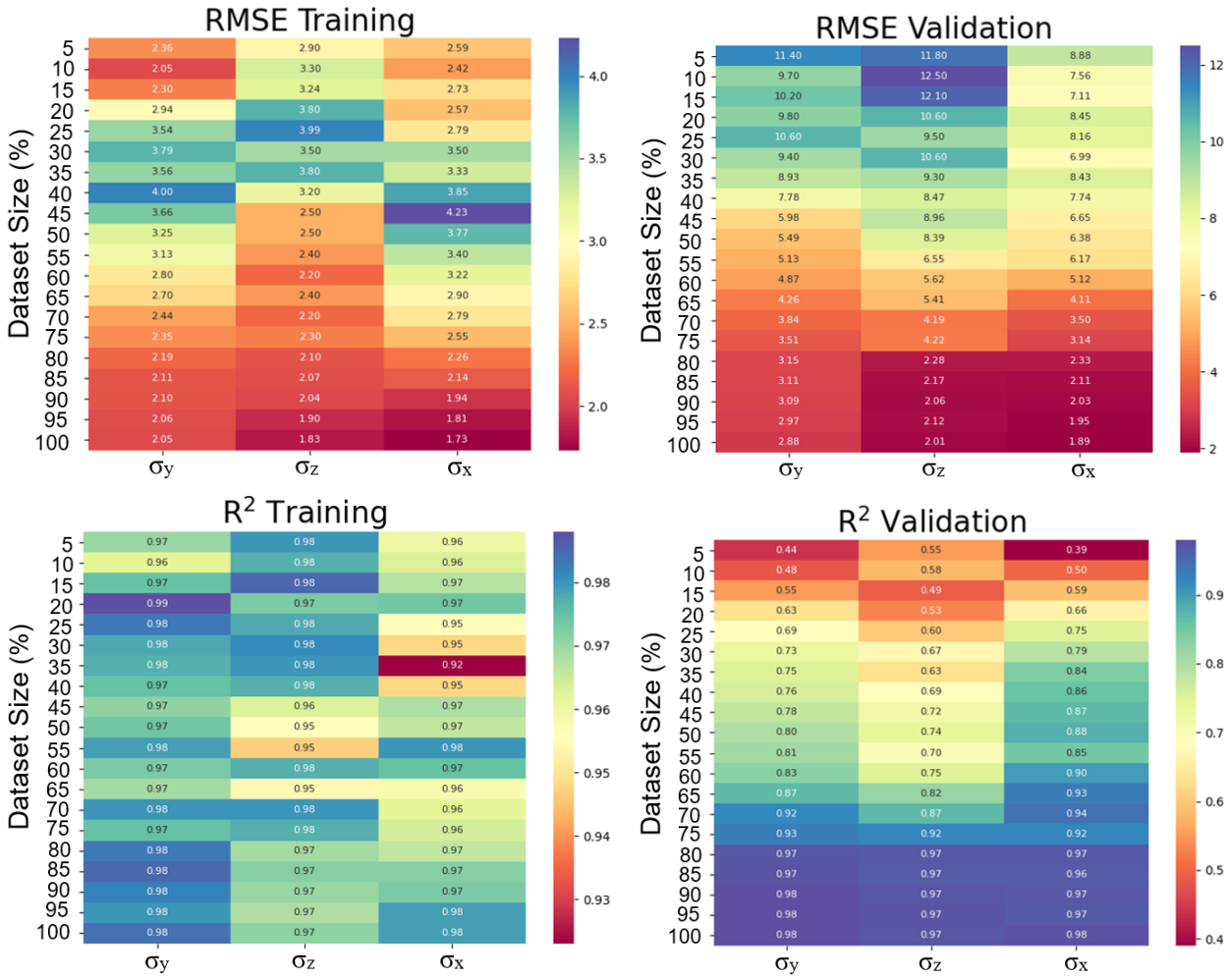


Figure 23: RMSE and R^2 scores of training and validation/testing of the DNN models as a function of dataset size for σ_z, σ_x, and σ_y stress models.

11. Interpretability of DNN Models - Global and Local Explanation

In this scientific application, the interpretability of the proposed DNN models is of high interest, which demonstrates the degree to which humans can comprehend the rationale behind the prediction and decisions of DNN models. Although DNN models demonstrated reliable and robust prediction outcomes, however black-box nature of these models results in a lack of interpretability and explainability. To overcome this limitation, SHapley Additive exPlanations (SHAP) analysis was applied for interpreting the global and local behavior of DNN models by explaining their inner workings to obtain specific predictions. In SHAP analysis, a specific Shapley value is assigned to each input feature for a specific target prediction, demonstrating an quantitative measure of average contribution of each input feature value towards the target prediction. Thus, in this work, different SHAP plots were employed to obtain insights about the behavior of individual input feature across different instances.

The SHAP summary plots provide a global perspective of SHAP values observing the contribution of each input feature to stress prediction compared to average prediction of models by integrating feature importance with feature impacts as illustrated in Figure 24A. Each point (dot) on the plot represents the Shapley value of a feature for a single instance in the dataset. Input features are ordered with respect to their importance, leading to the inference of the association between features' values and their influence on the model's predictions. For instance, for V_{zy} and V_{zx} features, higher values correspond to lower SHAP values, reflecting higher values of V_{zy} and V_{zx} lead to lower predicted stress values than the average model

prediction. Further, SHAP plots analysis can help infer the relationship between the features as well. The SHAP summary plots of σ_z and σ_x models are provided in Appendix-L. The average of absolute SHAP values across all instances (data points) are illustrated in the mean SHAP plot (Figure 24B), reflecting the impact of corresponding input features on model's prediction.

The SHAP waterfall plot provides the visual demonstration of SHAP values of input features illustrating the degree to which each input feature has impacted the model's prediction in comparison with mean prediction, thereby either decreasing or increasing the predicted stresses (σ_y, σ_z, and σ_x) for a specific instance in the dataset. The SHAP waterfall plot of σ_z model is presented in Figure 24C and plots of σ_z and σ_x models are provided in Appendix-L. The bottoms of the waterfall plots show the base values starting at $E[f(x)] = 54.84$ MPa for σ_y, 35.93 MPa for σ_z, and 44.5 MPa for σ_x, respectively, reflecting the average predicted stresses (σ_y, σ_z, and σ_x) across all the data points. Each value corresponding to specific features illustrates either negative (blue) or positive (red) contribution in shifting the expected output of the model (base value) to the predicted stress output of the model for that specific data point or instance. In these examples of the given specific instances, the model predicted stresses came out to be $f(x) = 50.575$ MPa, 38.475 MPa, and 50.107 MPa for σ_y, σ_z, and σ_x, respectively. The waterfall plots of specific instances (Table 5) show that the feature V_{zz} contributes negatively to the predicted stress σ_y and σ_z, thereby reducing the σ_y stresses by 11 MPa and σ_z stresses by 3.43 MPa, while contributing positively for stress σ_x, thereby boosting the σ_x by 7.27 MPa, respectively. The features V_{zx} and V_{zy} contribute positively to increasing the σ_y by 3.71 MPa and 3.02 MPa, respectively. The results obtained from SHAP waterfall plots outcomes are aligned with the findings of SHAP summary.

The SHAP force plot is utilized to visualize the SHAP values of each data point in the dataset, providing almost similar information as waterfall plot. The plot illustrates how the contribution of each feature leads to decrease or increase the model predicted stress values, resulting in the ultimate prediction of 50.575 MPa, 38.48 MPa, and 50.11 MPa, respectively. The SHAP force plot of σ_z model is presented in Figure 24D and plots of σ_z and σ_x models are provided in Appendix-L.

Another important plot namely SHAP decision plots are generated to obtain valuable insights into how the stress predictions are generated from DNN model of stress σ_y as shown in Figure 25A and 11B. The model's base value is illustrated by the line positioned at the bottom of the plot. The path of each line demonstrates the influence of SHAP value of each input feature on prediction, resulting ultimate predicted stresses of models at the end of line (on top of the plot). Each line corresponds to one data instance (observation). It is vital to note that results obtained from SHAP decision plots for a single instance (data point) are in agreement with the results obtained from SHAP summary, waterfall and force plots. The zig-zag pattern of SHAP decision plot (Figure 25B) indicates that feature V_{zz} have negative impact (negative SHAP values) whereas features V_{zy} and V_{zx} contribute positively to σ_y model prediction for the given instance. The decision plots of σ_z and σ_x models are provided in Appendix-L.

To comprehend the impact of a specific feature on model's prediction along with interaction impact of other features, SHAP dependence plots are generated and analysed. The impact of changes in input features (V_{zz}, V_{zx} and V_{zy}) and SHAP values on model predicted stresses σ_y is illustrated in Figure 25C. The dependence plots of σ_z and σ_x models are provided in Appendix-L. It is quite evident from dependence plot that the larger values of features V_{zz} tend to have positive SHAP values thus, lead to the higher predicted σ_y stress and vice versa. On the contrary, lower values of features V_{zx} and V_{zy} with positive SHAP values causing the predicted σ_y stress to increase and vice versa. While the interacting feature exhibited alternately increasing and decreasing trends specifically in the vicinity of the central values of feature (Figures 25C).

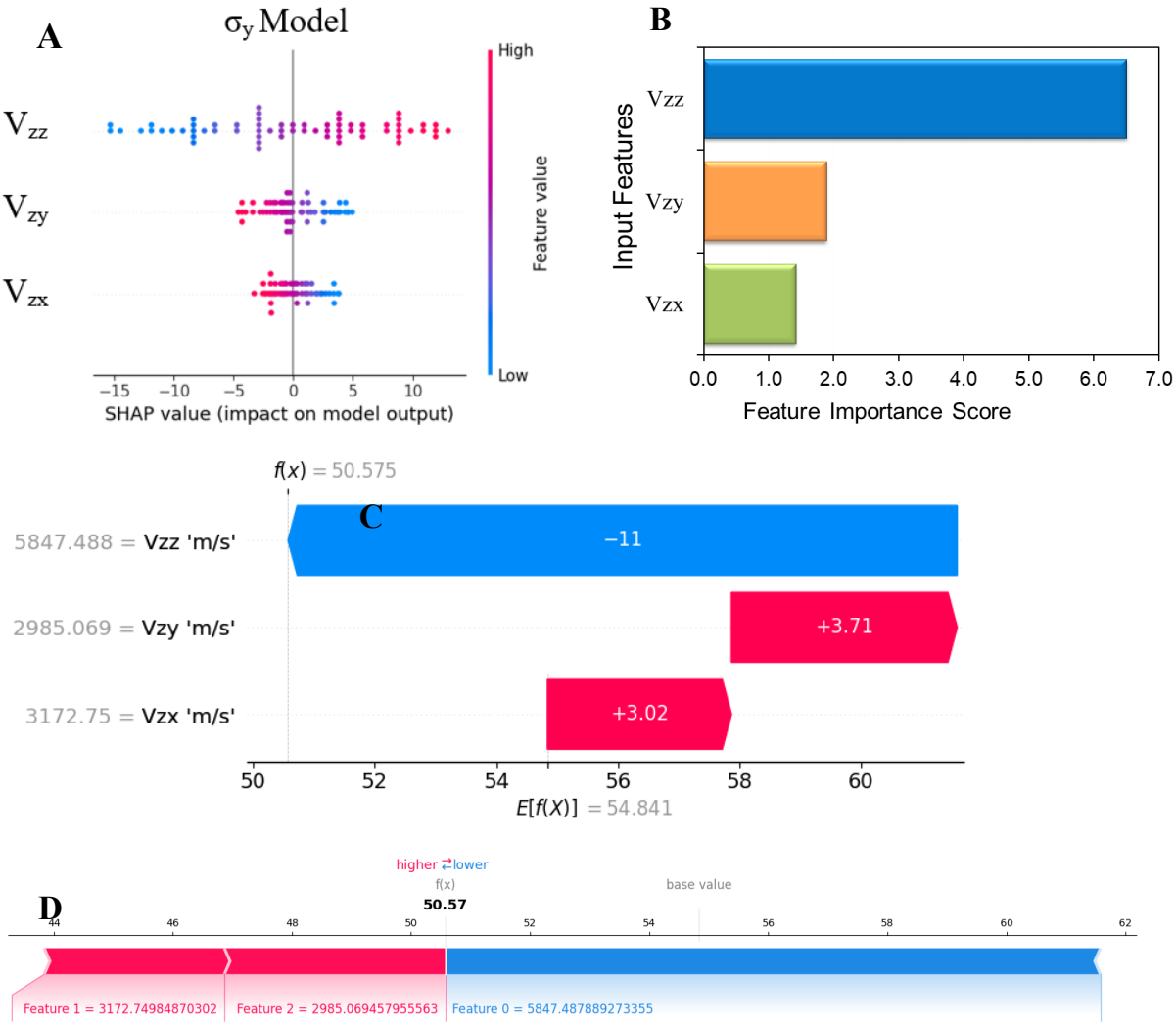


Figure 24: SHAP analysis of σ_y model; (A) SHAP summary plots, (B) Corresponding feature importance scores, (C) SHAP waterfall plot for a specific instance representing impact of individual feature on prediction of stress; (D) SHAP force plots representing individual feature contribution to get the final prediction of stress.

Table 5: Specific data points in the dataset of σ_y, σ_z, and σ_x models.

Models	V_{zz}	V_{zx}	V_{zy}	σ_y
σ_y Model	5847.5	3172.75	2885.07	----
σ_z Model	6207.74	3117.13	3303.13	79.625
σ_x Model	6207.74	3117.13	3303.13	79.625

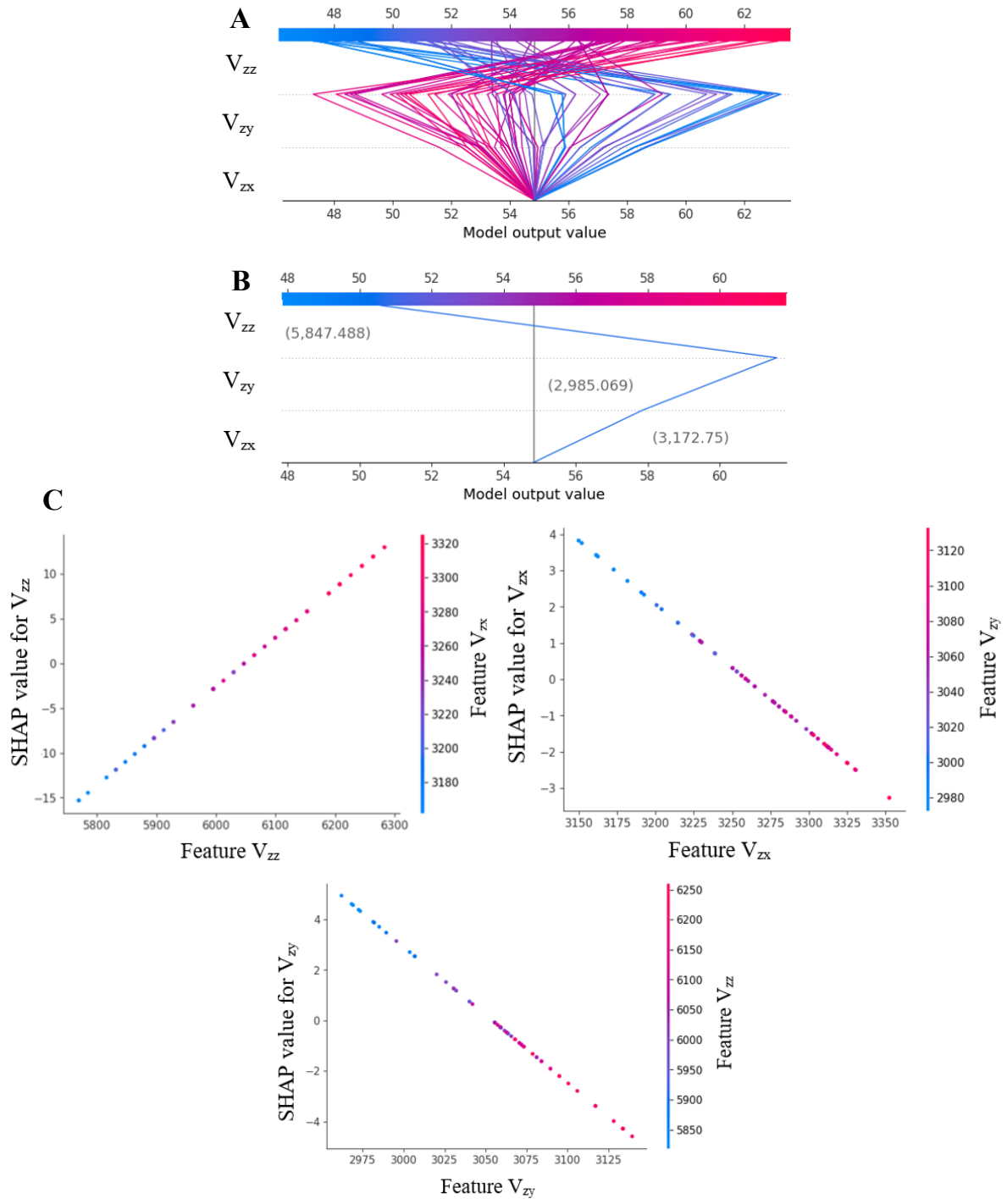


Figure 25: SHAP decision and dependence plots for σ_y model; (A) decision plot for few instances, (B) decision plot for a specific instance, (C) dependence plots for features V_{zz}, V_{zx}, and V_{zy}.

12. Petrofacies Analysis

At this stage, an unsupervised K-means algorithm was implemented to classify the subsurface rock formations into a number of clusters. Each cluster/group is the representative of a distinct petrofacies (PF) that encapsulates a set of petrophysical and formations characteristics. The clustering analysis was performed using the petrophysical well log data such as bulk density (ρ), neutron porosity (NPHI), photoelectric factor (PEF), and gamma ray (GR), obtained from the measured depth interval from 4835 - 10872 ft of the same well 16B(78)-32. The objective of subsurface rock classification is to recognize the representative PF which are located at the same depth of subsurface core samples used in this work.

The K-means algorithm was iteratively executed for different plurality of clusters in order to obtain the optimum numbers of clusters/petrofacies. For the execution of each run, sum of the squared distances of all the data points to the nearest cluster's centroid is computed, also termed as 'inertia'. Based on inertia values, elbow plot was generated and evaluated in order to determine the optimum number of clusters/petrofacies as shown in Figure 26.

Optimally, a total of six (06) PF were identified for the selected rock characteristics and depth interval indicated heterogeneity in the properties of subsurface geological formations. The identified PF along with the cross plots between the selected rock characteristics are demonstrated in Figure 26. It is worth to mention that a total of five core samples depths are represented by three PF .e. PF-1, PF-5, and PF-6. The PF-1 (sky blue) is the representative PF for core-A, PF-5 (blue) is representative of core-D and -E, whereas PF-6 (purple) is the representation of core-B and C. The representative PFs demonstrate petrophysical and formation characteristics that is similar to the subsurface core samples, thus lead to similar constitutive behavior. The representative PF were also marked/located above and below the sampling locations along the well. The identified PFs for the entire section of well along with well log suite are illustrated in Figure 27.

Further, good performance of clustering models was reflected by the SI score of 0.70 for the identified PFs indicating compactness and fitness of data points to specified cluster (PF). The data points of PF-1, 4, 5, and 6 are observed to be compacted leading to distinct PFs. It is important to note that the primary focus of this work is on PF-1, 5, and 6 which are the most relevant PFs and representative of core samples' characteristics. on the other hand, data points of PF-2, and 3 seems to be less compacted leading to less distinctive clusters. One of the prominent reason for this is the geological complexity with transitional or mixed facies. The gradual transition in the geological characteristics might lead to less compacted clusters.

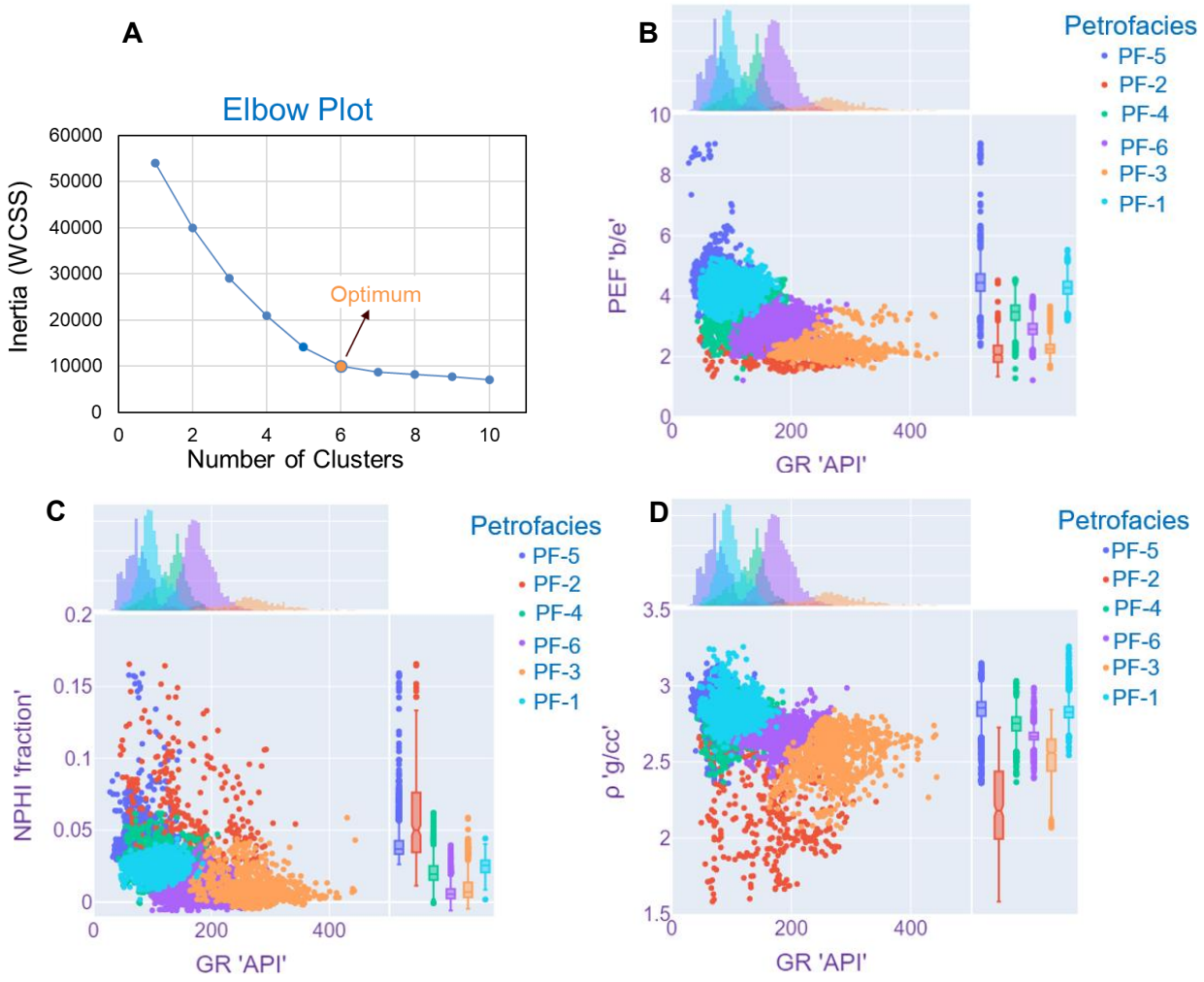


Figure 26: K-means clusters results; (A) Elbow plot, (B) GR vs NPHI, (C) GR vs ρ, (D) NPHI vs ρ for the depth 5000-6000 ft of well 16B(78)-32.

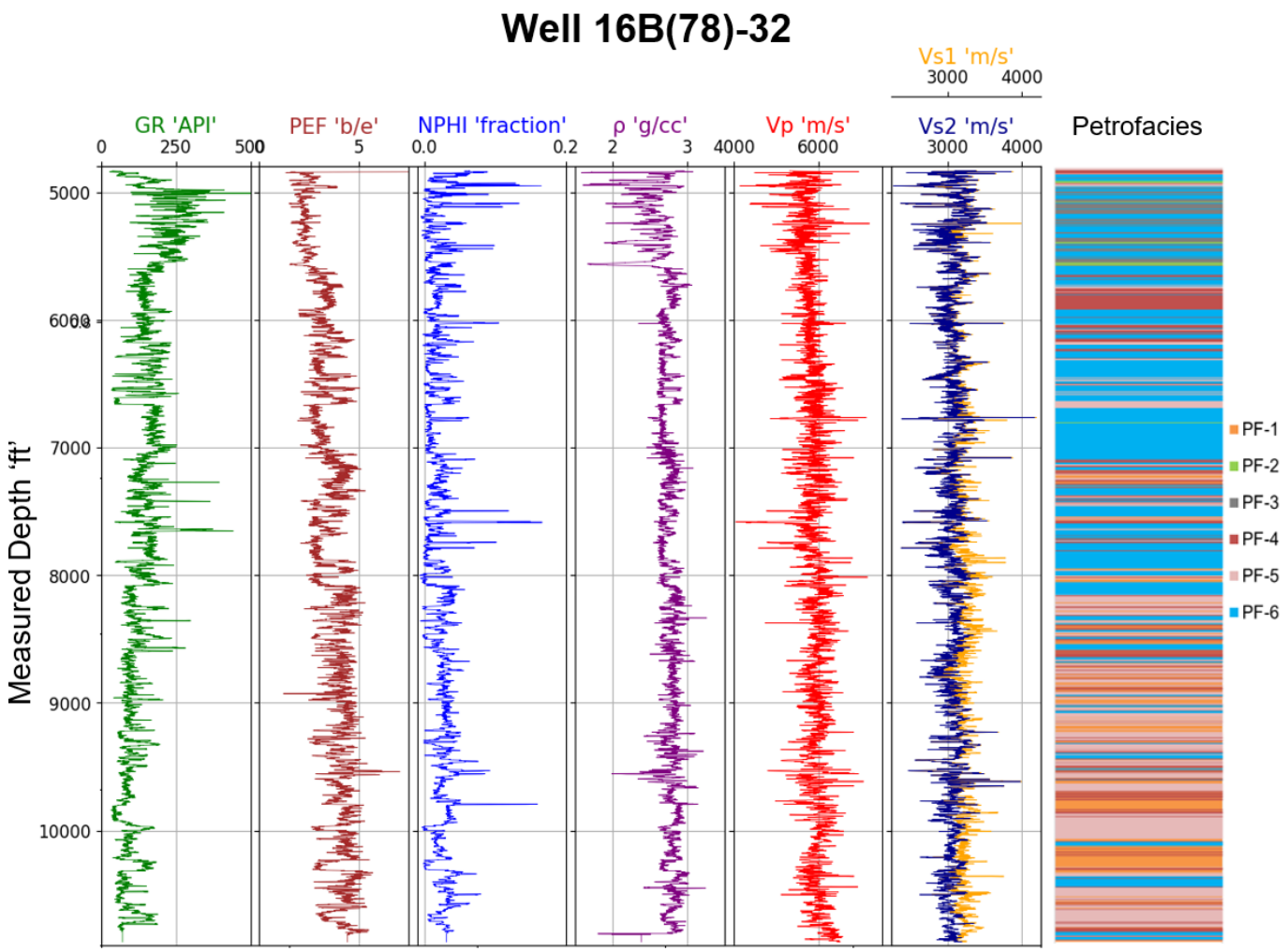


Figure 27: The well log suite and petrophysical clusters for the well 16B(78)-32

13. In-situ Stress in Well 16B(78)-32 using Trained DL/ML Models

At this stage, the DNN were applied for estimating near-field in-situ stresses in the subsurface rock formations using to field sonic log data acquired from the measured depth interval ranging from 4835 to 10872 ft of well 16B(78)-32. It is important to mention that the respective DNN models of three principal stresses i.e. σ_y (vertical), σ_x (maximum horizontal) and σ_z (minimum horizontal) stresses were employed to predict corresponding in-situ principal stresses, namely vertical (S_v), maximum horizontal (Sh_{max}), and minimum horizontal (Sh_{min}) in the subsurface rock formations, respectively.

The DNN models for principal stresses were applied corresponding to specific stress orientations relative to the well axis: the normal stresses parallel to well axis (σ_y') and the two mutually orthogonal principal stresses within the plane perpendicular to the well (σ_x' and σ_y'). The input features used for these respective stress predictions, namely V_p, $V_{s\text{-}slow}$, $V_{s\text{-}fast}$, and bulk density gradient, were acquired from field logs along variously oriented sections of well 16B(78)-32. The entire well 16B(78)-32 is comprised of a vertical and a deviated section as illustrated in Figure 29E. For the vertical section, predicted near-field stresses (Sh_v, Sh_{min}, and Sh_{max}) are aligned with the in-situ principal stresses. However, predicted stresses in the deviated section (inclined at 65˚ from vertical) need to be aligned with principal stress orientation through three-dimensional (3-D) stress transformation. The normal and shear stress components in a global coordinate system, aligned with three in-situ principal stresses (vertical, minimum horizontal and maximum horizontal), can be computed from the local (deviated wellbore) coordinate system (σ_x', σ_y', and σ_z') by

computing the rotation matrix (Q_{ij}) for stress tensor which is simply the direction cosines (l_i, m_i, and n_i where i=1,2,3) between the major axes in the global and local coordinate and can be expressed as

$$\begin{Bmatrix} \sigma_z' \\ \sigma_x' \\ \sigma_y' \end{Bmatrix} = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 \\ l_2^2 & m_2^2 & n_2^2 \\ l_3^2 & m_3^2 & n_3^2 \end{bmatrix} \begin{Bmatrix} Sh_{min} \\ Sh_{max} \\ S_v \end{Bmatrix} = \begin{bmatrix} \cos^2 25^\circ & \cos^2 90^\circ & \cos^2 65^\circ \\ \cos^2 90^\circ & \cos^2 0^\circ & \cos^2 90^\circ \\ \cos^2 65^\circ & \cos^2 90^\circ & \cos^2 155^\circ \end{bmatrix} \begin{Bmatrix} Sh_{min} \\ Sh_{max} \\ S_v \end{Bmatrix} \quad \text{(Eq. 9)}$$

A comparison was made between the field-based elastic geomechanical model (FB-EGM) and DNN prediction of in-situ stresses. It is important to note that field logs including density and sonic logs were measured in the near wellbore region where thermo-poro-elastic stress disturbance (Tao and Ghassemi, 2010; Lu et al., 2024) and near-wellbore stress concentration (Kirsh, 1998) may be expected. Therefore, DNN predicted in-situ principal stresses may be influenced by near-wellbore alterations and therefore require translation to far-field stresses through coupled thermo-poro-mechanical simulations (Lu et al, 2024). Nevertheless, DNN predicted in-situ stresses are in good agreement with FB-EGM horizontal stresses and bulk-density-based vertical stress specifically for the representative PFs (PF-1, 5, and 6). Thus, the performance of DNN models was observed to be optimum only for the representative PFs (PF-1, 5, and 6) as they are illustrative of constitutive behavior of subsurface core samples. The zones/layers of representative PF were also recognized at several locations along the wellbore. In contrast, the performances of DNN models for non-representative PFs (PF-2, 3, and 4) were examined not to be as good as for representative PFs due to attributes different from the subsurface core locations/depths. Thus, DNN predicted stresses were observed to be deviated from the FB-EGM stresses estimation for the non-representative PF.

A cross plot between the DNN predicted and FB-EGM based Sh_{min} and S_v stresses demonstrated excellent prediction performance of DNN models with R^2 score of 0.813 and 0.865 for the representative PF (PF-1, 5, and 6) and 0.024 and 0.288 for the non-representative PF (PF-2, 3, and 4) respectively, as illustrated in Figure 28. The comparison between DNN predicted and FB-EGM stresses for the representative and non-representative PF are illustrated in two separate plots as shown in Figures 29. The DNN predicted in-situ stresses at the core depths of well 16B are shown in Table 6. Hence, this study analysis revealed that the developed DL/ML models were able to efficiently capture the constitutive relationship between acoustic velocities and three principal in-situ stresses in the subsurface core samples and successfully predicted the in-situ stresses in the subsurface core locations and zones of representative PFs. Further, the results revealed that the constitutive behavior of geological formations is strongly influenced by their petrophysical and formation characteristics, thus might lead to impact the distribution of in-situ stress and variation in sonic velocities. Hence, this work demonstrated the stress-velocity relationship that leads to the inference that variation in compression and shear wave velocities is strongly impacted by the in-situ stress variations in the subsurface rock formations.

Although, this approach offers an effective method to predict sin-situ stresses by capturing and implying stress-velocity relations but with certain limitations. Firstly, the stress-velocity relationship explored by the DL/ML models are applicable only to the representative PFs that contains geological and petrophysical characteristics similar to core sample locations. Therefore, to extend stress interpretation to other PF encountered within the well log interval, subsurface cores need to be acquired from at least one of the zones/layers of each PF. Additionally, accurate estimation of in-situ stress can be possible only if the oriented subsurface cores are available. Further, DNN models were generated on TUV data performed on intact or unfractured core samples, therefore, predicted in-situ stresses may have uncertainty in highly fractured or fault shear zones due to the localized stress changes.

In spite of certain limitations, this approach can provide reliable estimation of in-situ stresses in subsurface rock formations if applied under favorable environment/conditions. Consequently, this study leads to the cost-effective, time-saving solution of estimating in-situ stresses without the performance of costly field injection tests such as mini-frac or micro-frac tests in the wellbore.

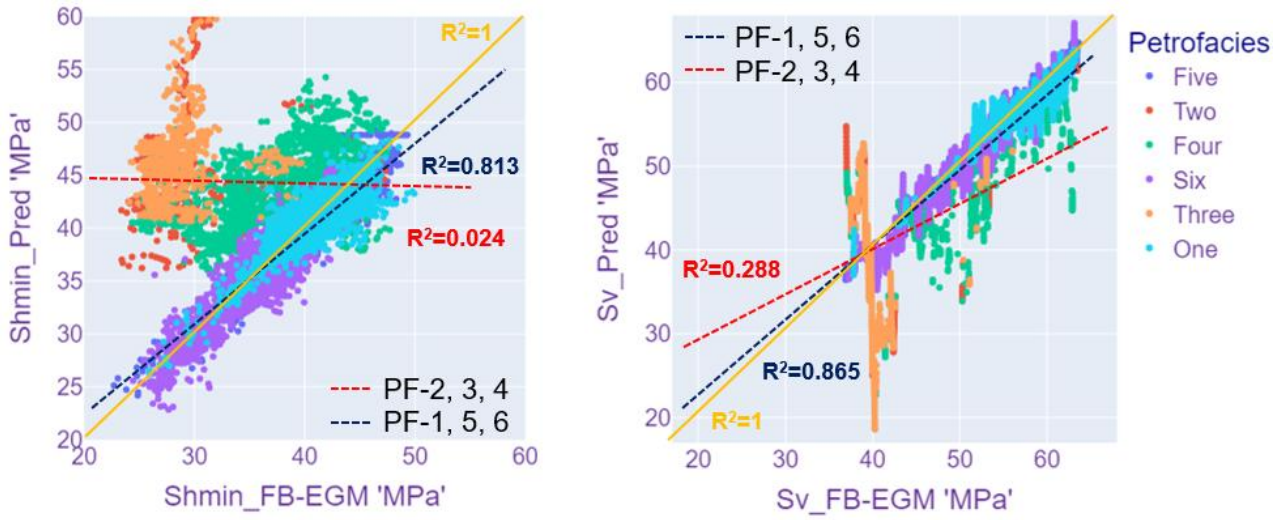


Figure 28: Cross plot between the DNN prediction and FB-EGM based Sh_{min} (left) and S_v (right) for the representative and non-representative PF in well 16B(78)-32.

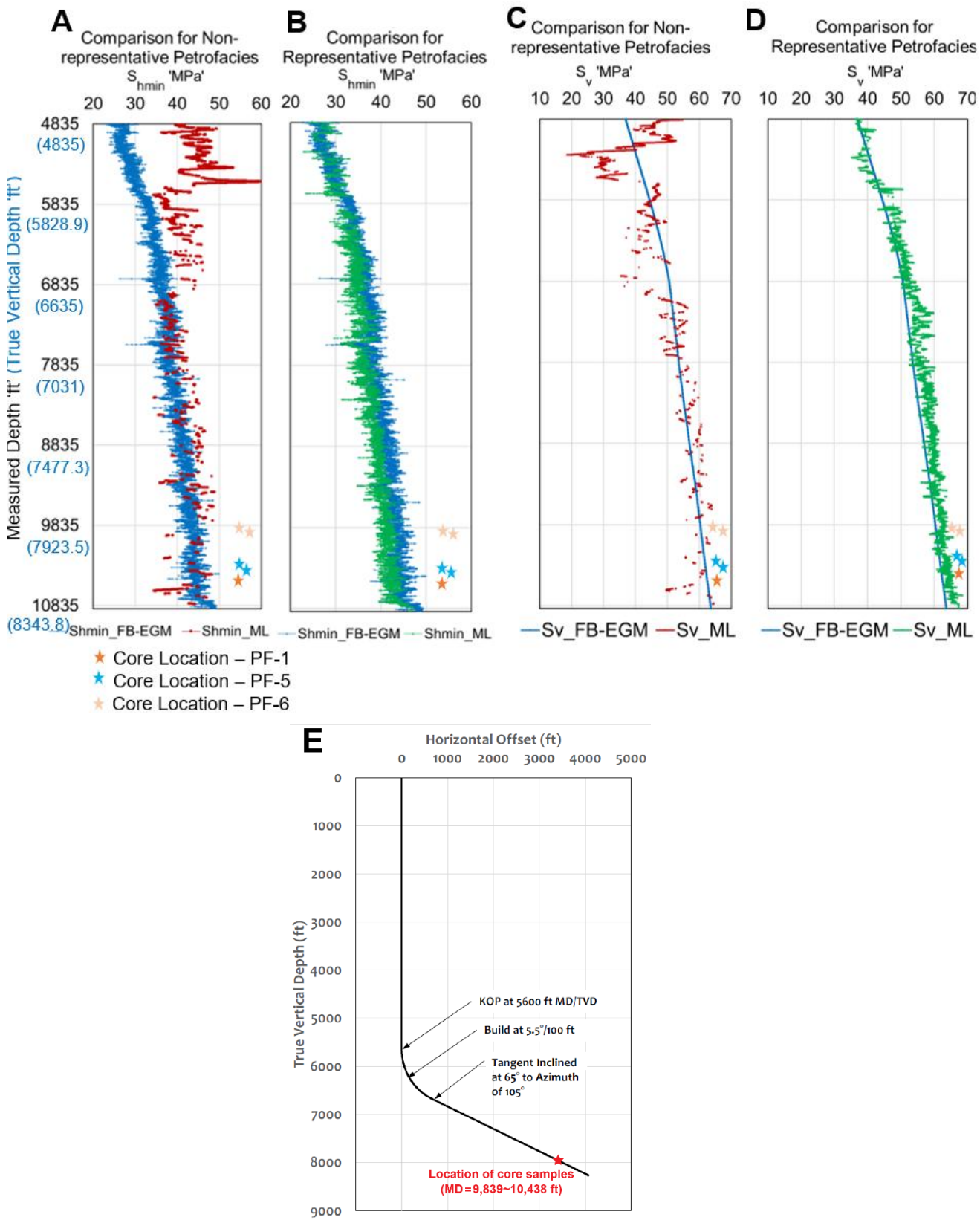


Figure 29: A comparison between in-situ stresses in well 16B(78)-32 obtained from ML prediction and FB-EGM; (A) shows a comparison of Sh_{min} for non-representative PF, (B) shows a comparison of Sh_{min} for the representative PF, (C) shows a comparison of S_v for non-representative PF, (D) shows a comparison of S_v for representative PF. Core sampling locations/depths of different PF is shown by stars. (E) Well trajectory shown in vertical plane (modified from Lu et al., (2023)).

Table 6: DNN predicted in-situ stresses at core locations of well 16B(78)-32.

Core ID	Measured Depth 'ft'	True Vertical Depth 'ft'	Sh_{min} 'MPa'	Sh_{min} Gradient 'psi/ft'	Sv 'MPa'	Sv Gradient 'psi/ft'	Shmax 'MPa'	Shmax Gradient 'psi/ft'
D	9,839	7,925.10	38.100	0.697	64.949	1.189	49.577	0.907
E	9,842	7,926.30	39.139	0.716	65.013	1.190	49.642	0.908
B	10,253	8,090.10	39.802	0.714	66.143	1.186	48.938	0.877
C	10,264	8,094.79	39.754	0.712	65.335	1.171	49.210	0.882
A	10,438	8,170.35	38.412	0.682	66.136	1.174	49.515	0.879

14.Conclusions

The project Milestone covered the workflow of DL/ML models development using TUV dataset and implementation of DL/ML models for predicting in-situ stresses in well 16B(78)-32 at the Utah FORGE geothermal site. To complete the milestone, extensive TUV experiments were conducted to generate TUV dataset in order to train and validate the DL/ML predictive models for three principal stresses i.e. σ_x, σ_y, and σ_z for each of the five subsurface cores. The milestone presents the analysis of DL/ML models performance for different sizes of TUV datasets in order to determine the minimum number of TUV experiments required to generate reliable and robust predictive models. The milestone presents the optimized DL/ML predictive models with excellent generalization capabilities capturing the underlying physics of the models as reflected by parametric analysis. The major outcome of the entire workflow is that stress-velocity relationships-based DL/ML models are capable to predict the in-situ vertical stress using sonic logs (V_p, V_{slow}, and V_{fast}) as input data and two principal horizontal stresses using sonic logs and density-based vertical stresses in the subsurface rock formations in well 16B(78)-32. The evaluation of learning curve revealed that 75 TUV experiments will be sufficient for generating reliable and generalized ML/DL models. Adding more data points does not significantly improve the models performance and prediction accuracy. Further, SHAP analysis provided how the input features work and interact to obtain the final prediction of models making the complex ML/DL models a white box approach. Intuitive explanation of models working enhanced the trust and scientific validation of the ML/DL models.

The conclusion is drawn based on high accuracy and reliability of the predicted in-situ stresses in the field demonstrating robustness of DL/ML models in terms of low prediction errors, AAPE and RMSE, and high R^2. Further, the generalization capabilities of the models captured the velocity-stress constitutive relationship that actually resulted in successful implementation of DL/ML models for the in-situ stress prediction, especially in the zones/layers of representative PF. The promising results of this work are quite evident from a close match between the DL/ML predicted and FB-EGM-based stresses for the zones/layers of representative PF. Representative PF are the illustrative of the constitutive behavior of the subsurface core sampling locations/depths. Exceptions are possible for the non-representative PF as their petrophysical and formation attributes are different from the subsurface cores, thus, demonstrate different velocity-stress constitutive behavior. Nonetheless, DL/ML models based prediction of in-situ stresses in subsurface rock formations are of acceptable quality and provide a promising path forward for providing the economical, quick and robust solution for estimating in-situ stresses.

15.Acknowledgement

This work was performed at the University of Pittsburgh with funding provided by DOE EERE Geothermal Technologies Office to Utah FORGE and the University of Utah under Project DE-EE0007080 Enhanced Geothermal System Concept Testing and Development at the Milford City, Utah Frontier Observatory for Research in Geothermal Energy (Utah FORGE) site.

16. References

Ahmad, A., & Khan, S. S. (2019). Survey of state-of-the-art mixed data clustering algorithms, IEEE Access 7, 31883–31902.

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/

Belyadi, H., & Haghighat, A. (2021). Machine learning guide for oil and gas using Python, Vol. 10. Elsevier.

Bunger A. P., 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

Bunger, A. P., Higgins, J., Huang, Y., Hartz, O., & Kelley, M. (2024). Integration of Triaxial Ultrasonic Velocity and Deformation Rate Analysis for Core-Based Estimation of Stresses at the Utah FORGE Geothermal Site. Geothermics, 120, 103008.

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.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I. & Zhou, T. (2015). Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4), 1-4.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Drineas, P., Frieze, A. M., Kannan, R., Vempala, S. S., & Vinay, V. (1999). Clustering in large graphs and matrices, Proceedings, ACM-SIAM Symposium on Discrete Algorithms, (SODA), 17-19 Jan 1999, 291–299.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. The Annals of Statistics 28(2), 337-407.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Guo, Z., Moosavi, V., & Leitao, J. P. (2022). Data-driven rapid flood prediction mapping with catchment generalizability. Journal of Hydrology, 609, 127726.

Hosseini, S. A., Weissend, M. A., Maier, A., & Christlein, V. (2021). Convolutional neural networks for regression: applications in magnetic resonance imaging. Applied Intelligence, 51, 8411–8429.

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

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters 31(8), 651–666.

Jancey, R. C. (1966). Multidimensional group analysis, Australian Journal of Botany 14, 127–130.

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

Kirsch, E. G. (1898). Die Theorie der Elastizitat und die Bedurfnisse der Festigkeitslehre. Zeitshrift des Vereines deutscher Ingenieure 42, 797-807.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: A big comparison for NAS. arXiv preprint arXiv:1912.06059

Lu, G., Lu, Y., Kelley, M., Raziperchikolaee, S., & Bunger, A. (2023). Interpretation of Mini-frac Test Data Accounting for Wellbore Cooldown in an EGS Well at the Utah FORGE Site: A Numerical Study. In 2023 Geothermal Rising Conference, Reno, 1-5 October 2023. GRC Transactions, Vol. 47, 2023.

Lu, G., Kelley, M., Raziperchikolaee, S., & Bunger, A. P. (2024). Modeling the impact of thermal stresses induced by wellbore cooldown on the breakdown pressure and geometry of a hydraulic fracture from an EGS well. Rock Mechanics and Rock Engineering.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations, in: Fifth Berkeley Symposium on Mathematics. Statistics and Probability, University of California Press, Berkeley,

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

Microsoft. (2015). Visual Studio Code. [Computer software]. Retrieved from https://code.visualstudio.com/

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.

Mustafa, A., Kelley, M., Lu, G., & Bunger, A. P. (2024). An Integrated Machine Learning Workflow to Estimate In-Situ Stresses Based on Downhole Sonic Logs and Laboratory Triaxial Ultrasonic Velocity Data. Journal of Geophysical Research: Machine Learning and Computation, 1(4), e2024JH000318. https://doi.org/10.1029/2024JH000318

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.

Palmer, D. S., O'Boyle, N. M., Glen, R. C., & Mitchell, J. B. O. (2007). Random forest models to predict aqueous solubility. Journal of Chemical Information and Modeling 47(1):150 – 8. https://doi.org/10.1021/ci060164k

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.

Perkins, T. K., & Gonzalez, J. A. (1984). Changes in earth stresses around a wellbore caused by radially symmetrical pressure and temperatura gradients. Society of Petroleum Engineers Journal 24(02), 129-140.

Python Software Foundation. (2023, May 17). Python 3.12.1. Retrieved from https://www.python.org/downloads/release/python-3913/

Rousseeuw, P.J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65. https://doi.org/10.1016/0377-0427(87)90125-7.

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.

Steinhaus, H. (1956). Sur la division des corps matériels en parties. Bulletin de l'Académie Polonaise des Sciences, Classe III, vol. IV, no. 12, 801–804.

Svetnik, V., Liaw, A., Tong, C., Culberson C. J., Sheridan, R. P., & Feuston, B. P. (2003). Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. Journal of Chemical Information & Computer Sciences 43, 1947 – 58. https://doi.org/10.1021/ci034160g.

Tao, Q., & Ghassemi, A. (2010). Poro-thermoelastic borehole stress analysis for determination of the in-situ stress and rock strength. Geothermics 39(3), 250-259.

Wang, C., Deng, C., & Wang, S. (2002). Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost. Pattern Recognition Letters 136, 190-197.

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

Yang, X., Wang, J., & Zhang, I. (2017). Application of XGBoost in iltra-short load forecasting. Electric Drive Automation 39, 21-25.

Yin, Z., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923.

Zhang, Y., Chen, W., Wang, J., & Lin, X. (2017). Deep regression networks for predicting physical properties. IEEE Transactions on Neural Networks and Learning Systems, 28(8), 1715–1726. https://doi.org/10.1109/TNNLS.2016.2566058

Zhao, T., Zheng S., Li, W., & Liu, K. (2018). Research on credit risk analysis based on XGBoost. Software Engineering 21 (6), 29-32.

Zhou, Y., Xu, X., Zhang, C., Xiong, Y., & Lin, W. (2016). Beyond Short Snippets: Deep Networks for Video Captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4707–4715.

Zhu, X., Li, Y., Xia, J., Yan, N., & Wang, Q. (2021). Geospatial Regression From Satellite Imagery With Convolutional Neural Networks. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, 4920–4933. https://doi.org/10.1109/JSTARS.2021.3076819

Appendices

Appendix A: Graphical representation of TUV Datasets of five cores used for DL/ML models.

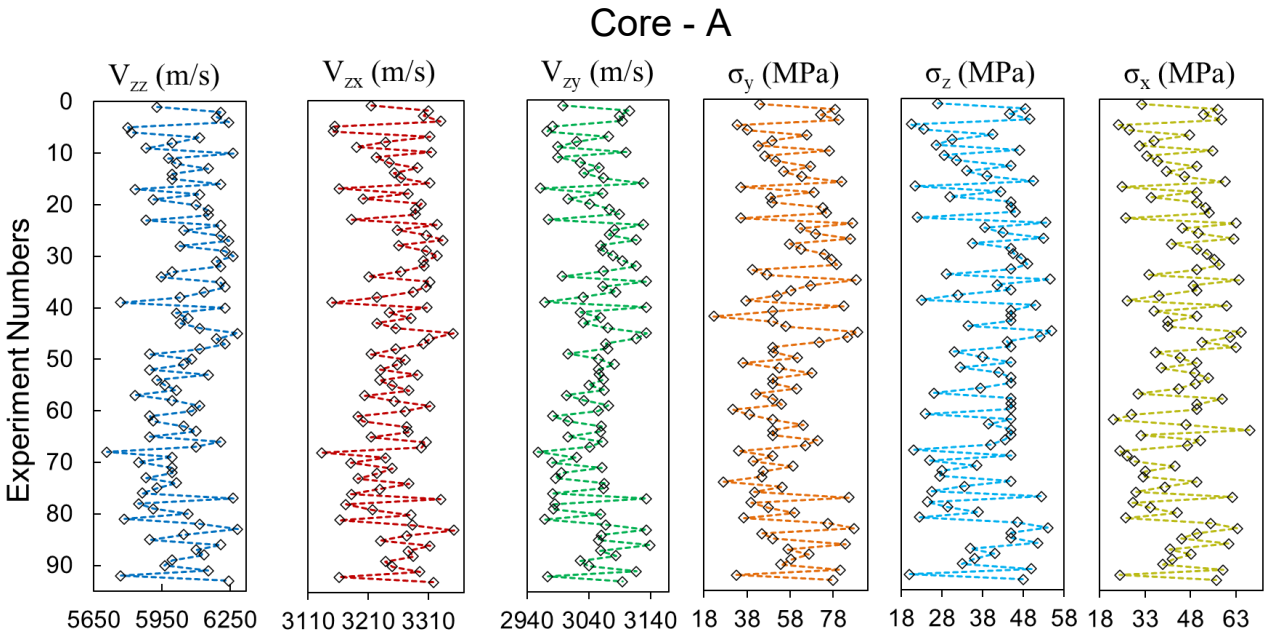


Figure A-1: The suite of TUV dataset for core-A

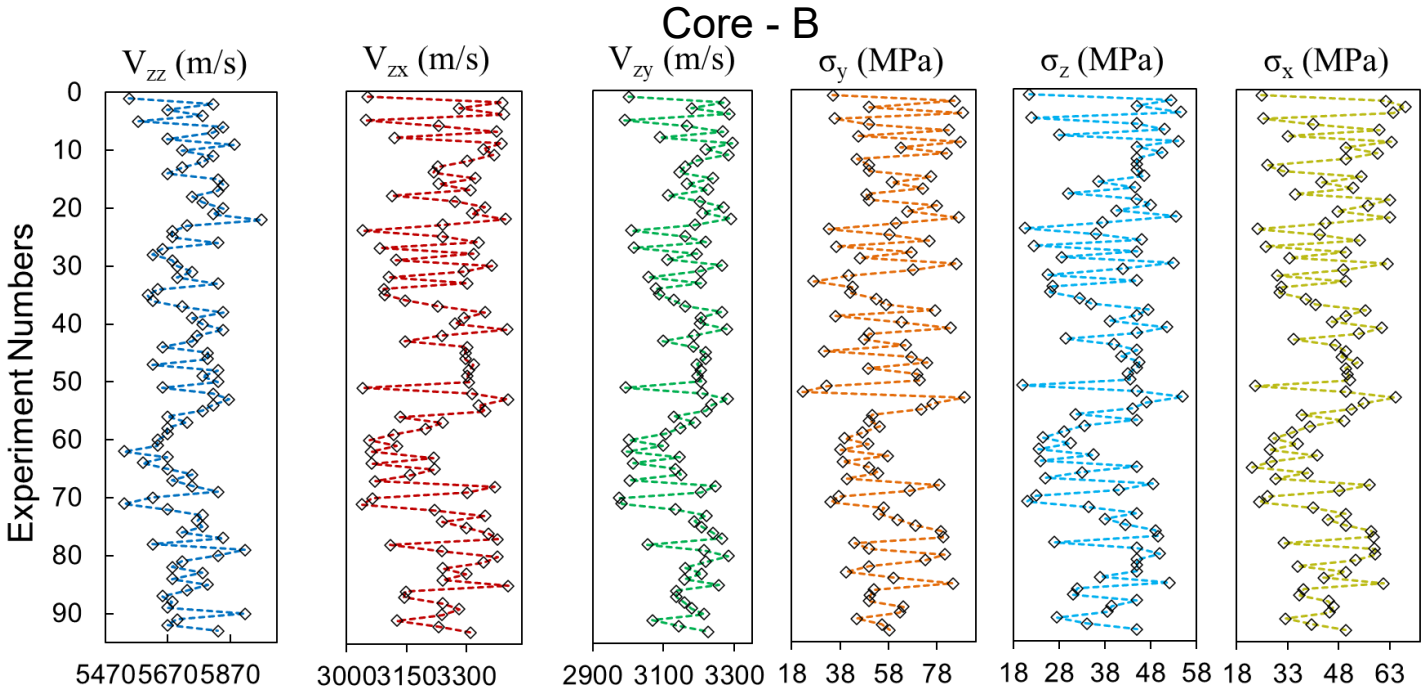


Figure A-2: The suite of TUV dataset for core-B

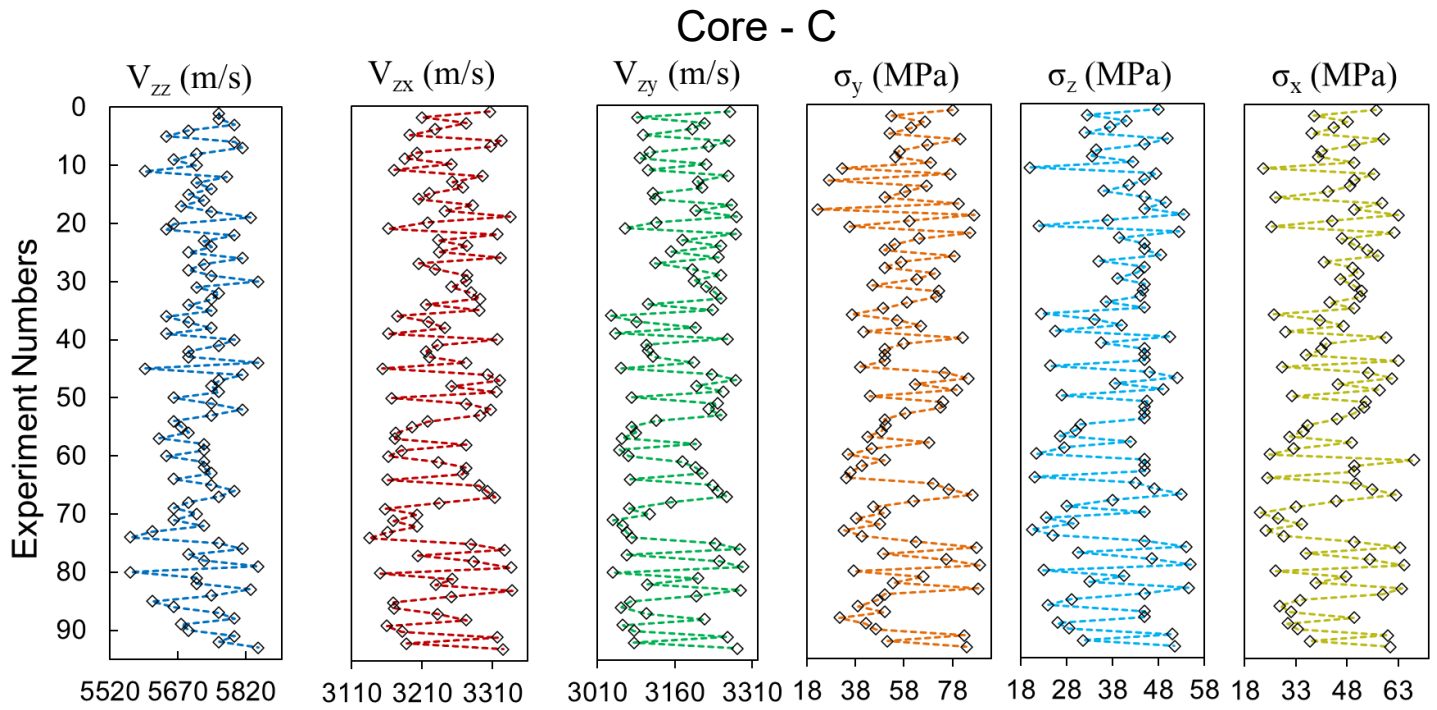


Figure A-3: The suite of TUV dataset for core-C

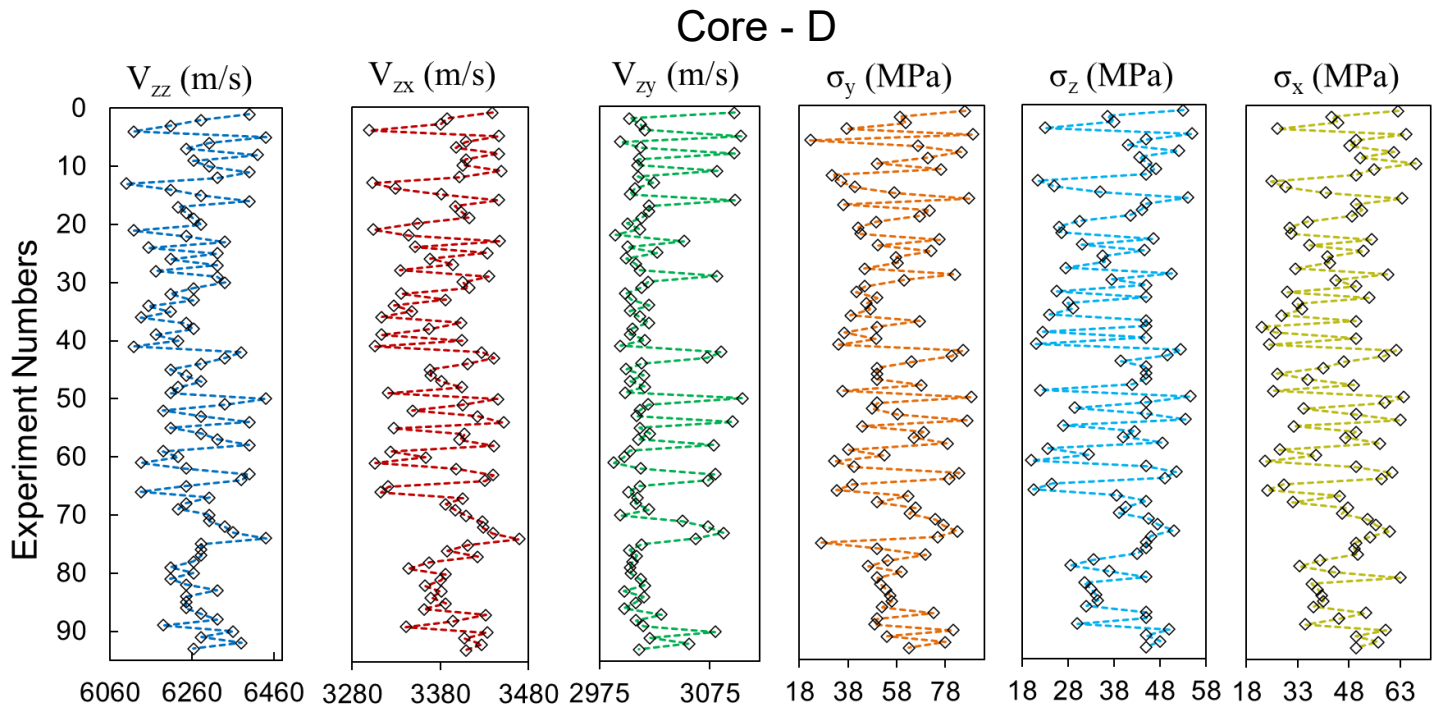


Figure A-4: The suite of TUV dataset for core-D

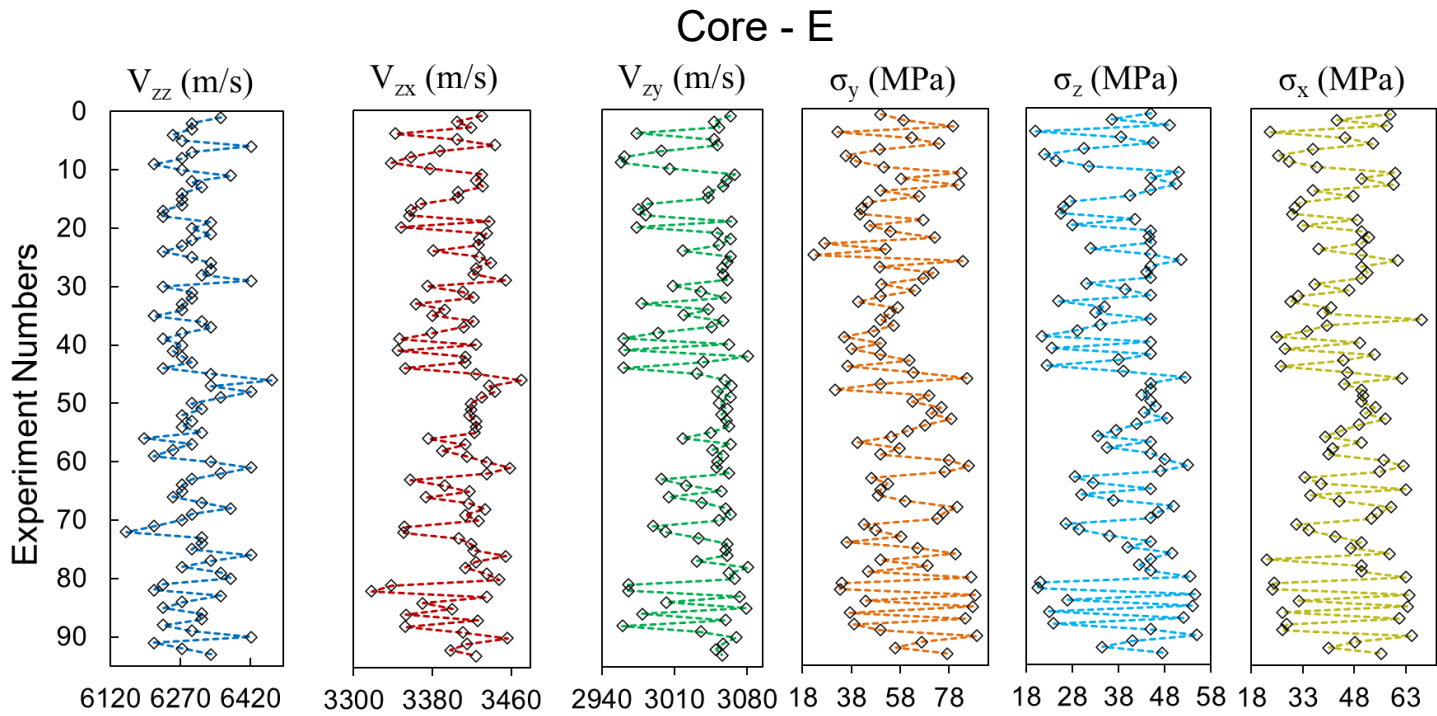


Figure A-5: The suite of TUV dataset for core-E

Appendix B: Histograms of datasets for cores-B, -C, -D, and -E.

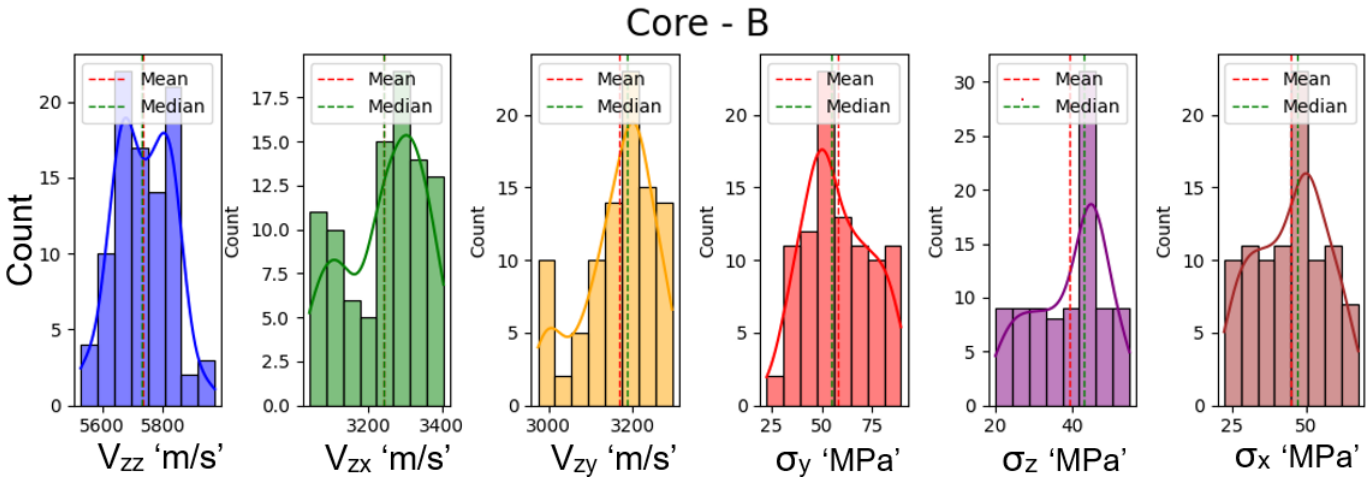


Figure B-1: Histograms of TUV dataset for core-B

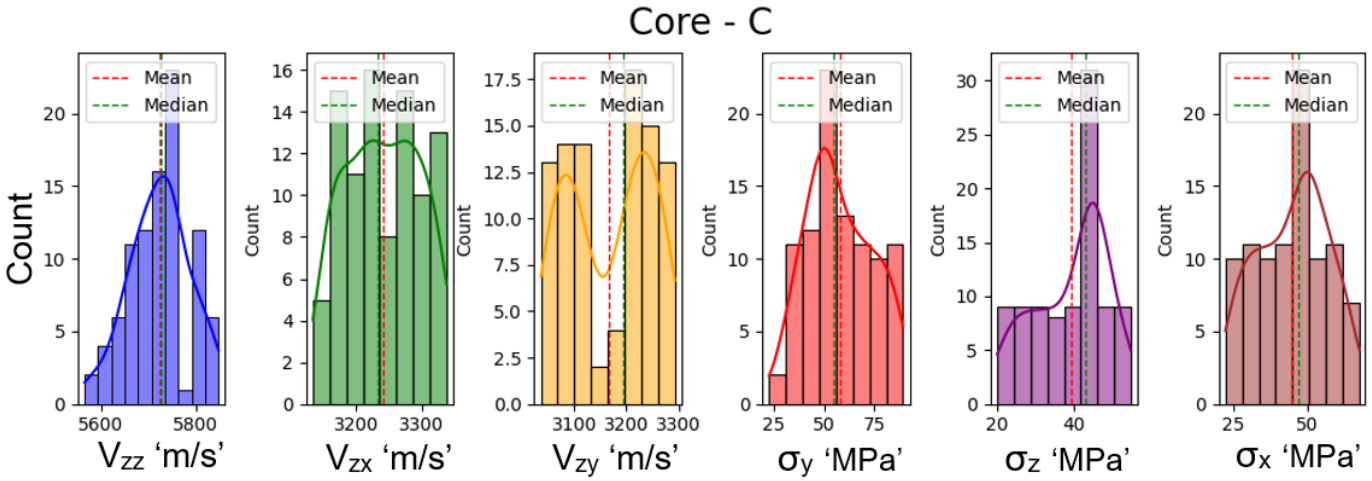


Figure B-2: Histograms of TUV dataset for core-C

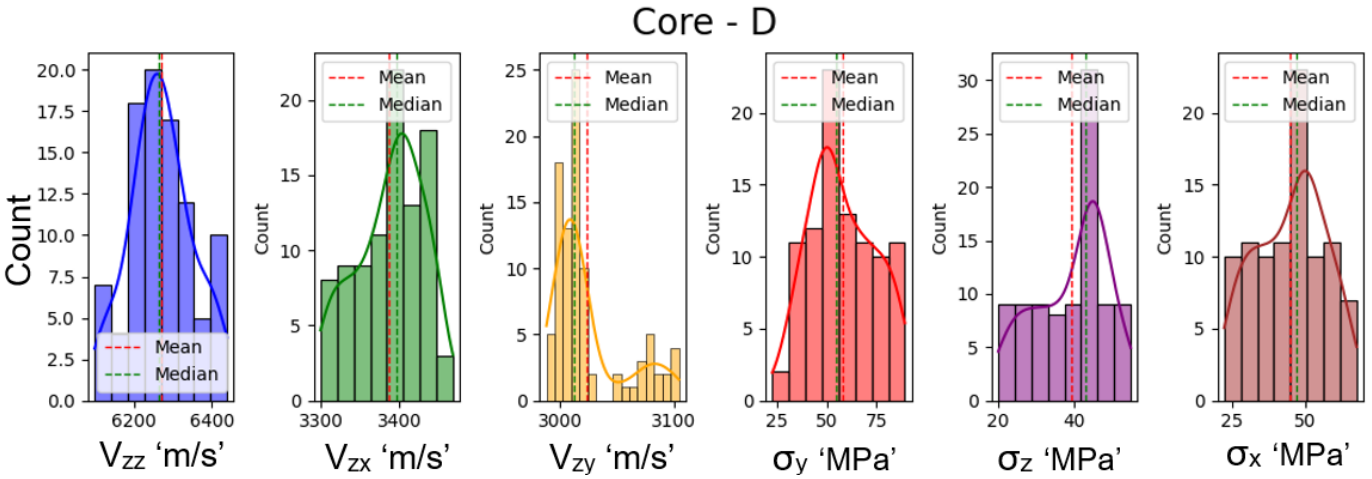


Figure B-3: Histograms of TUV dataset for core-D

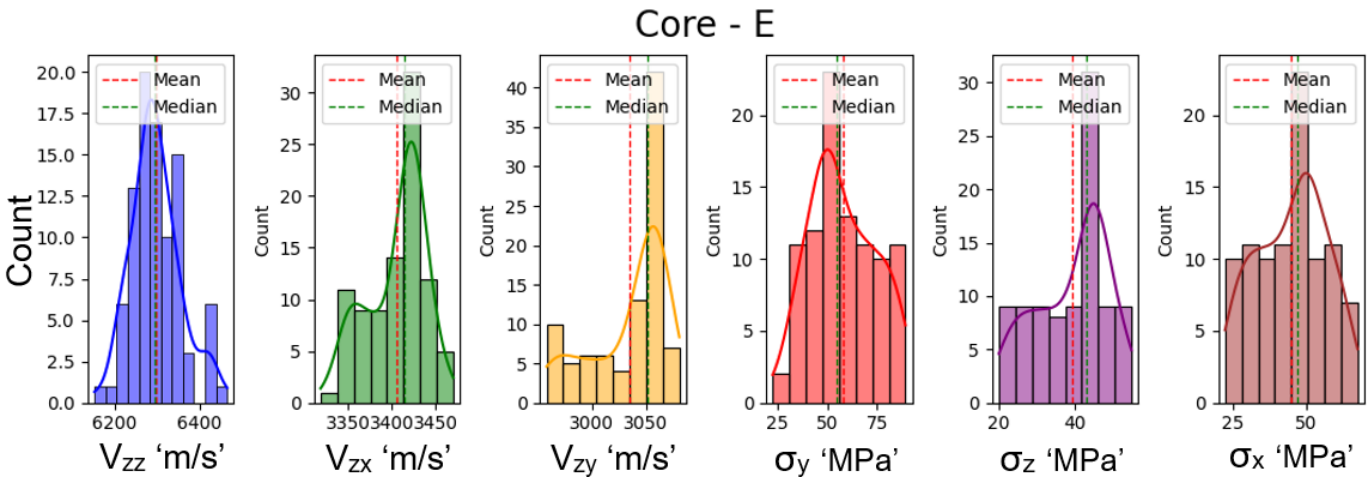


Figure B-4: Histograms of TUV dataset for core-E

Appendix C: Violin Plots of datasets for cores-B, C, D, and E

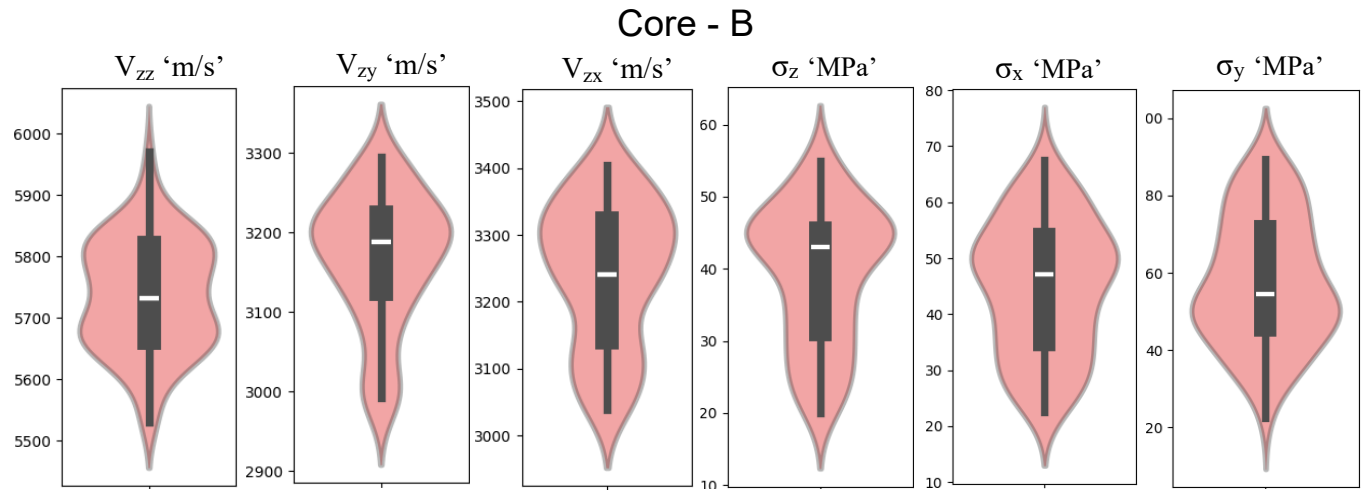


Figure C-1: Violin plots of TUV dataset for core-B

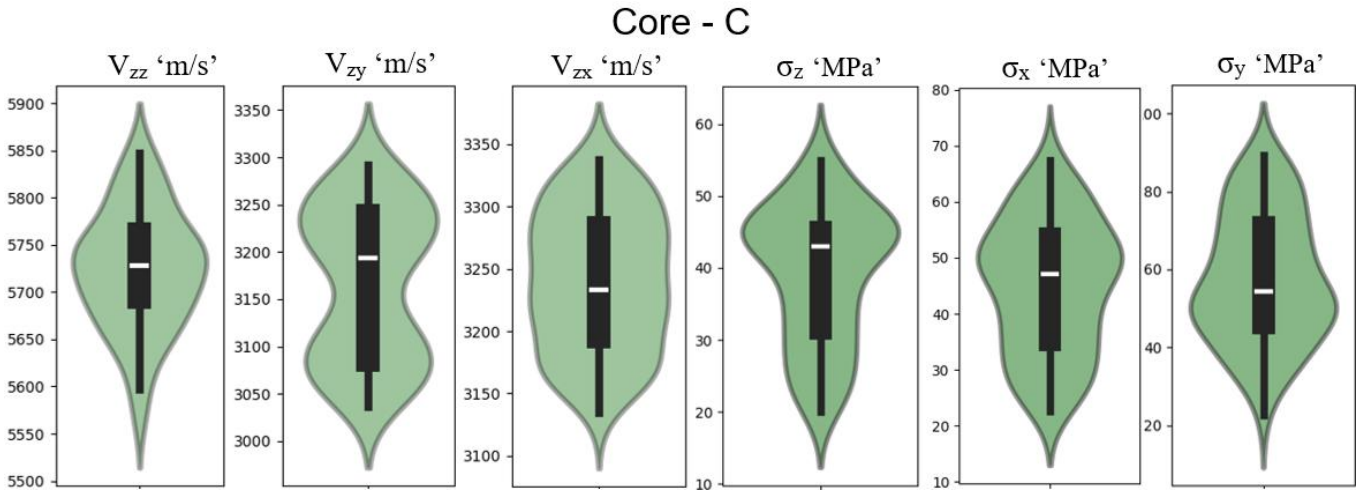


Figure C-2: Violin plots of TUV dataset for core-C

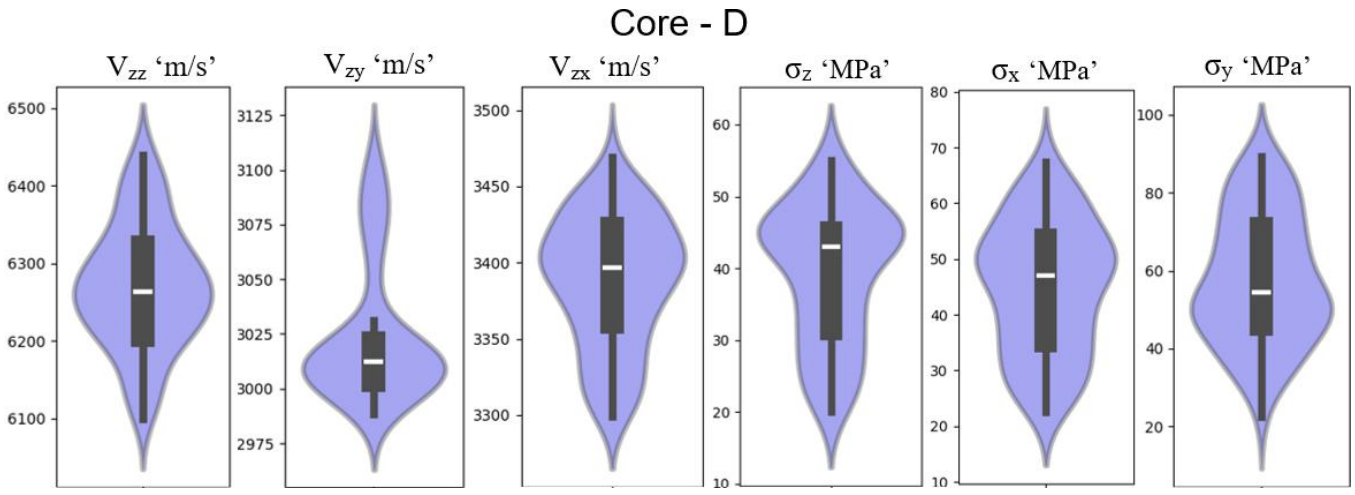


Figure C-3: Violin plots of TUV dataset for core-D

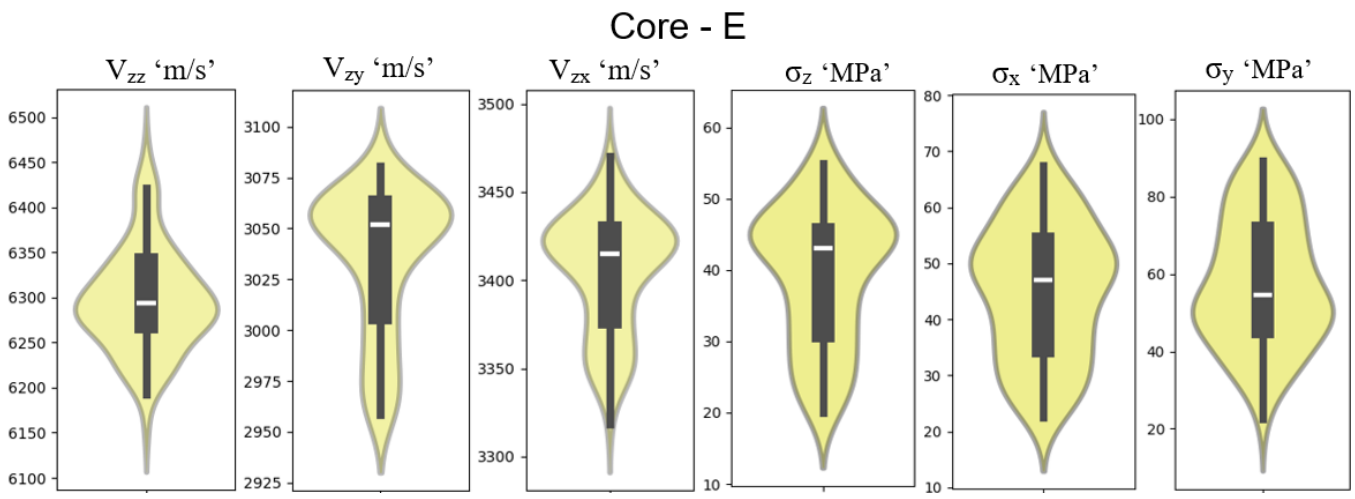


Figure C-4: Violin plots of TUV dataset for core-E

Appendix D: Heatmaps shows collinearity between each pair of input and output features.

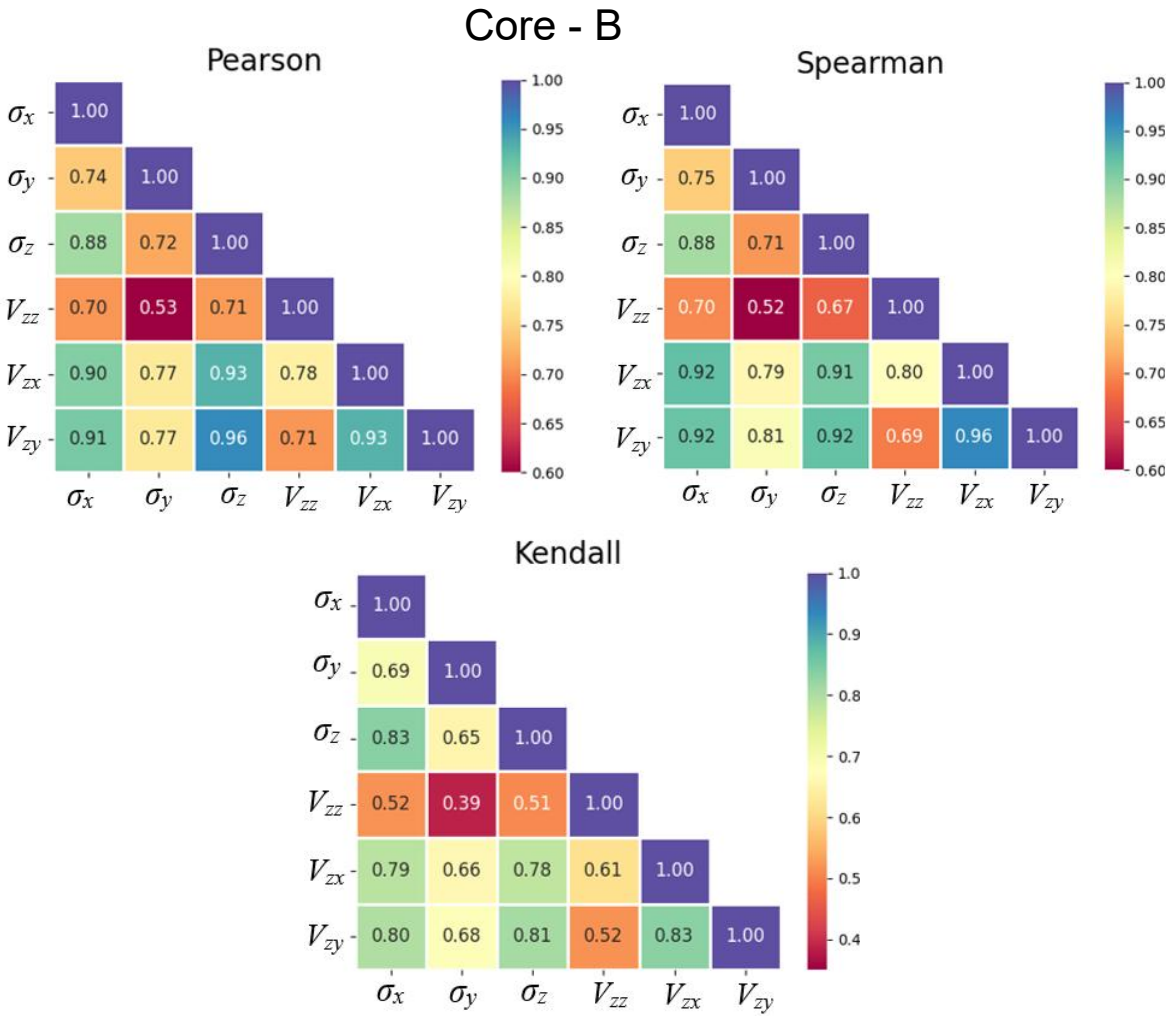


Figure D-1: Heatmaps showing collinearity between each pair of input and output features for core-B.

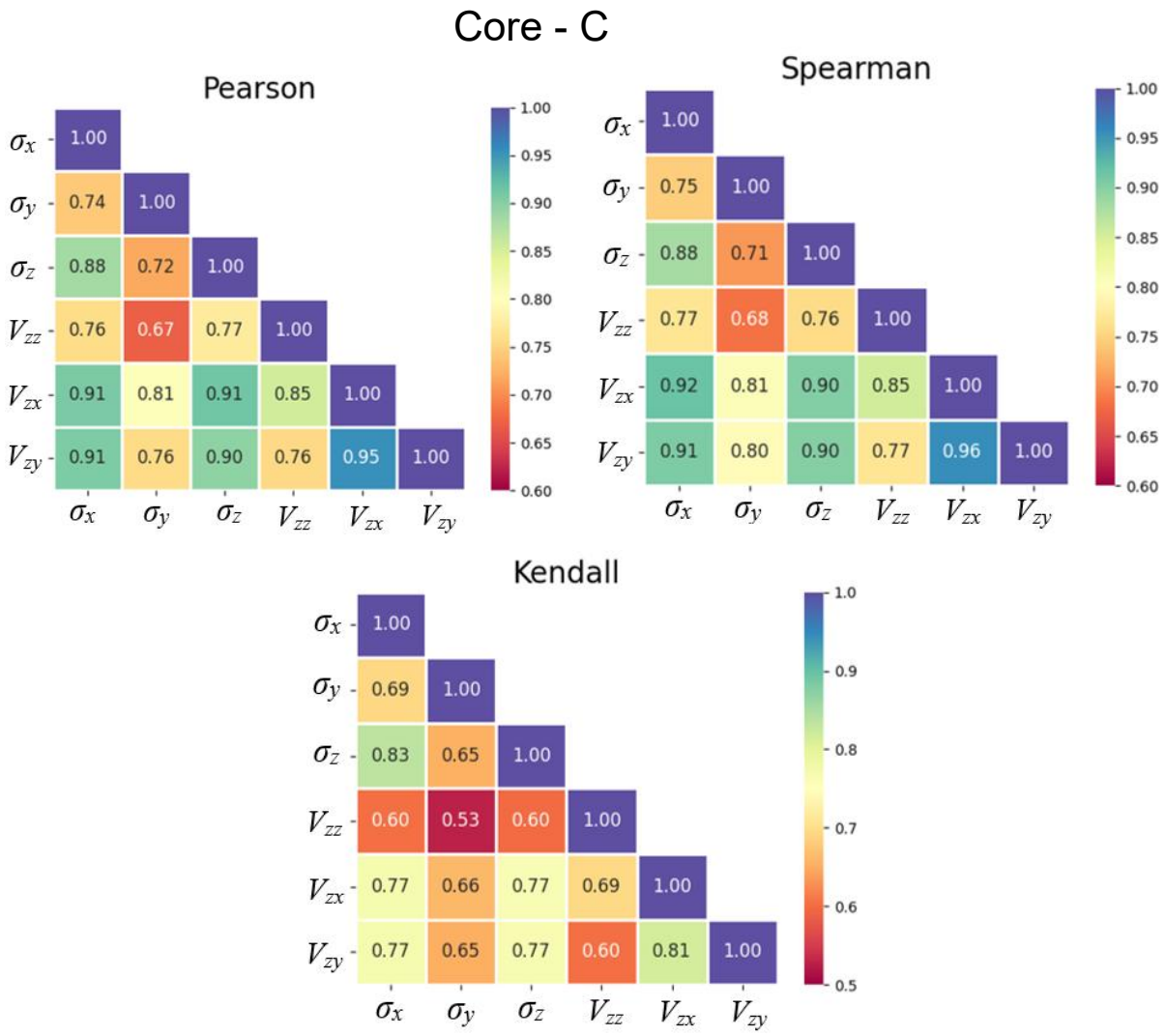


Figure D-2: Heatmaps showing collinearity between each pair of input and output features for core-C.

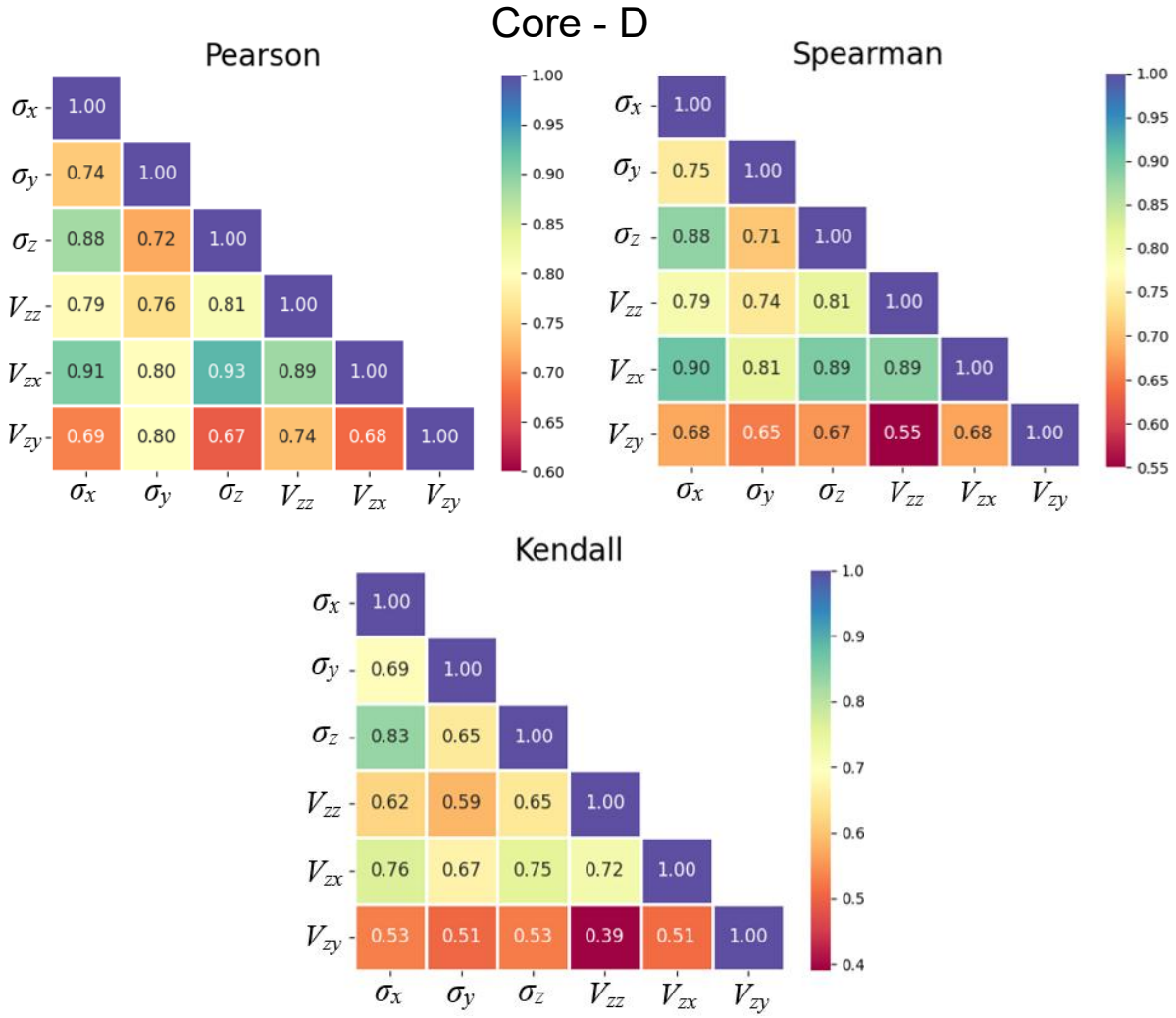


Figure D-3: Heatmaps showing collinearity between each pair of input and output features for core-D.

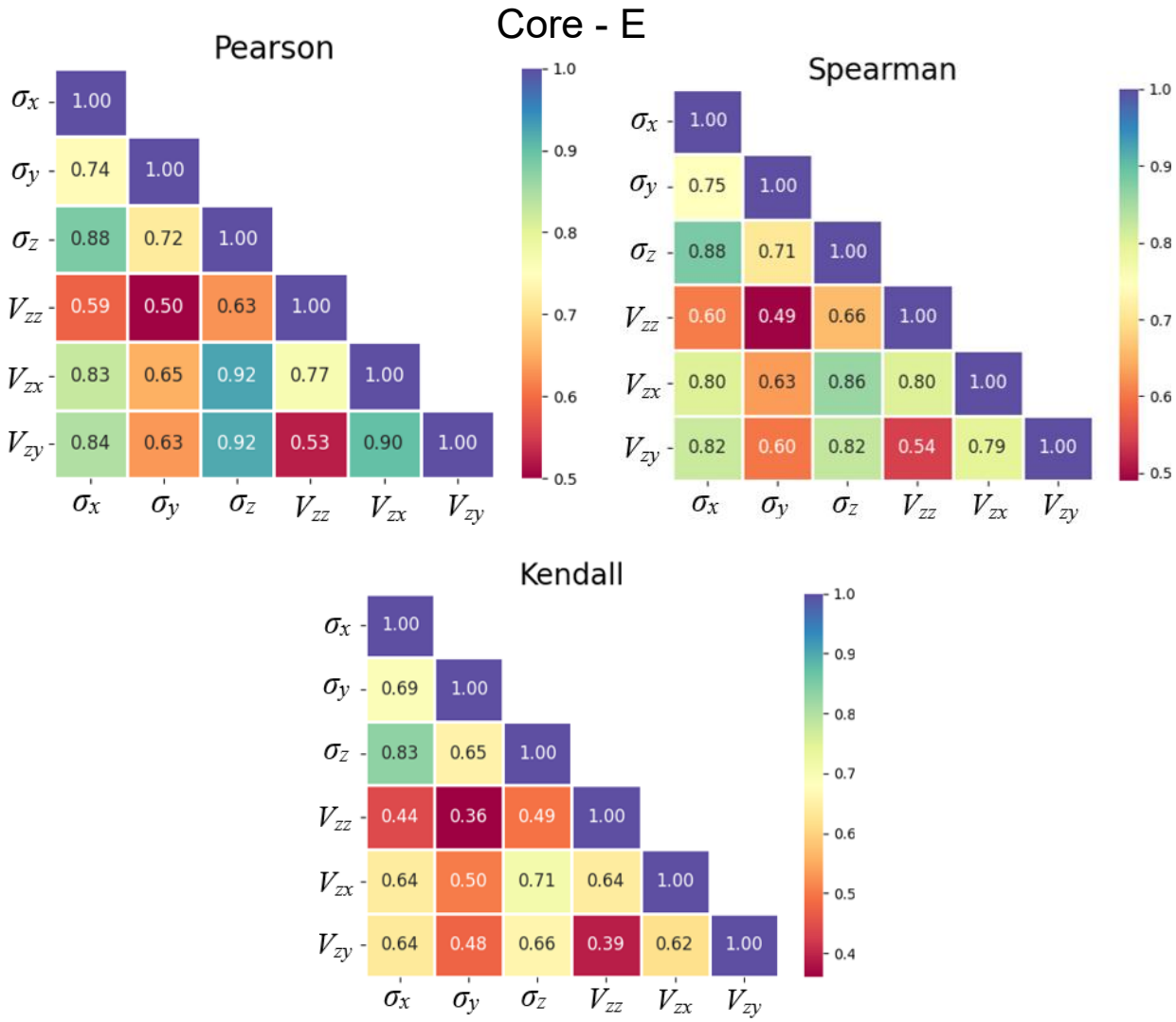


Figure D-4: Heatmaps showing collinearity between each pair of input and output features for core-E.

Appendix E: Relative importance of input features for σ_y, σ_z, and σ_x stress models.

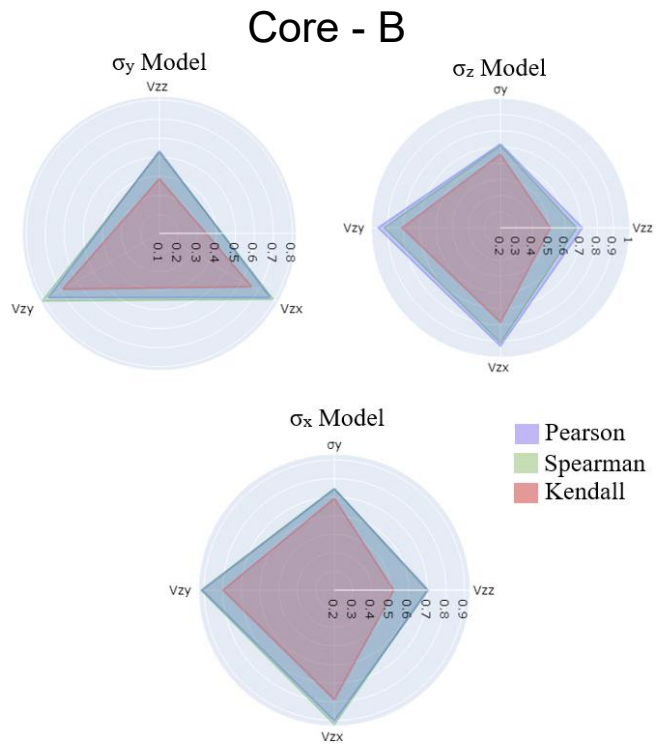


Figure E-1: Radar chart illustrates relative importance of input features for σ_y, σ_z, and σ_x stress models of core-B.

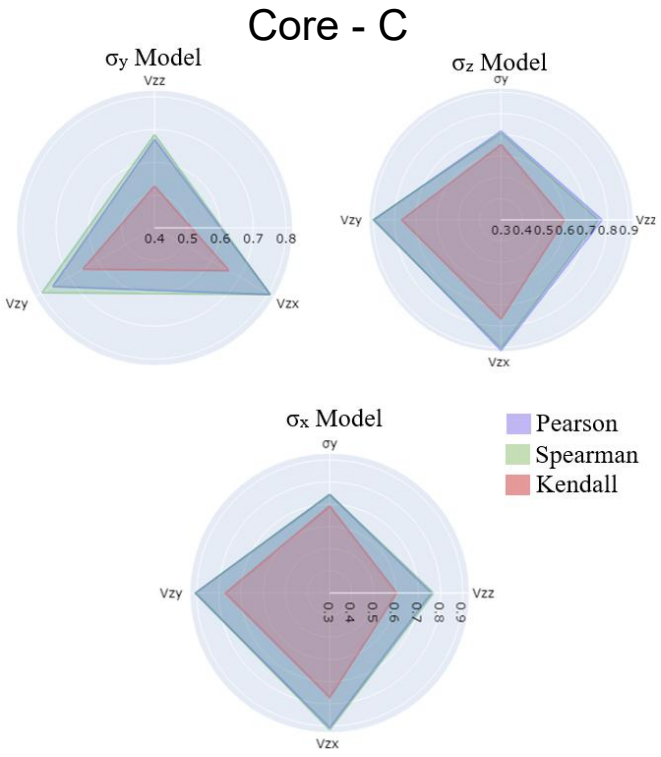


Figure E-2: Radar chart illustrates relative importance of input features for σ_y, σ_z, and σ_x stress models of core-C.

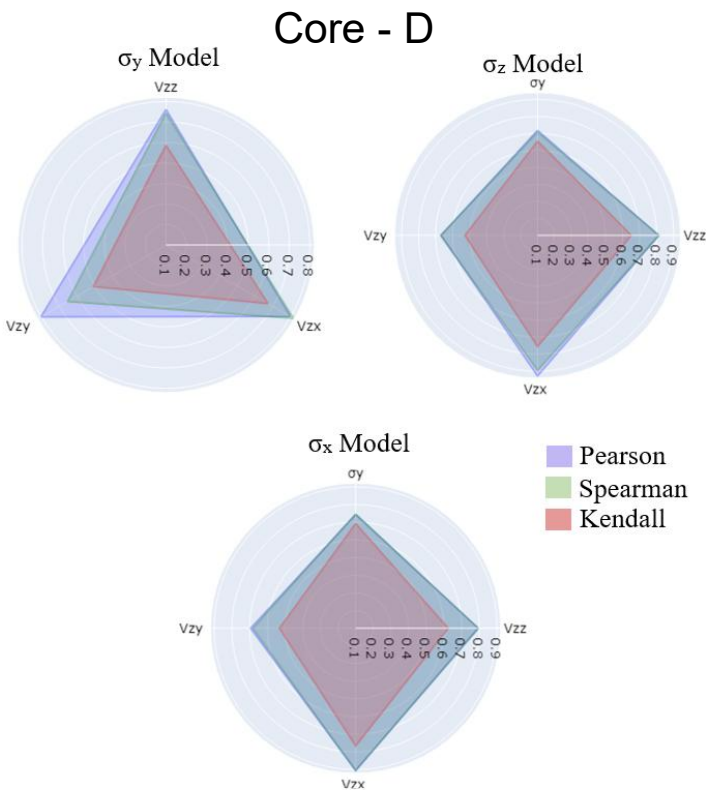


Figure E-3: Radar chart illustrates relative importance of input features for σ_y, σ_z, and σ_x stress models of core-D.

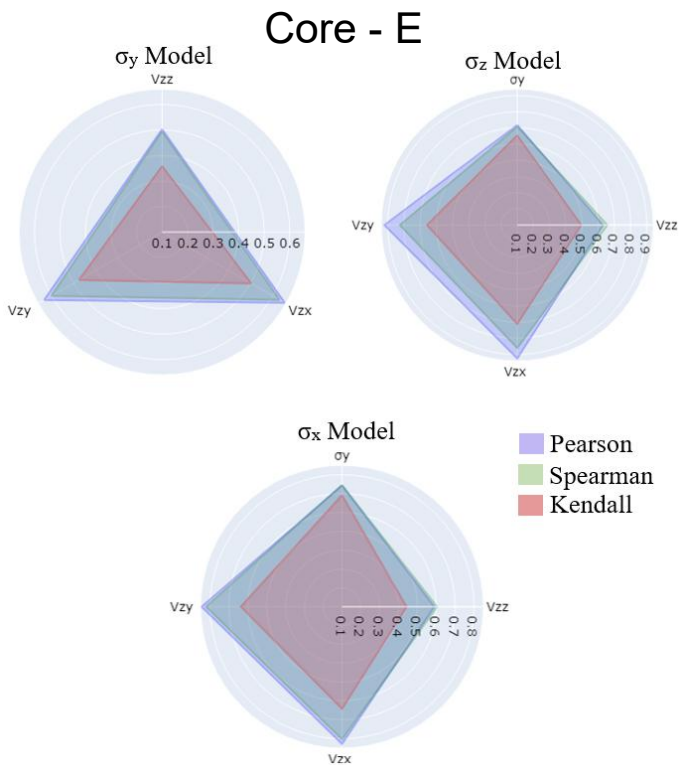


Figure E-4: Radar chart illustrates relative importance of input features for σ_y, σ_z, and σ_x stress models of core-E.

Appendix F: Neurons sensitivity analysis using evaluation metrics (RMSE and R^2) of σ_z, σ_x, σ_y models for Core-B, -C, -D, and -E.

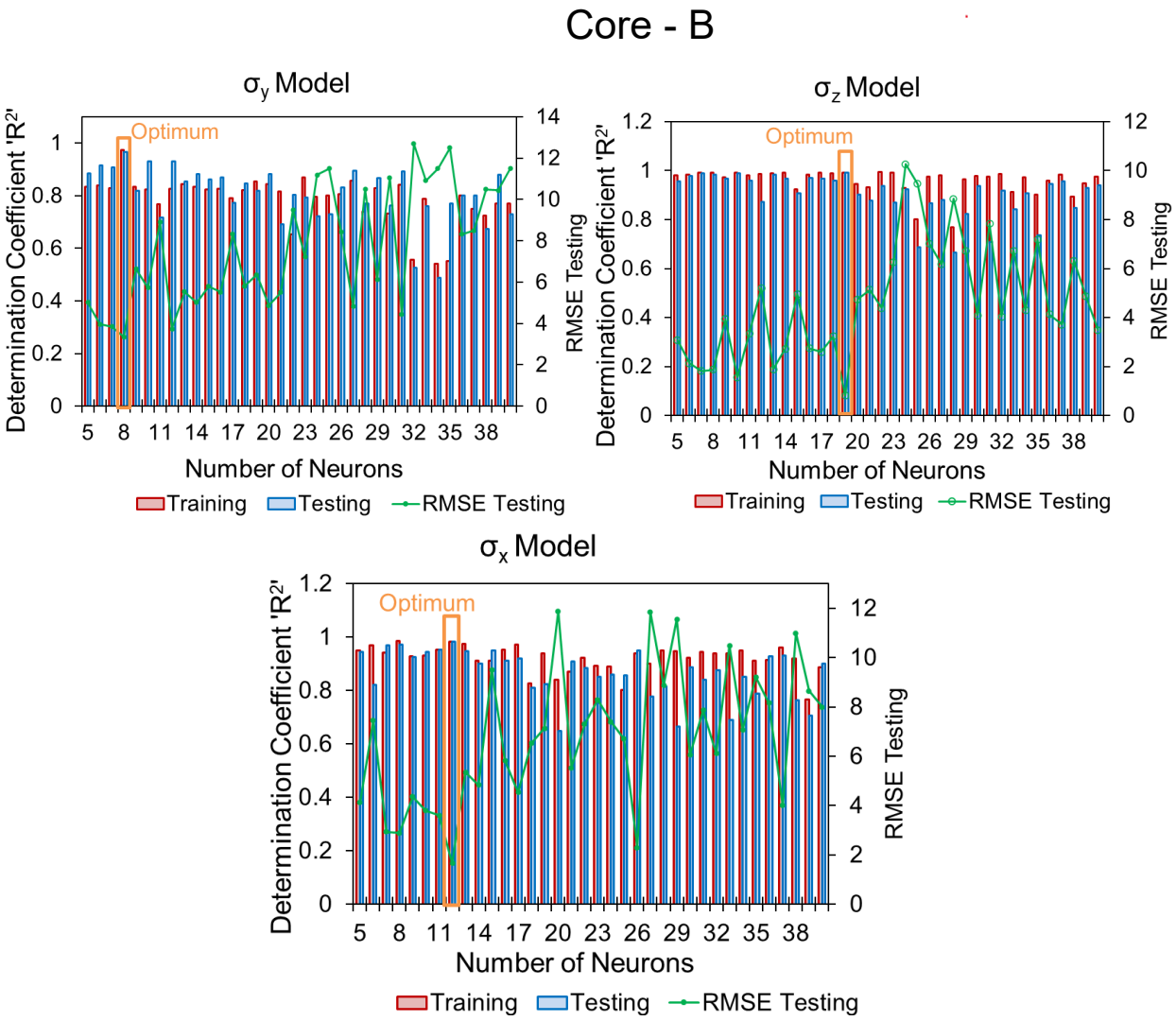


Figure F-1: Neurons sensitivity analysis by comparing evaluation metrics of σ_z, σ_x, σ_y models for core-B.

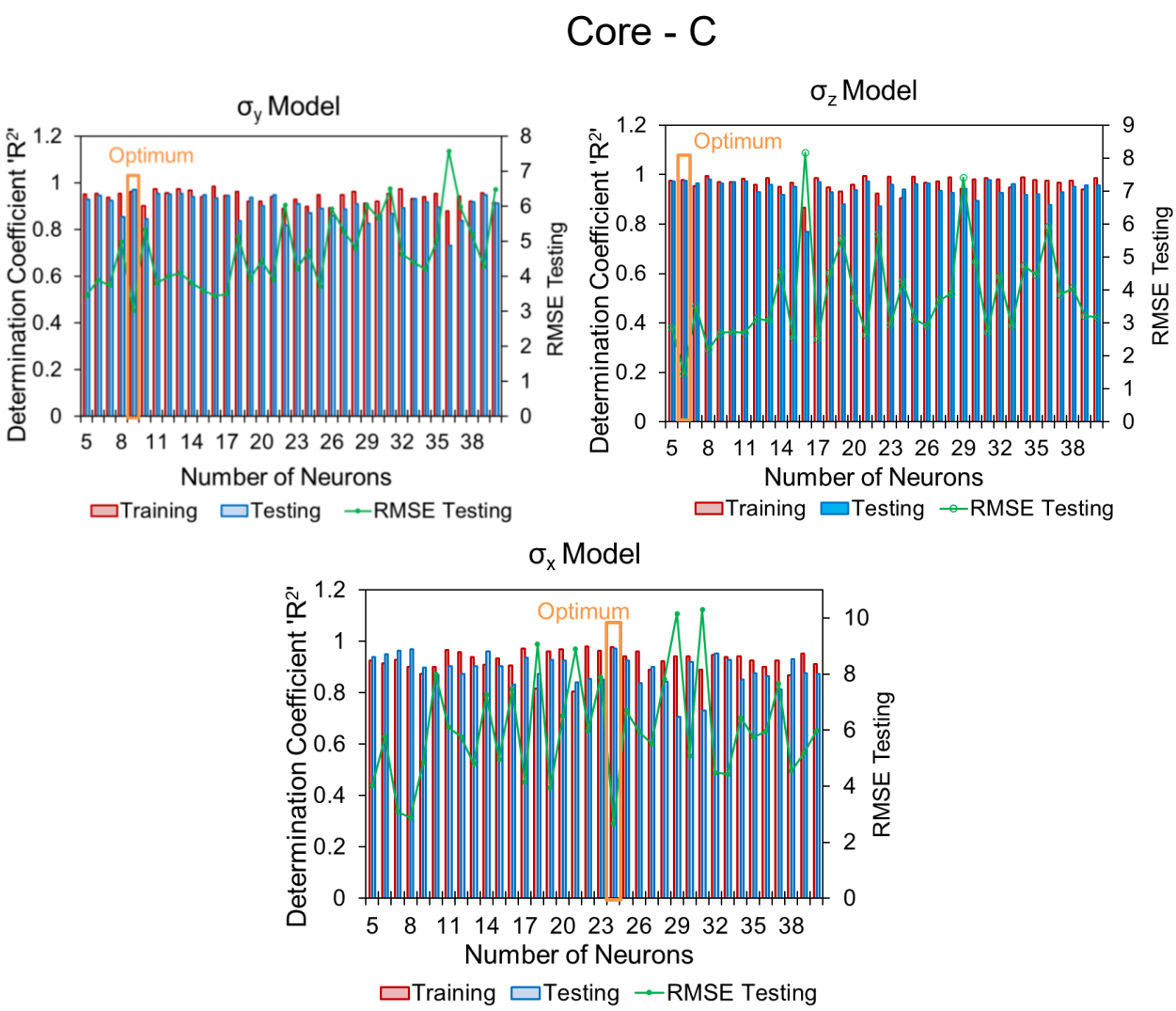


Figure F-2: Neurons sensitivity analysis by comparing evaluation metrics of σ_z, σ_x, σ_y models for core-C.

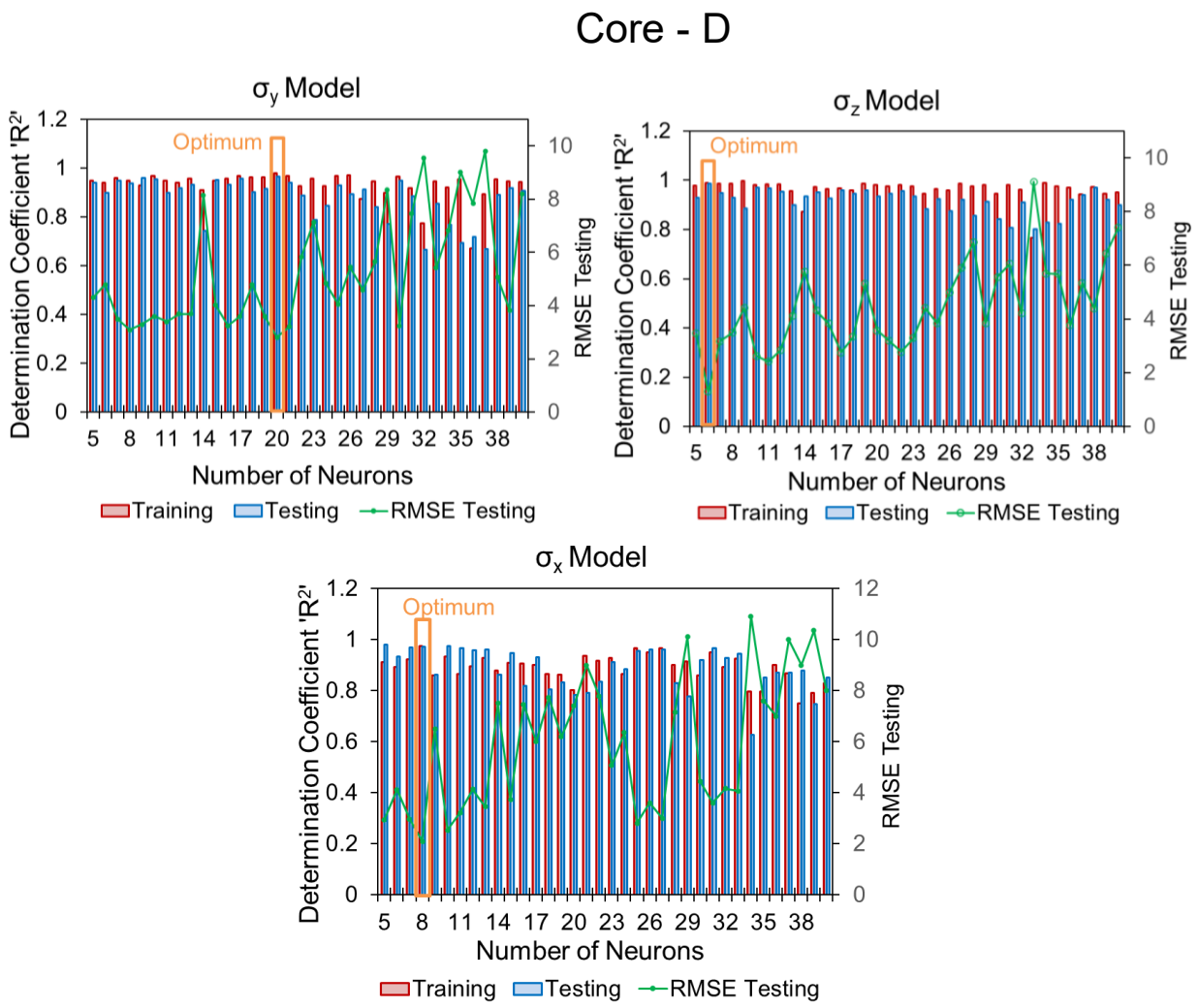


Figure F-3: Neurons sensitivity analysis by comparing evaluation metrics of σ_z, σ_x, σ_y models for core-D.

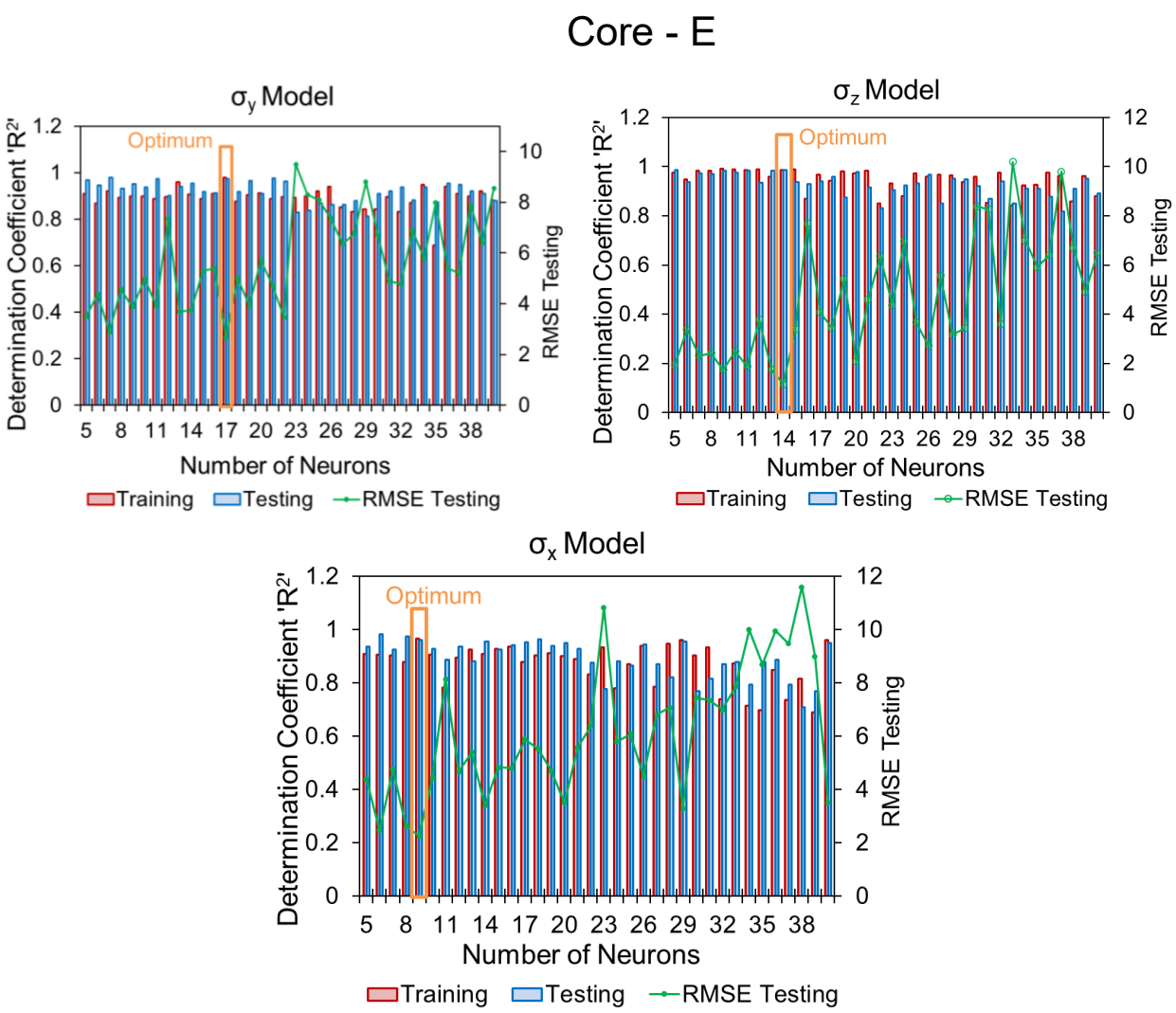


Figure F-4: Neurons sensitivity analysis by comparing evaluation metrics of σ_z, σ_x, σ_y models for core-E.

Appendix G: Optimum values of hyperparameters of the proposed DL/ML models

Table G-1: Optimum values of hyperparameters used for σ_z, σ_x, σ_y models of Core-B.

Model Type	Hyperparameter	Tested Values	Selected value for 'σ_y'	Selected value for 'σ_z'	Selected value for 'σ_x'
DNN	Number of hidden layers	1-3	2	2	2
	Number of neurons	5-40	8	19	12
	Number of realizations	1-100	82	76	48
	Optimizer	Adam, nadam, sgd	Adam	Adam	nadam
	Activation function	ReLU, LeakyReLU, Tanh	ReLU	ReLU	LeakyReLU
	Learning rate	$1x10^{-4}$ – $1x10^{-1}$	$1x10^{-3}$	$1x10^{-4}$	$1x10^{-4}$
	Batch size	8, 16, 32, 64	32	32	32
	Dropout Rate	0.1-0.5	0.2	0.2	0.2
	Loss Function	MSE	MSE	MSE	MSE
GRU	GRU Units	10-200	80	75	80
	Activation Function	ReLU, LeakyReLU, Tanh	ReLU	ReLU	ReLU
	Dropout rate	0.1-0.5	0.2	0.2	0.2
	Batch Size	8, 16, 32, 64	32	32	32
	Learning rate	$1x10^{-4}$ – $1x10^{-1}$	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-3}$
	Optimizer	Adam, nadam, sgd	Adam	nadam	nadam
	Loss Function	MSE	MSE	MSE	MSE
CNN	Number of Filters	8-64	16	32	32
	Kernel Size	2-5	4	4	4
	Activation Function for Conv1D layer	ReLU, LeakyReLU, Tanh	ReLU	ReLU	ReLU
	Dense Units	8-64	32	64	64
	Batch Size	8, 16, 32, 64	32	32	32
	Optimizer	Adam, nadam, sgd	Adam	Adam	Adam
	Loss Function	MSE	MSE	MSE	MSE
RF	Sample count necessary to split the internal node	2-20	12	15	15
	Maximum depth of the trees	3-10	8	9	8
	Total count of trees in the forest	50-2500	1000	1200	1200
	Sample count at the leaf node	1-5	2	2	2
XGB	Criterion	-	friedman mse	friedman mse	friedman mse
	Count of boosting stages	100-1000	250	300	300
	Learning rate	0.001-0.1	0.005	0.005	0.005
	Alpha	-	0.03	0.03	0.02
	Minimum sample split	1-5	3	3	3
	Max. depth	3-10	8	8	10
	Minimum sample leaf	1-5	2	2	2
	Loss	MSE, RMSE	MSE	MSE	MSE

Table G-2: Optimum values of hyperparameters used for σ_z, σ_x, σ_y models of Core-C.

Model Type	Hyperparameter	Tested Values	Selected value for 'σ_y'	Selected value for 'σ_z'	Selected value for 'σ_x'
DNN	Number of hidden layers	1-3	2	2	2
	Number of neurons	5-40	9	6	24
	Number of realizations	1-100	100	92	50
	Optimizer	Adam, nadam, sgd	nadam	nadam	nadam
	Activation function	ReLU, LeakyReLU, Tanh	ReLU	LeakyReLU	LeakyReLU
	Learning rate	$1x10^{-4}$ – $1x10^{-1}$	$1x10^{-4}$	$1x10^{-4}$	$1x10^{-4}$
	Batch size	8, 16, 32, 64	16	32	32
	Dropout Rate	0.1-0.5	0.2	0.2	0.2
	Loss Function	MSE	MSE	MSE	MSE
GRU	GRU Units	10-200	80	85	85
	Activation Function	ReLU, LeakyReLU, Tanh	ReLU	ReLU	ReLU
	Dropout rate	0.1-0.5	0.2	0.2	0.2
	Batch Size	8, 16, 32, 64	32	32	32
	Learning rate	$1x10^{-4}$ – $1x10^{-1}$	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-3}$
	Optimizer	Adam, nadam, sgd	Adam	Adam	Adam
	Loss Function	MSE	MSE	MSE	MSE
CNN	Number of Filters	8-64	32	32	32
	Kernel Size	2-5	4	4	4
	Activation Function for Conv1D layer	ReLU, LeakyReLU, Tanh	LeakyReLU	LeakyReLU	LeakyReLU
	Dense Units	8-64	32	32	32
	Batch Size	8, 16, 32, 64	16	16	32
	Optimizer	Adam, nadam, sgd	sgd	sgd	sgd
	Loss Function	MSE	MSE	MSE	MSE
RF	Sample count necessary to split the internal node	2-20	12	14	15
	Maximum depth of the trees	3-10	8	8	9
	Total count of trees in the forest	50-2500	1100	1000	1000
	Sample count at the leaf node	1-5	2	2	2
XGB	Criterion	-	friedman mse	friedman mse	friedman mse
	Count of boosting stages	100-1000	300	400	400
	Learning rate	0.001-0.1	0.005	0.001	0.01
	Alpha	-	0.02	0.03	0.03
	Minimum sample split	1-5	2	2	2
	Max. depth	3-10	8	9	8
	Minimum sample leaf	1-5	2	2	2
	Loss	MSE, RMSE	MSE	MSE	MSE

Table G-3: Optimum values of hyperparameters used for σ_z, σ_x, σ_y models of Core-D.

Model Type	Hyperparameter	Tested Values	Selected value for 'σ_y'	Selected value for 'σ_z'	Selected value for 'σ_x'
DNN	Number of hidden layers	1-3	2	2	2
	Number of neurons	5-40	20	6	8
	Number of realizations	1-100	7	94	34
	Optimizer	Adam, nadam, sgd	Sgd	Sgd	Sgd
	Activation function	ReLU, LeakyReLU, Tanh	ReLU	ReLU	ReLU
	Learning rate	$1x10^{-4}$ – $1x10^{-1}$	$1x10^{-4}$	$1x10^{-3}$	$1x10^{-3}$
	Batch size	8, 16, 32, 64	16	32	16
	Dropout Rate	0.1-0.5	0.2	0.2	0.2
	Loss Function	MSE	MSE	MSE	MSE
GRU	GRU Units	10-200	100	90	100
	Activation Function	ReLU, LeakyReLU, Tanh	ReLU	Tanh	ReLU
	Dropout rate	0.1-0.5	0.2	0.2	0.2
	Batch Size	8, 16, 32, 64	32	16	16
	Learning rate	$1x10^{-4}$ – $1x10^{-1}$	$1x10^{-3}$	$1x10^{-4}$	$1x10^{-4}$
	Optimizer	Adam, nadam, sgd	Adam	nadam	nadam
	Loss Function	MSE	MSE	MSE	MSE
CNN	Number of Filters	8-64	32	16	16
	Kernel Size	2-5	2	3	3
	Activation Function for Conv1D layer	ReLU, LeakyReLU, Tanh	ReLU	ReLU	ReLU
	Dense Units	8-64	64	64	64
	Batch Size	8, 16, 32, 64	32	32	16
	Optimizer	Adam, nadam, sgd	nadam	Adam	Adam
	Loss Function	MSE	MSE	MSE	MSE
RF	Sample count necessary to split the internal node	2-20	11	10	15
	Maximum depth of the trees	3-10	6	8	8
	Total count of trees in the forest	50-2500	1400	1500	1300
	Sample count at the leaf node	1-5	2	2	2
XGB	Criterion	-	friedman mse	friedman mse	friedman mse
	Count of boosting stages	100-1000	350	300	300
	Learning rate	0.001-0.1	0.05	0.05	0.001
	Alpha	-	0.03	0.02	0.03
	Minimum sample split	1-5	3	3	2
	Max. depth	3-10	8	7	7
	Minimum sample leaf	1-5	2	2	2
	Loss	MSE, RMSE	MSE	MSE	MSE

Table G-4: Optimum values of hyperparameters used for σ_z, σ_x, σ_y models of Core-E.

Model Type	Hyperparameter	Tested Values	Selected value for 'σ_y'	Selected value for 'σ_z'	Selected value for 'σ_x'
DNN	Number of hidden layers	1-3	2	2	2
	Number of neurons	5-40	17	14	9
	Number of realizations	1-100	100	52	8
	Optimizer	Adam, nadam, sgd	Adam	nadam	nadam
	Activation function	ReLU, LeakyReLU, Tanh	LeakyReLU	LeakyReLU	LeakyReLU
	Learning rate	$1x10^{-4}$ – $1x10^{-1}$	$1x10^{-2}$	$1x10^{-2}$	$1x10^{-2}$
	Batch size	8, 16, 32, 64	16	32	32
	Dropout Rate	0.1-0.5	0.2	0.2	0.2
	Loss Function	MSE	MSE	MSE	MSE
GRU	GRU Units	10-200	70	80	100
	Activation Function	ReLU, LeakyReLU, Tanh	ReLU	ReLU	Tanh
	Dropout rate	0.1-0.5	0.2	0.2	0.2
	Batch Size	8, 16, 32, 64	32	16	16
	Learning rate	$1x10^{-4}$ – $1x10^{-1}$	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-2}$
	Optimizer	Adam, nadam, sgd	nadam	nadam	Adam
	Loss Function	MSE	MSE	MSE	MSE
CNN	Number of Filters	8-64	16	16	16
	Kernel Size	2-5	3	3	3
	Activation Function for Conv1D layer	ReLU, LeakyReLU, Tanh	ReLU	LeakyReLU	LeakyReLU
	Dense Units	8-64	64	64	64
	Batch Size	8, 16, 32, 64	32	32	16
	Optimizer	Adam, nadam, sgd	Adam	Sgd	Adam
	Loss Function	MSE	MSE	MSE	MSE
RF	Sample count necessary to split the internal node	2-20	15	14	12
	Maximum depth of the trees	3-10	5	7	6
	Total count of trees in the forest	50-2500	1000	900	1100
	Sample count at the leaf node	1-5	2	2	2
XGB	Criterion	-	friedman mse	friedman mse	friedman mse
	Count of boosting stages	100-1000	250	200	200
	Learning rate	0.001-0.1	0.01	0.01	0.01
	Alpha	-	0.02	0.02	0.02
	Minimum sample split	1-5	3	3	2
	Max. depth	3-10	5	6	6
	Minimum sample leaf	1-5	3	3	2
	Loss	MSE, RMSE	MSE	MSE	MSE

Appendix H: Cross plots between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_z, σ_x, σ_y stresses Core-B, -C, -D, and -E.

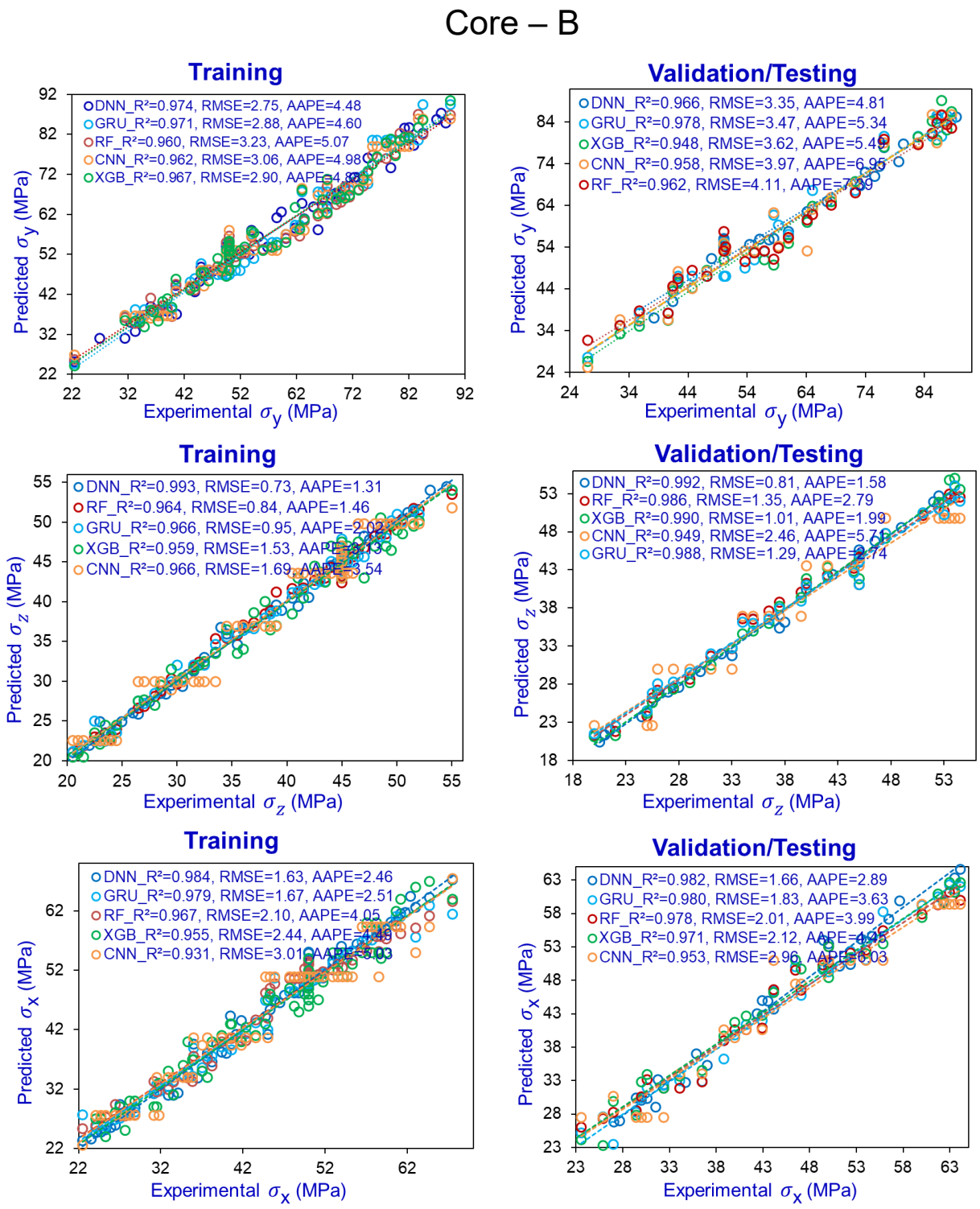


Figure H-1: Cross plots between predicted and experimental stresses for the proposed DNN, GRU, CNN, RF, XGB models of σ_z, σ_x, and σ_y, stresses for Core-B.

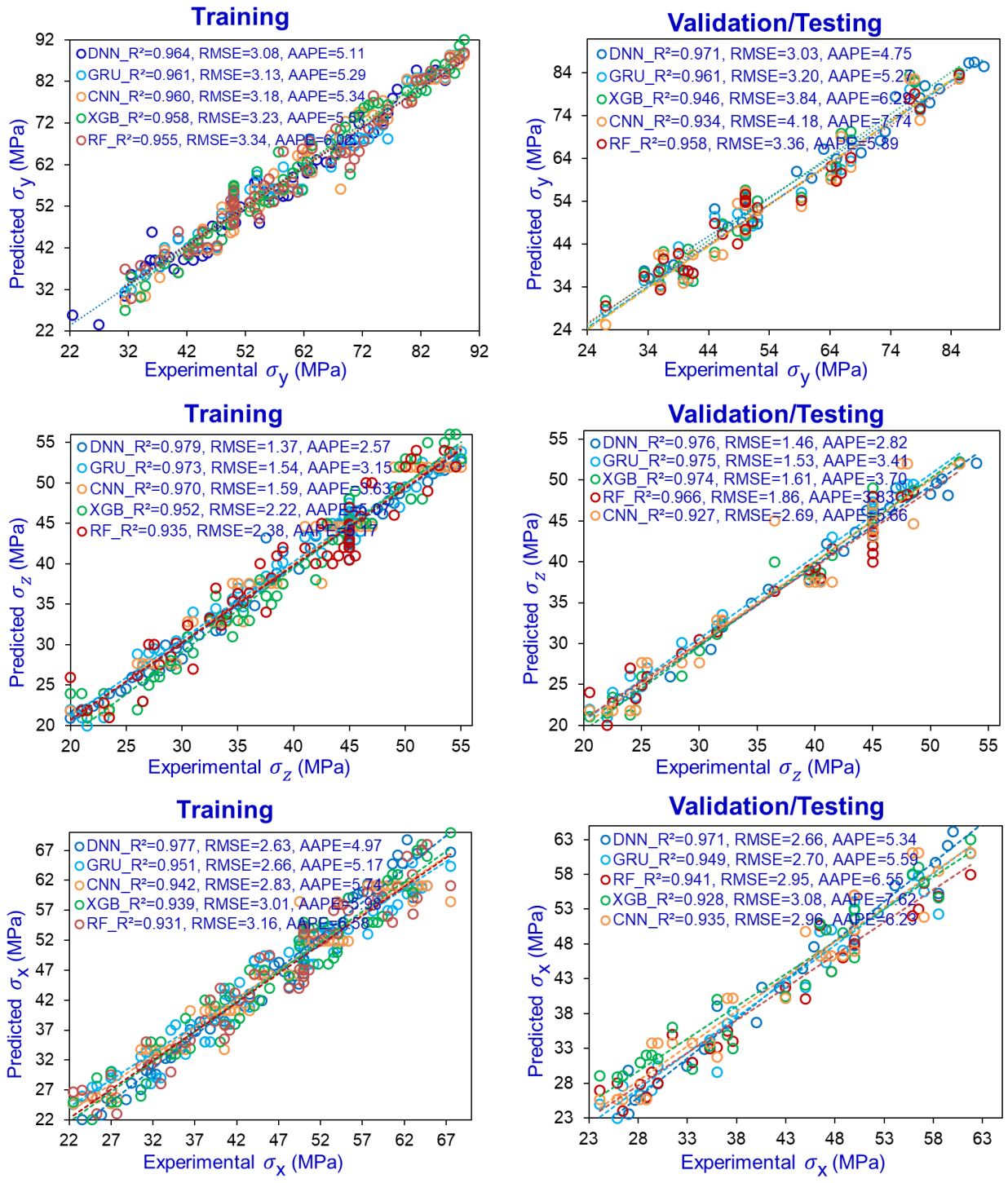


Figure H-2: Cross plots between predicted and experimental stresses for the proposed DNN, GRU, CNN, RF, XGB models of σ_z, σ_x, and σ_y, stresses for Core-C.

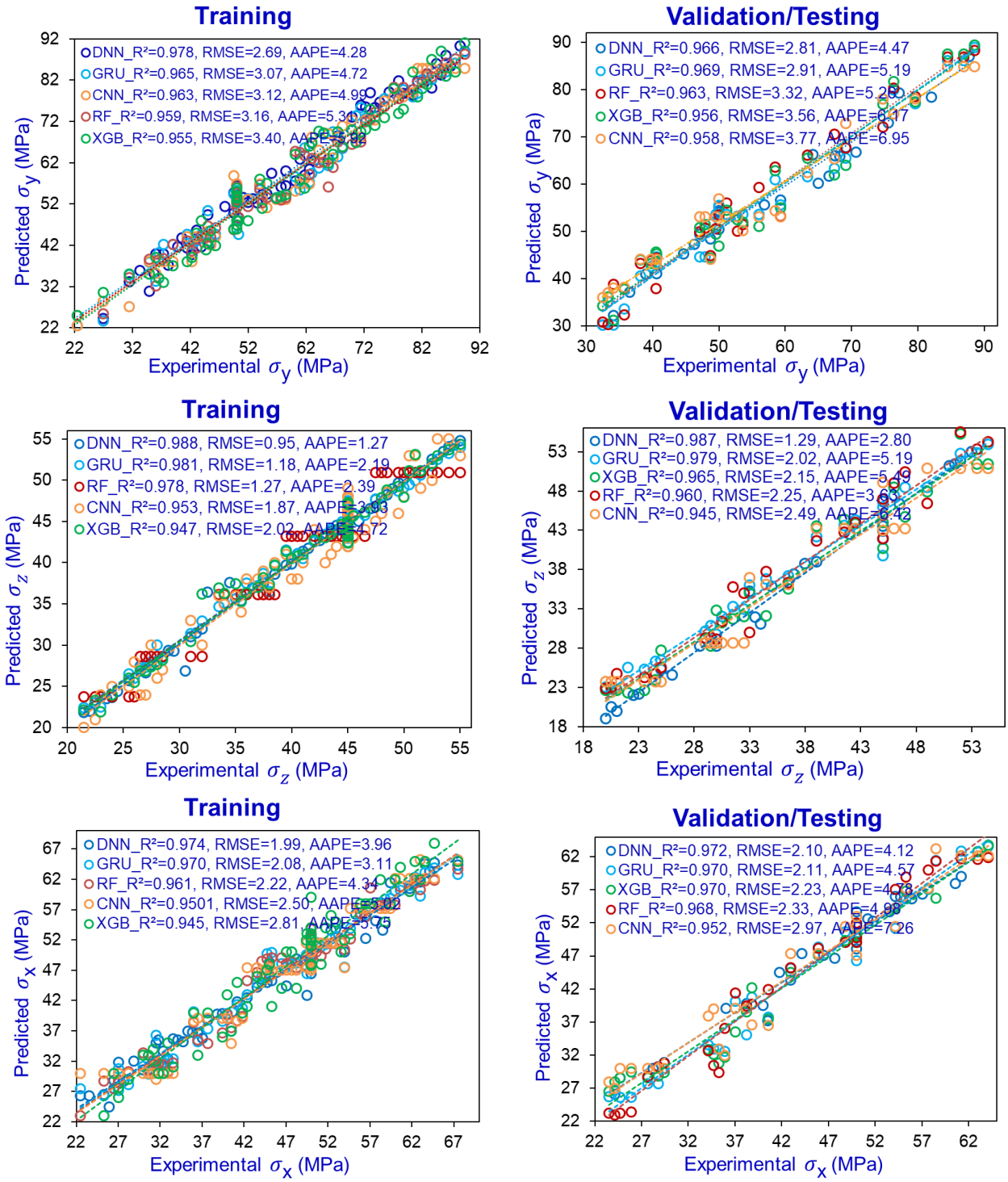


Figure H-3: Cross plots between predicted and experimental stresses for the proposed DNN, GRU, CNN, RF, XGB models of σ_z, σ_x, and σ_y, stresses for Core-D.

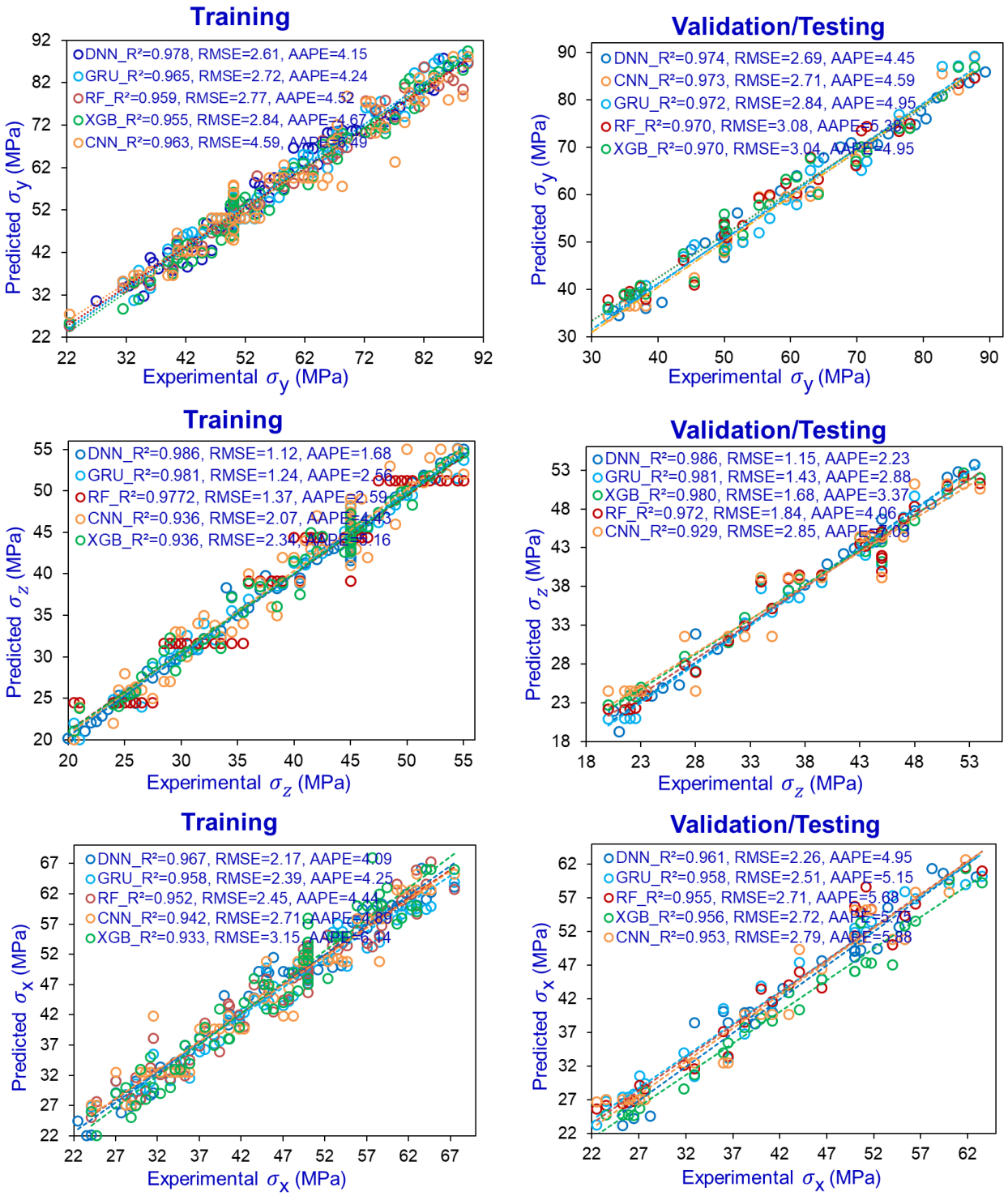


Figure H-4: Cross plots between predicted and experimental stresses for the proposed DNN, GRU, CNN, RF, XGB models of σ_z, σ_x, and σ_y, stresses for Core-E.

Appendix I: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_z, σ_x, σ_y stresses for Core-B, -C, -D, and -E.

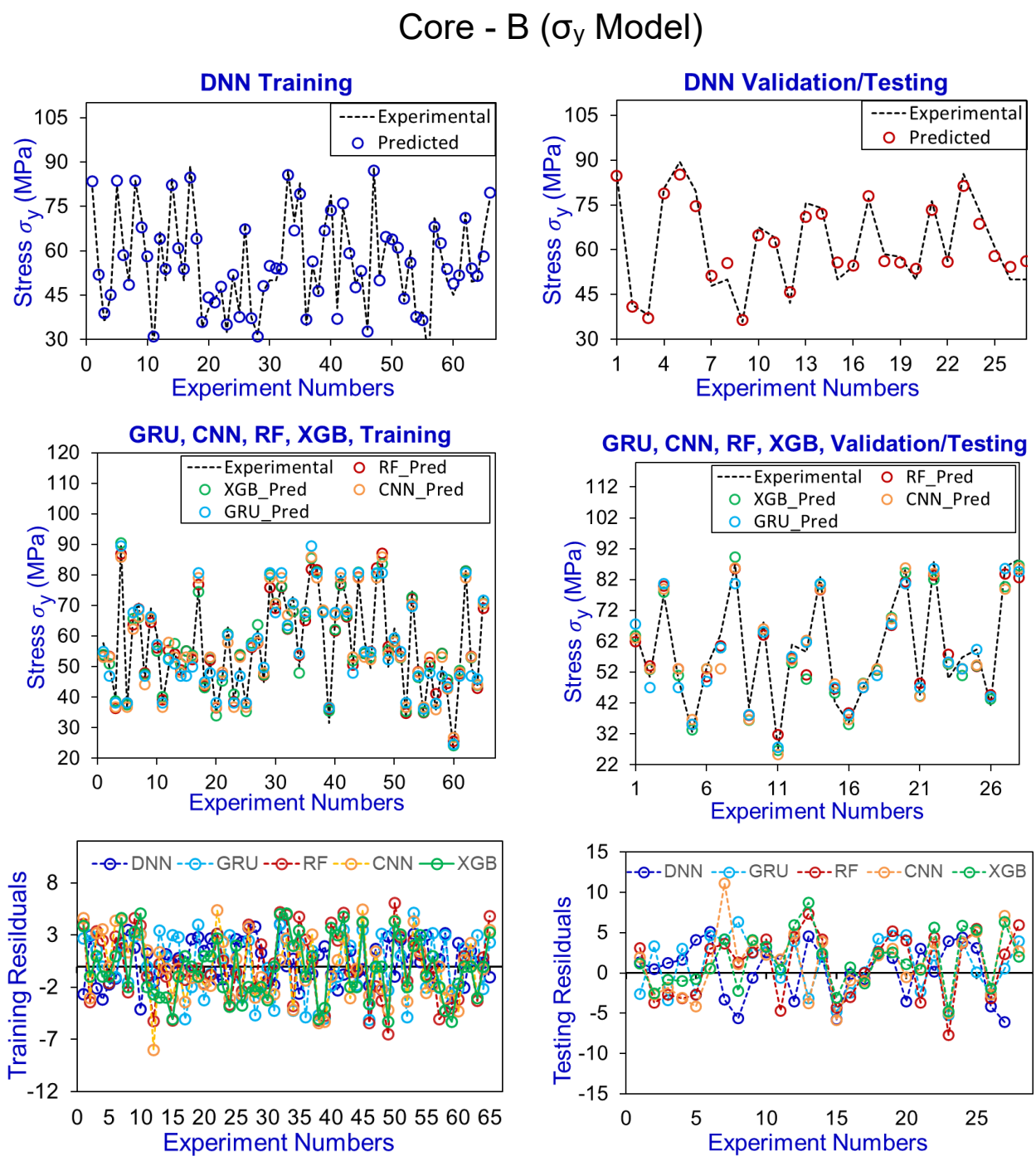


Figure I-1: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_y stress for Core-B.

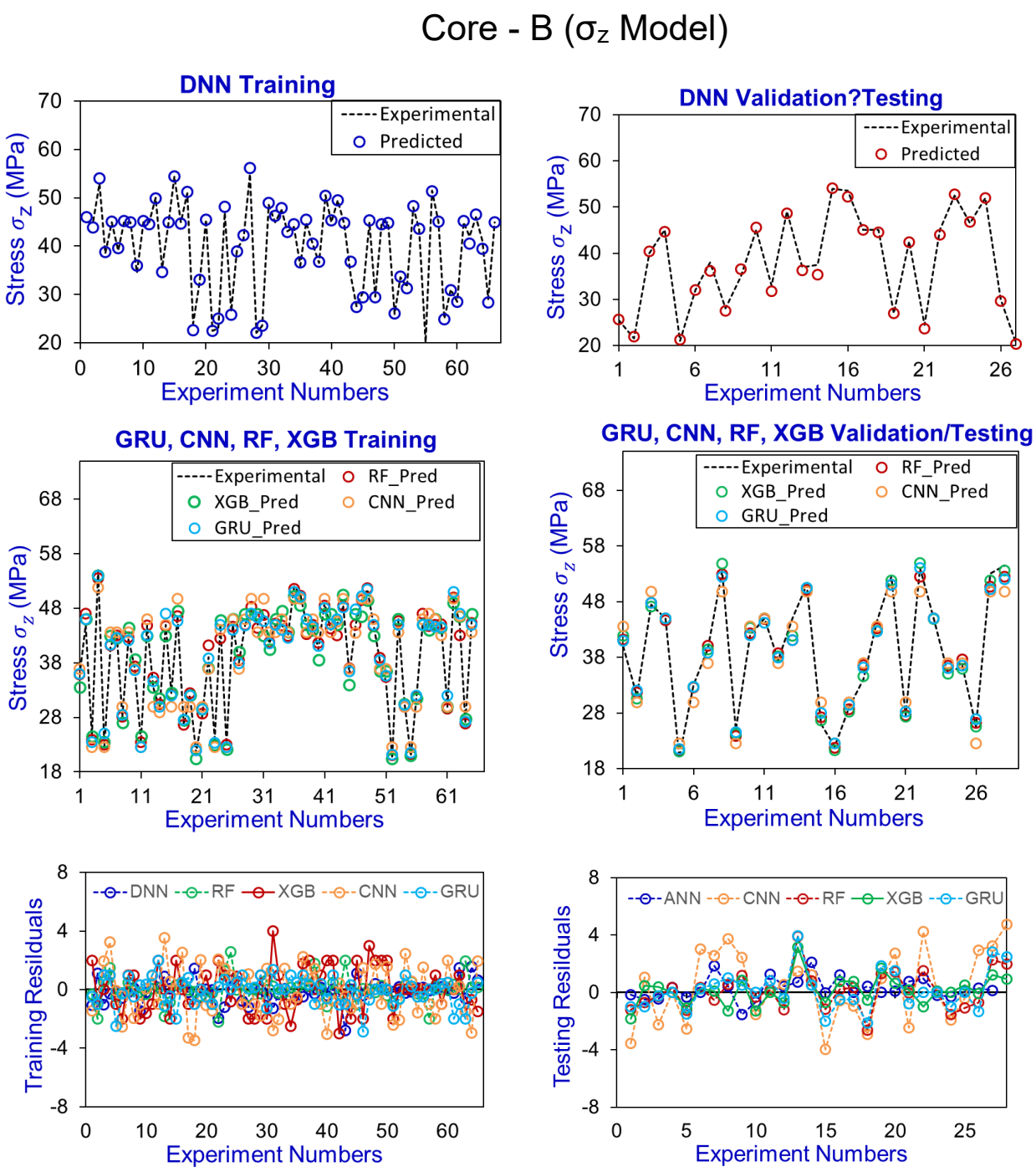


Figure I-2: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_z stress for Core-B.

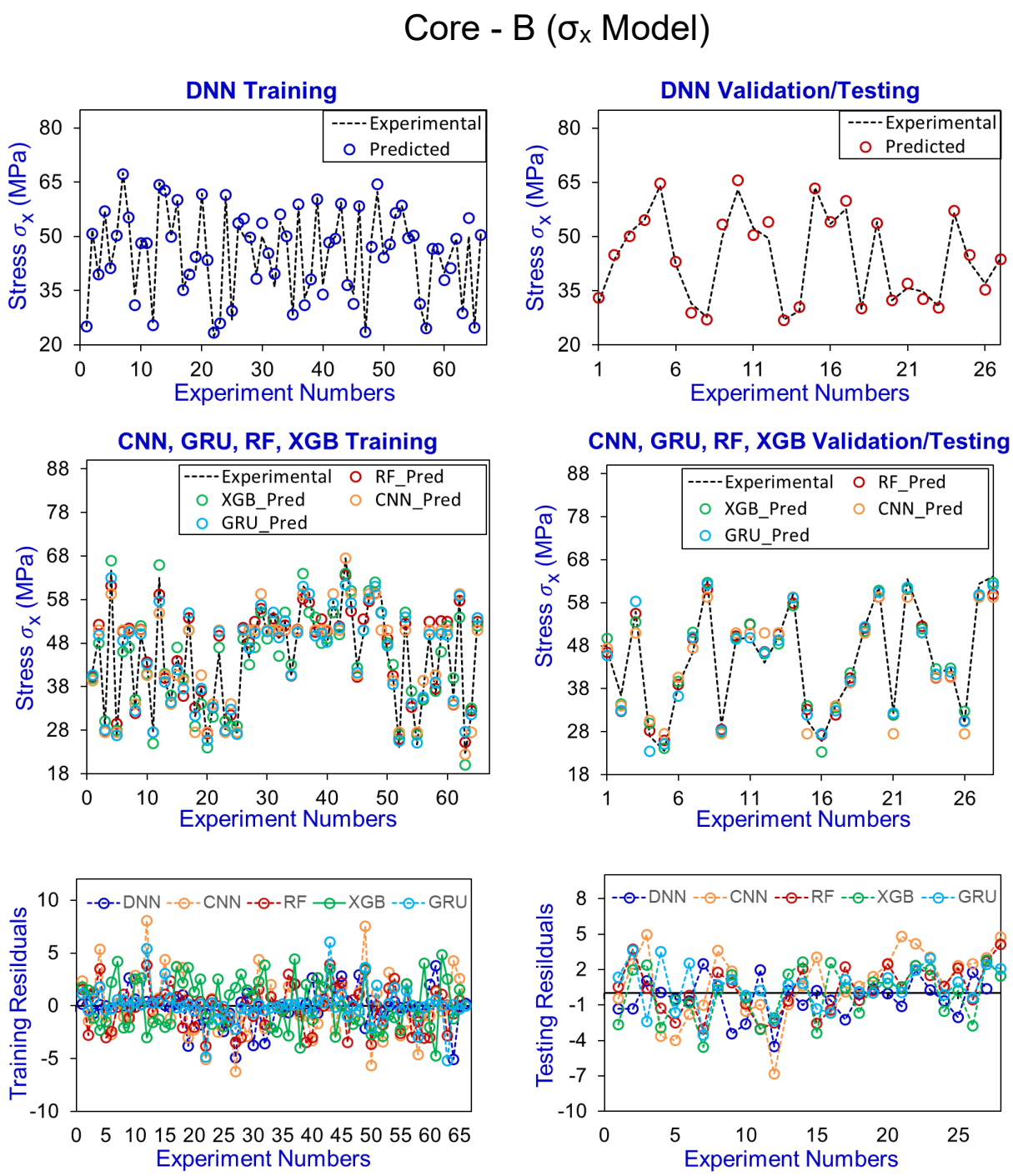


Figure I-3: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_x stress for Core-B.

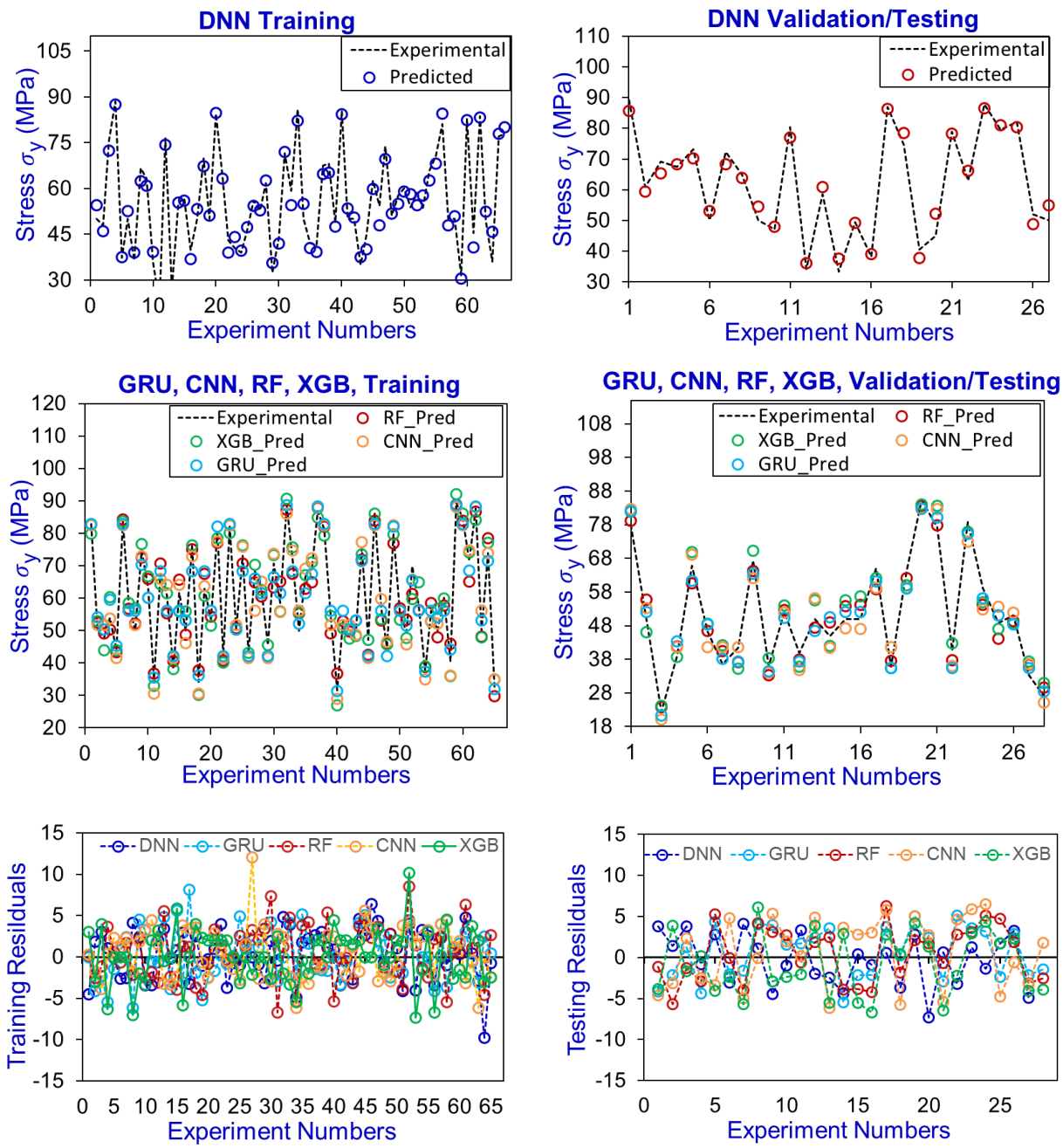


Figure I-4: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_y stress for Core-C.

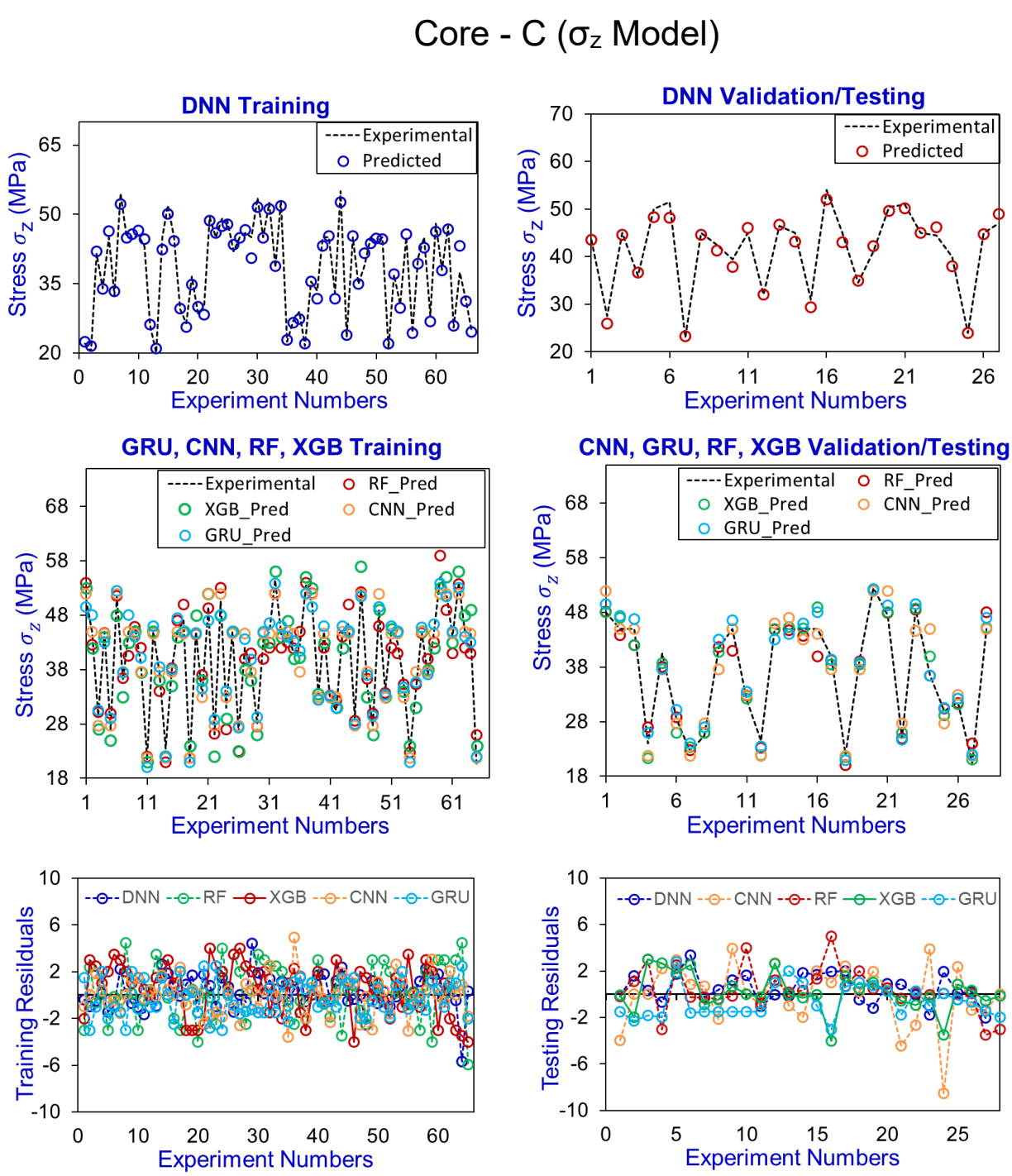


Figure I-5: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_z stress for Core-C.

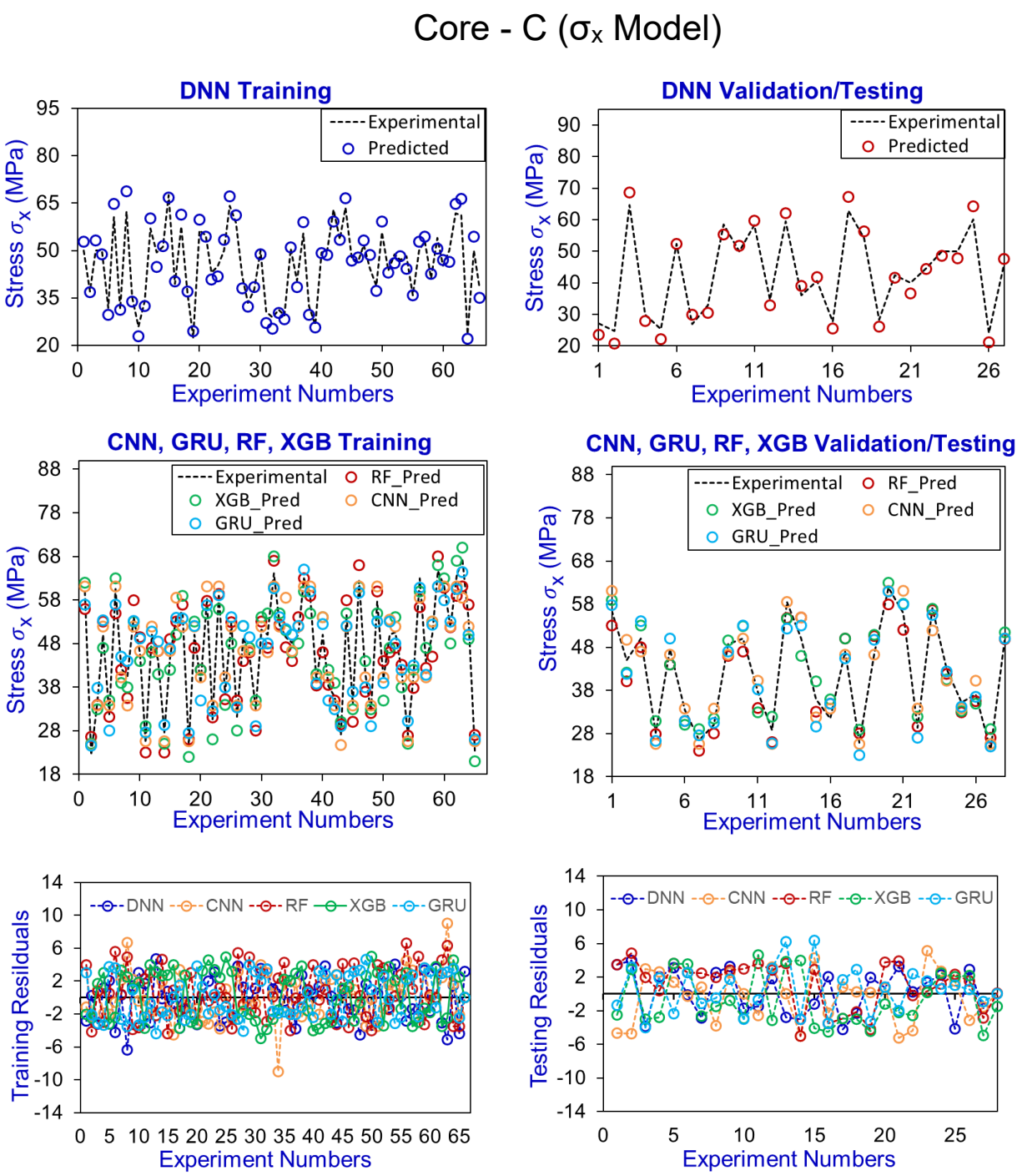


Figure I-6: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_x stress for Core-C.

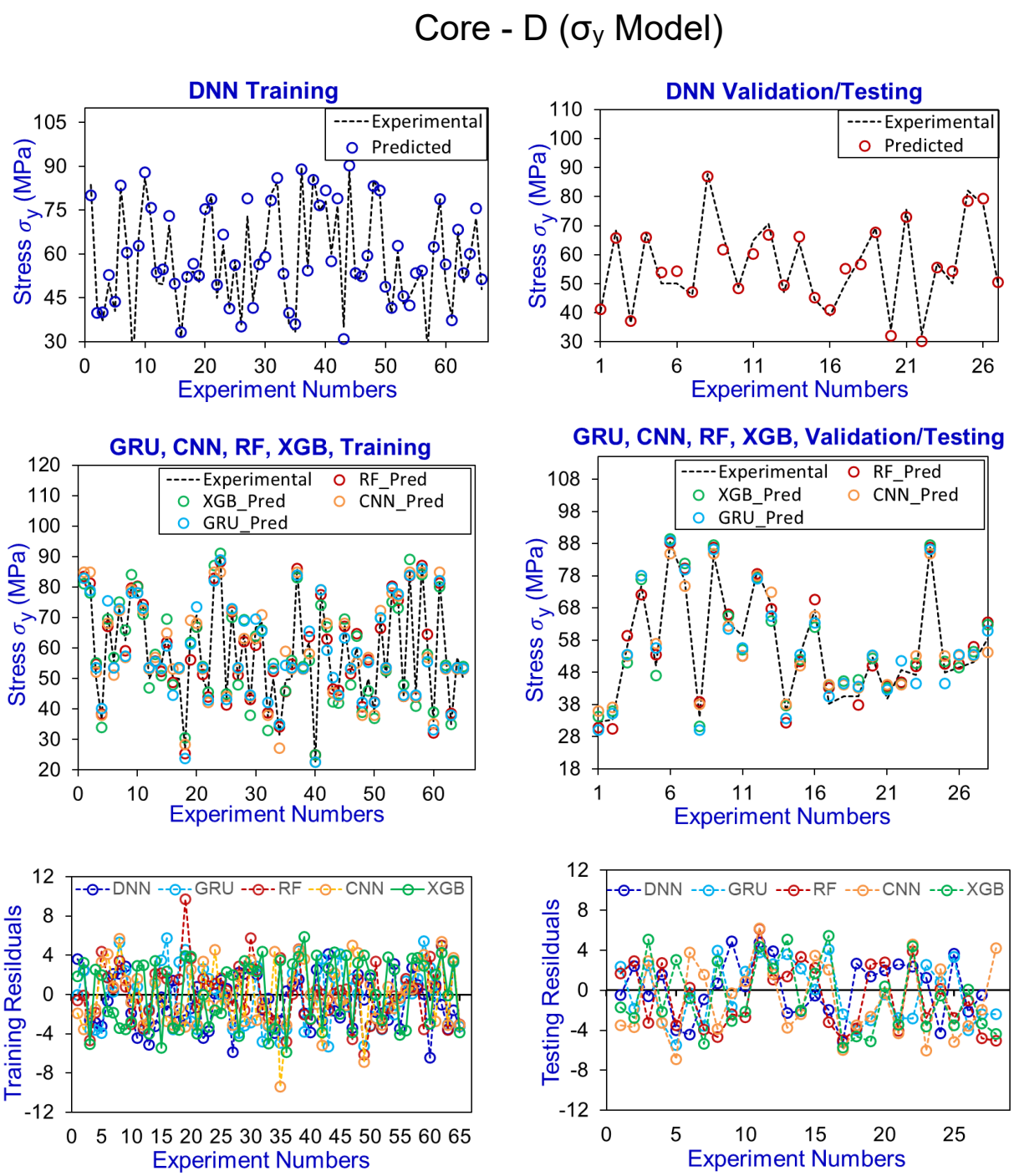


Figure I-7: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_y stress for Core-D.

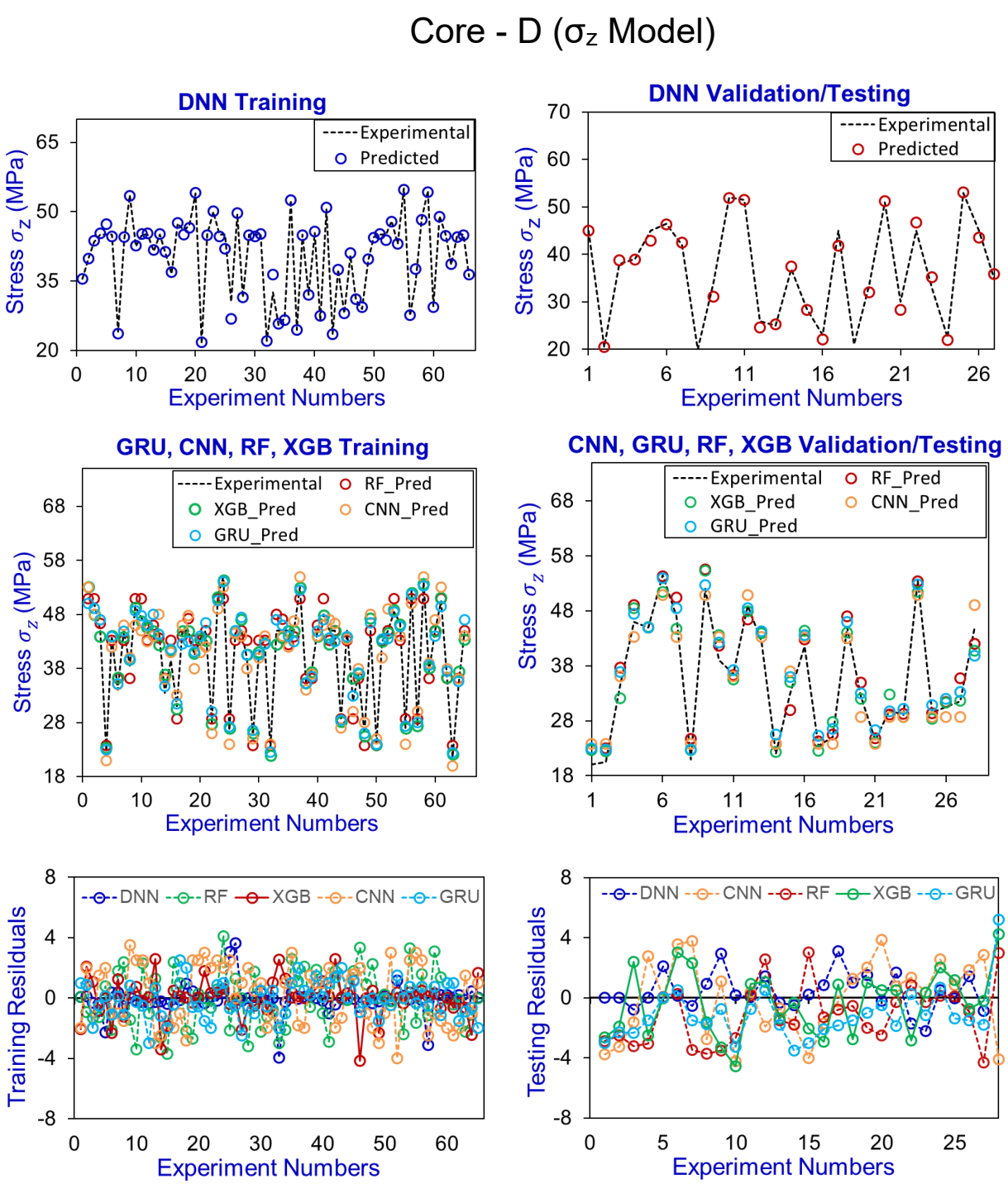


Figure I-8: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_z stress for Core-D.

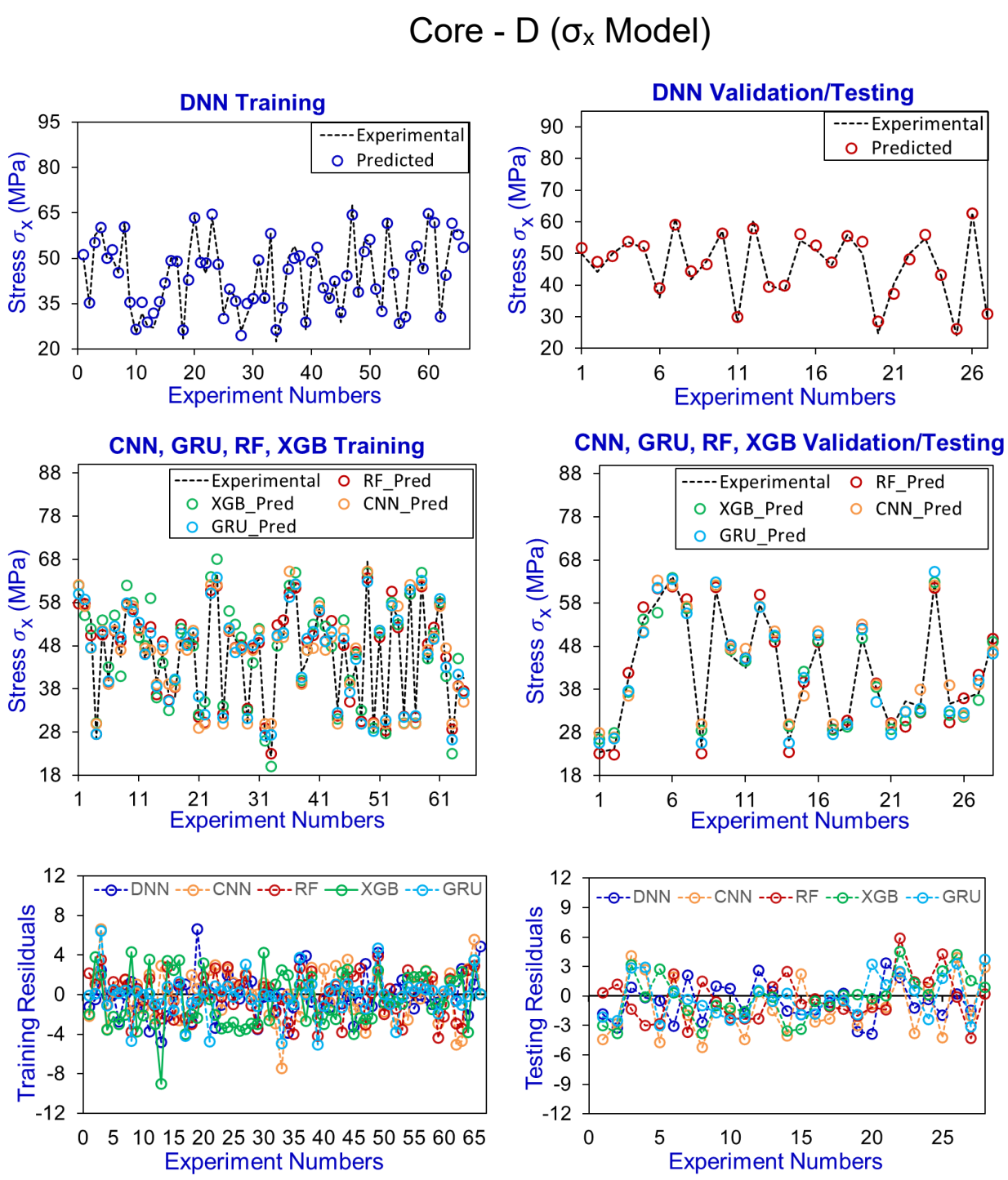


Figure I-9: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_x stress for Core-D.

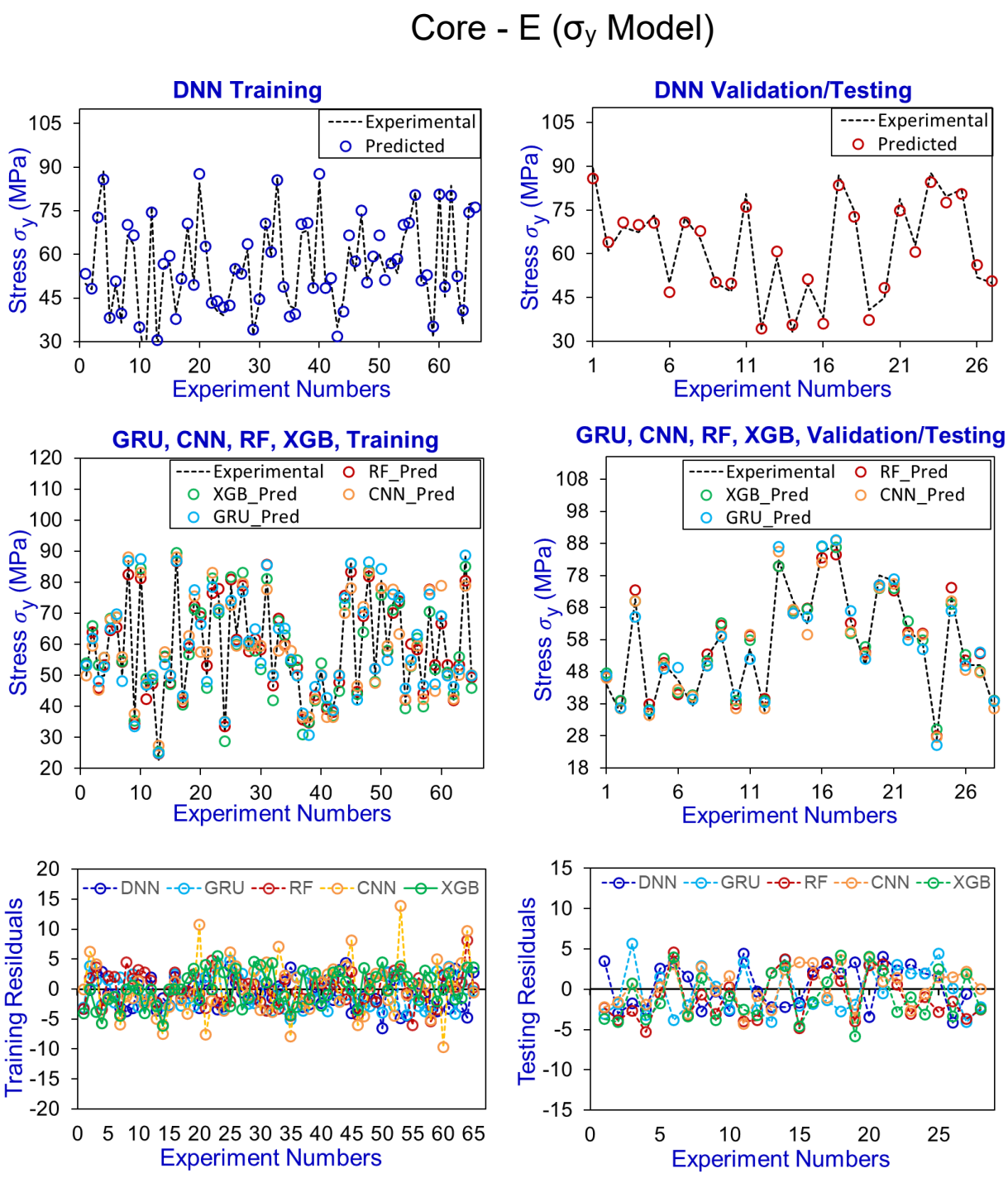


Figure I-10: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_y stress for Core-E.

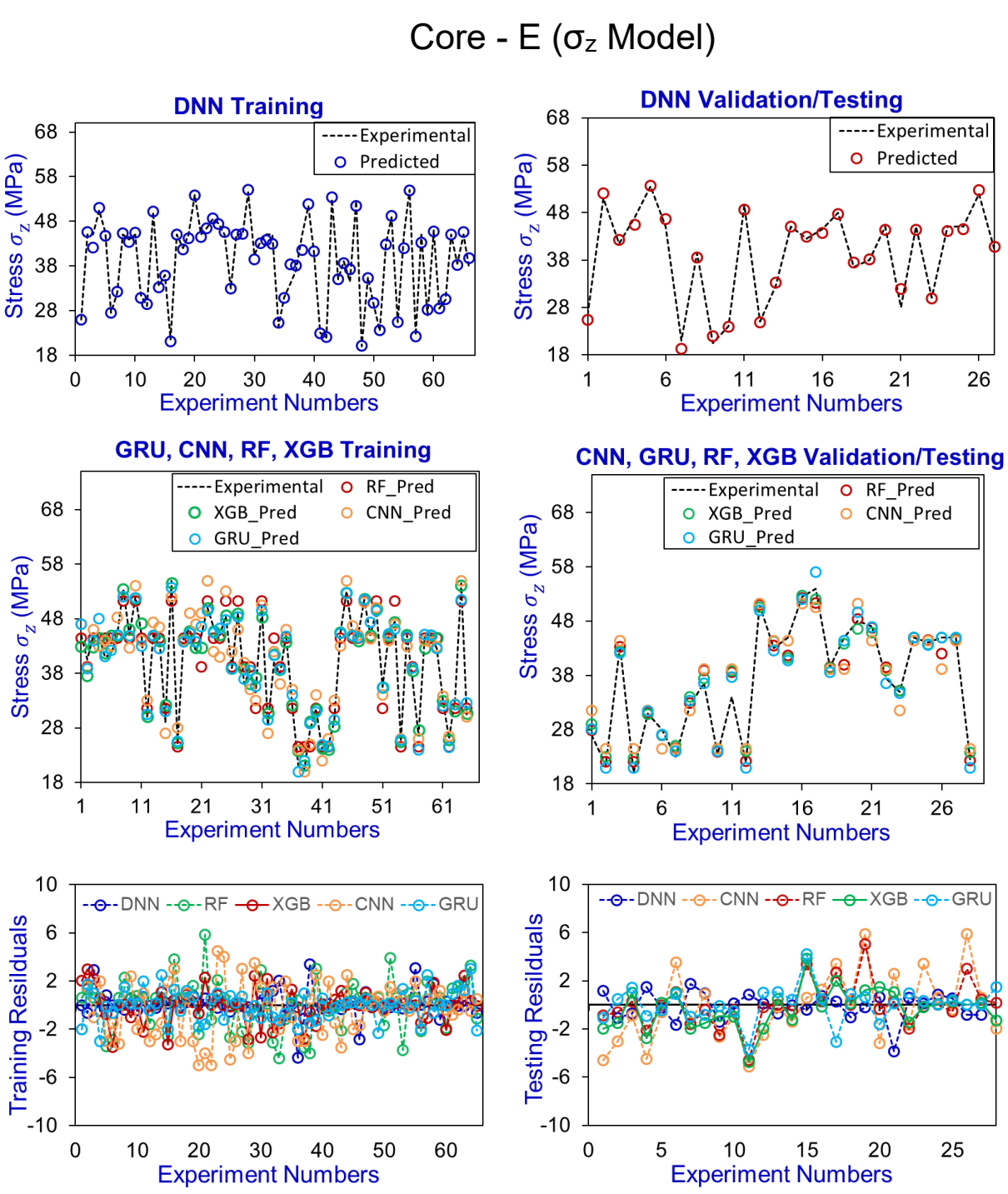


Figure I-11: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_z stress for Core-E.

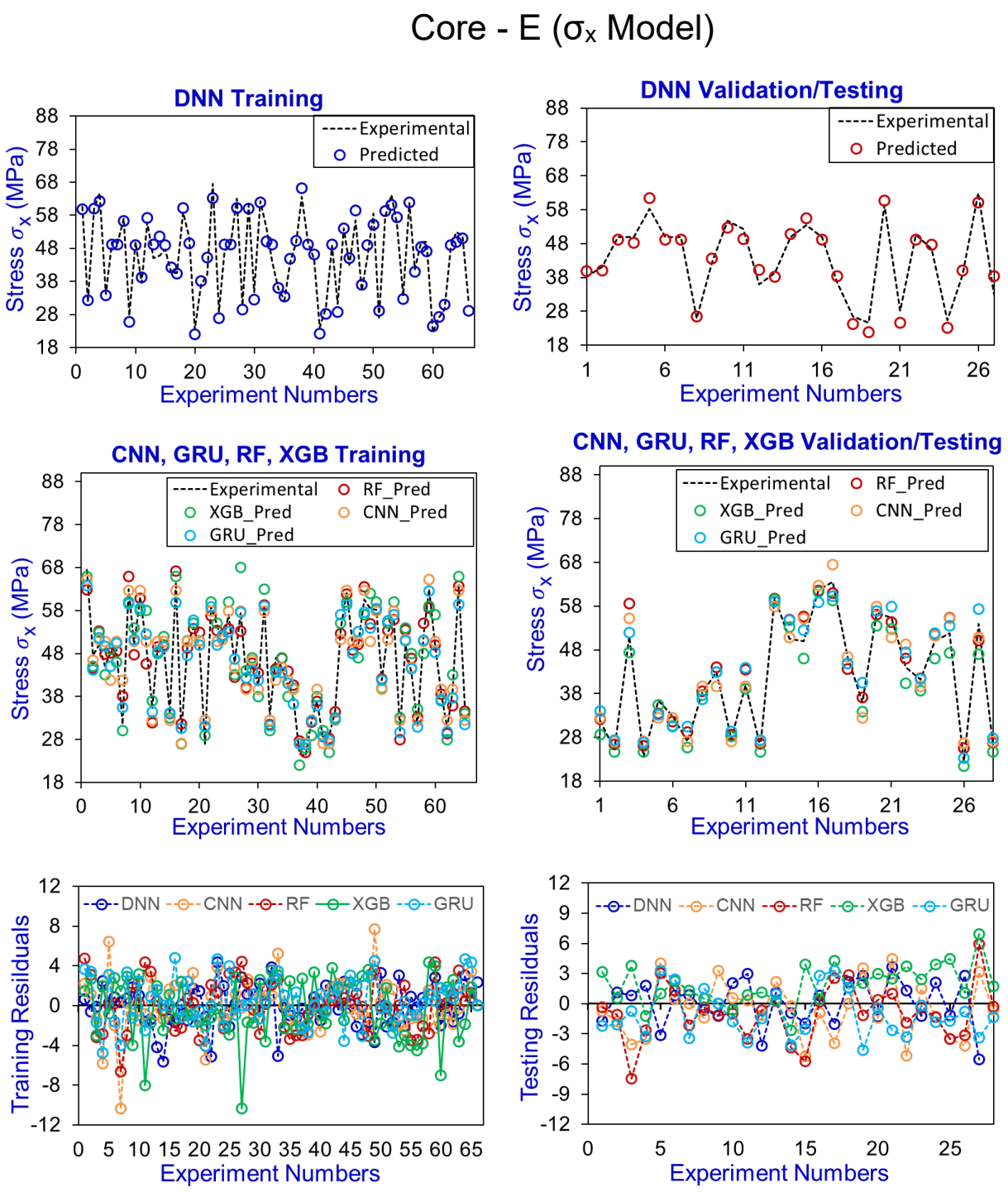


Figure I-12: Comparison between predicted and experimental stresses for the proposed DNN, RF, XGB, CNN, GRU models of σ_x stress for Core-E.

Appendix J: Residual errors for the training and testing/validation phases of the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses.

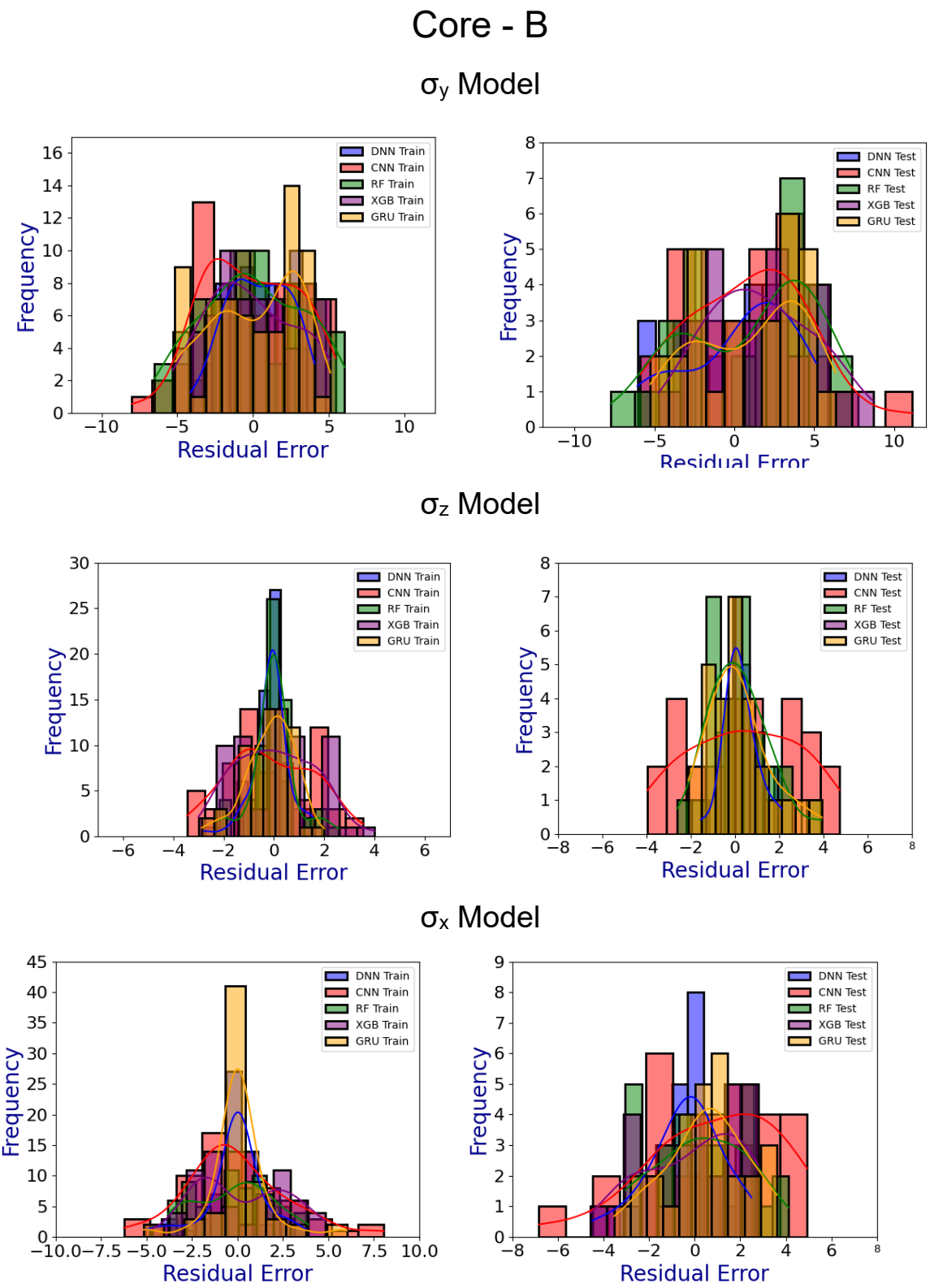


Figure J-1: Residual errors for the training and validation/testing phases of the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses Core-B.

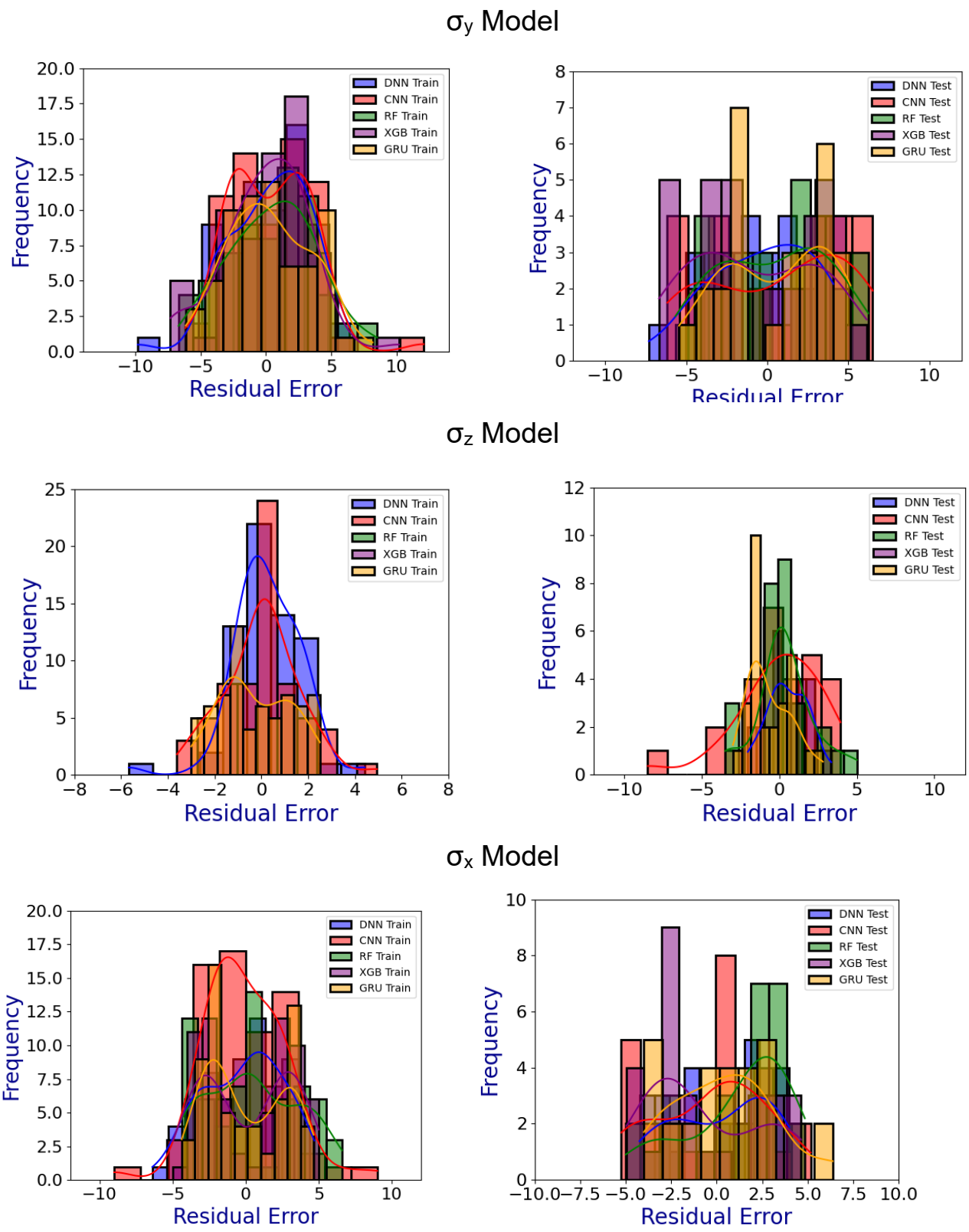
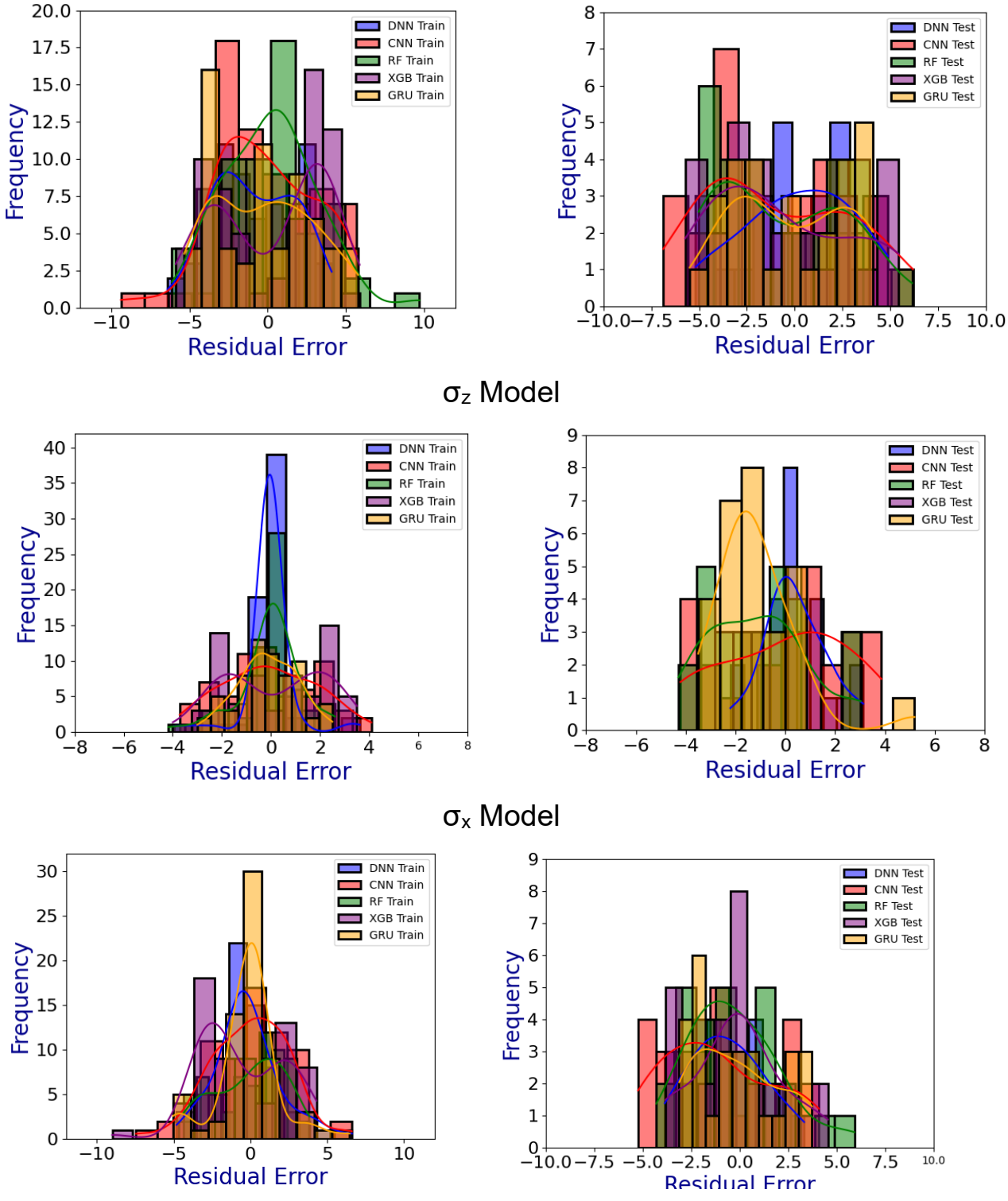


Figure J-2: Residual errors for the training and validation/testing phases of the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses Core-C.

Core - D

σ_y Model



σ_z Model

σ_x Model

Figure J-3: Residual errors for the training and validation/testing phases of the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses Core-D.

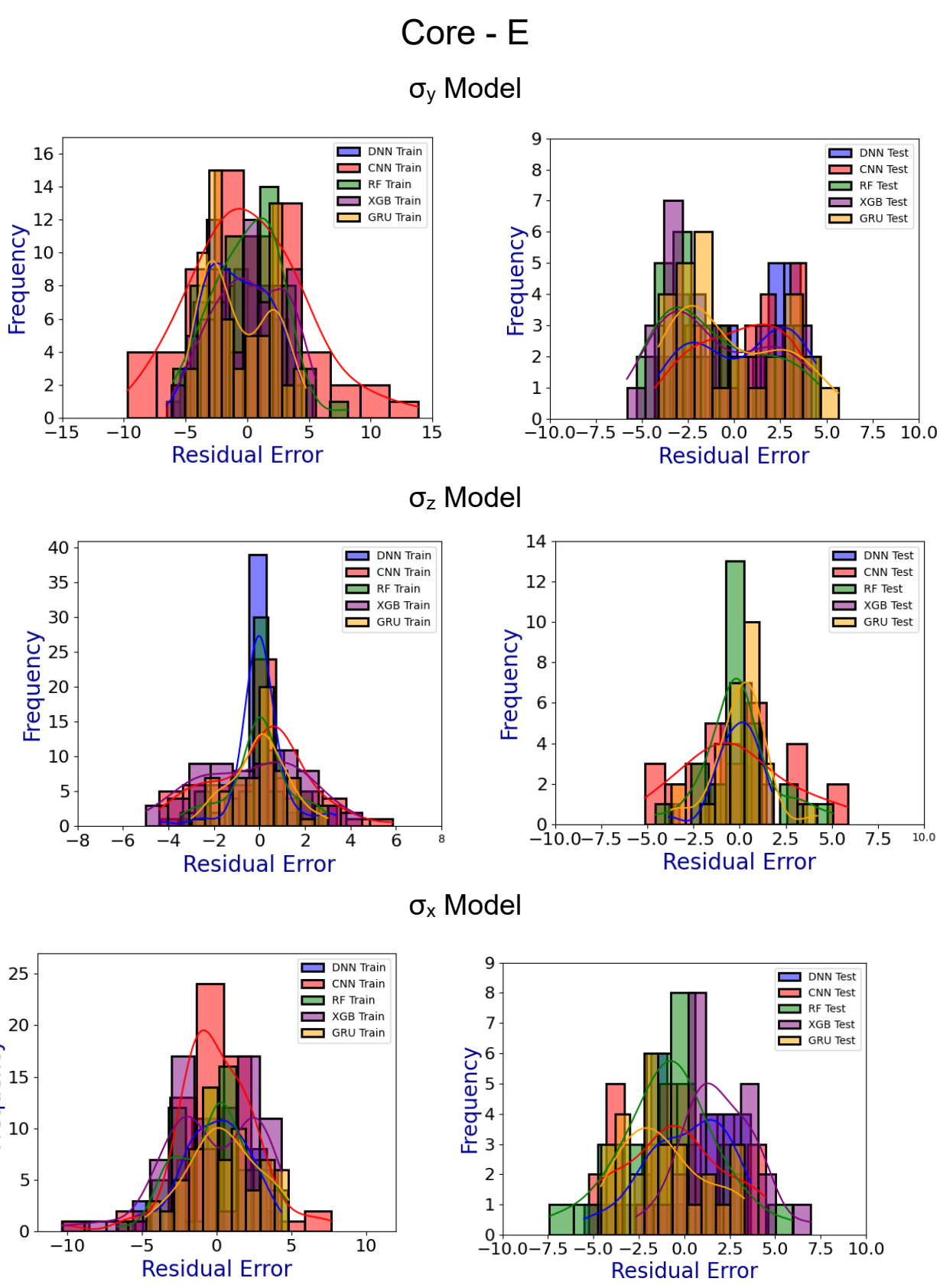


Figure J-4: Residual errors for the training and validation/testing phases of the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses Core-E.

Appendix K: Comparison of prediction RMSE and AAPE for the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses.

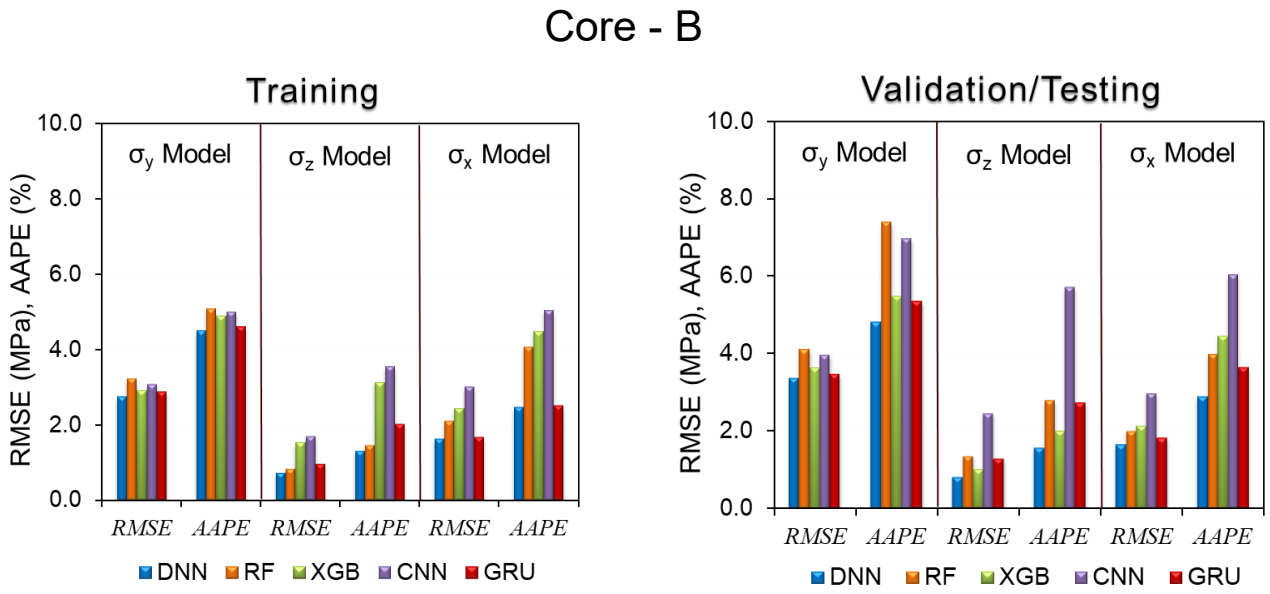


Figure K-1: Comparison of prediction RMSE and AAPE of the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses for Core-B.

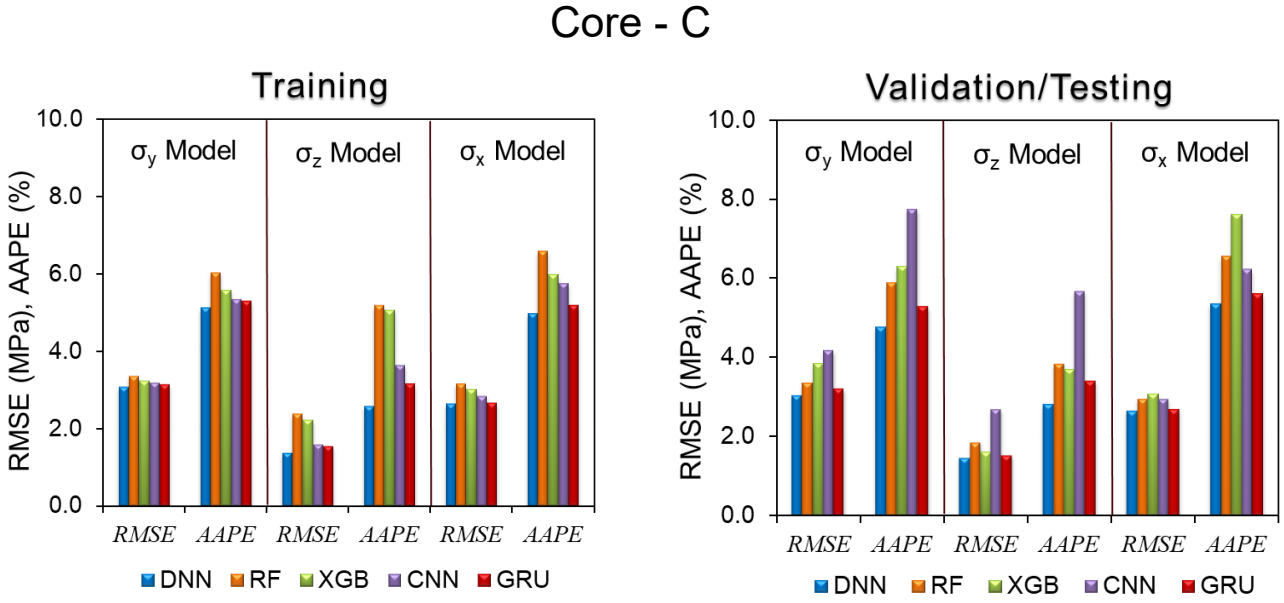


Figure K-2: Comparison of prediction RMSE and AAPE of the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses for Core-C.

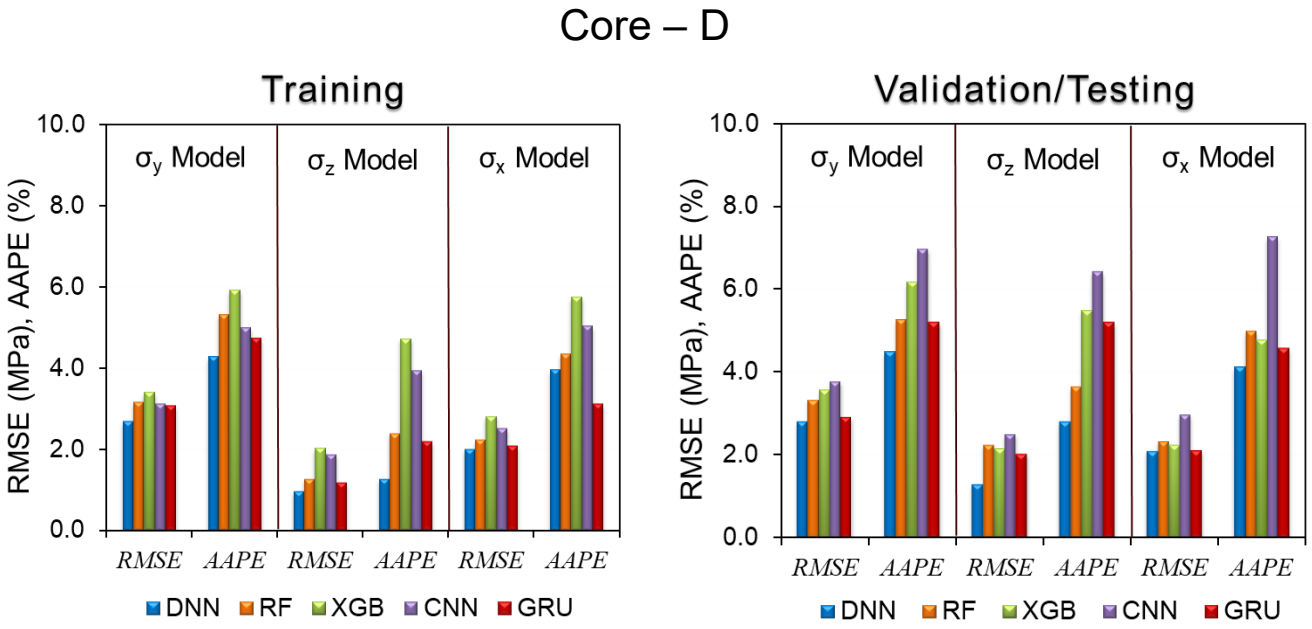


Figure K-3: Comparison of prediction RMSE and AAPE of the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses for Core-D.

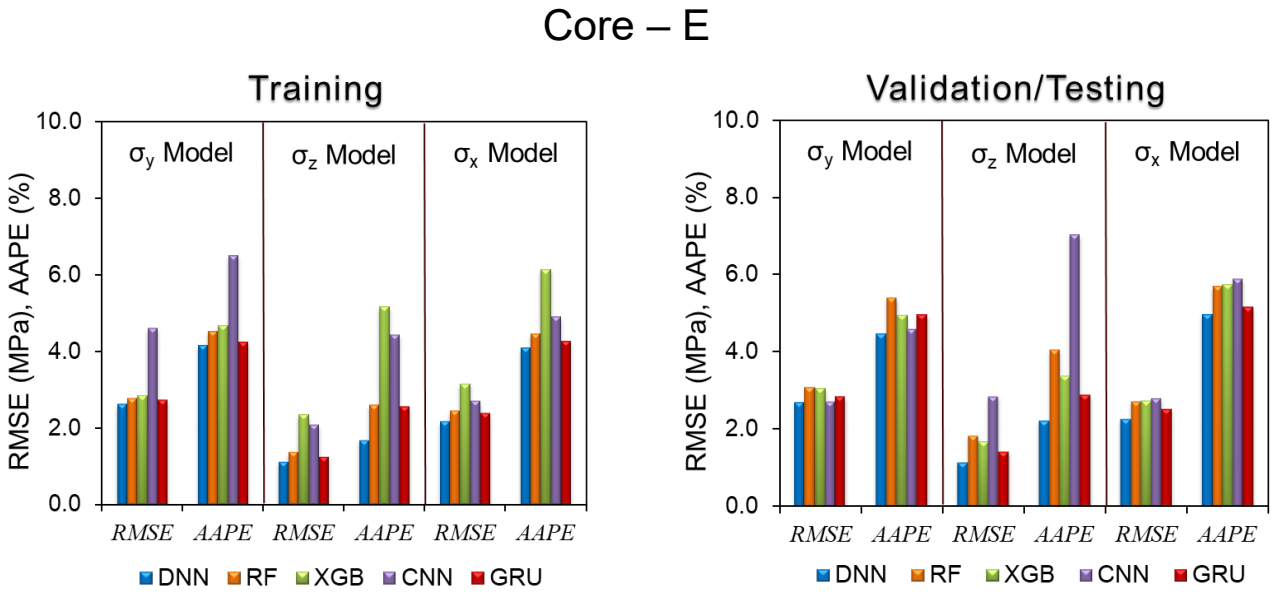


Figure K-4: Comparison of prediction RMSE and AAPE of the proposed DNN, RF, XGB, GRU, CNN models of σ_z, σ_x, σ_y stresses for Core-E.

Appendix L: Results of SHAP analysis of DNN models of σ_z and σ_x stresses.

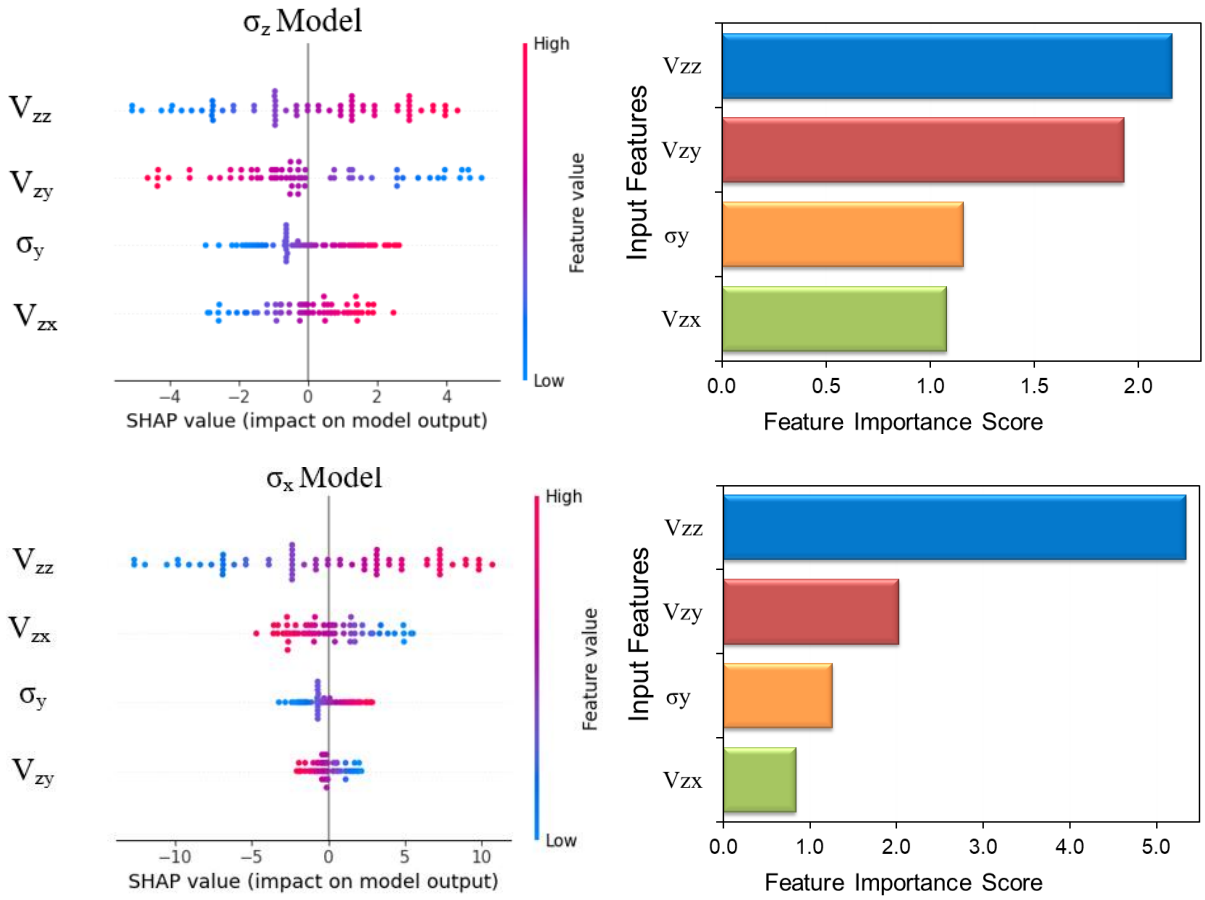


Figure L-1: SHAP summary plots (left column) and corresponding feature importance scores (right column) of σ_z and σ_x stress models.

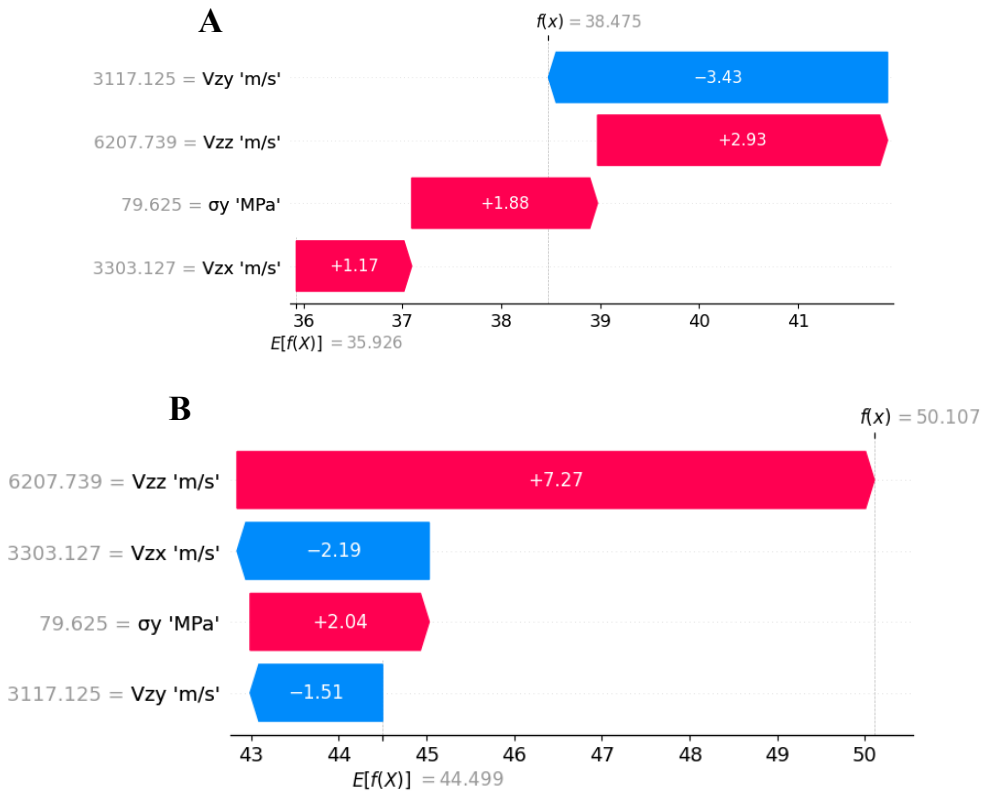


Figure L-2: SHAP waterfall plots for a specific instance representing impact of individual feature on prediction of stresses; (A) σ_z, (B) σ_x.

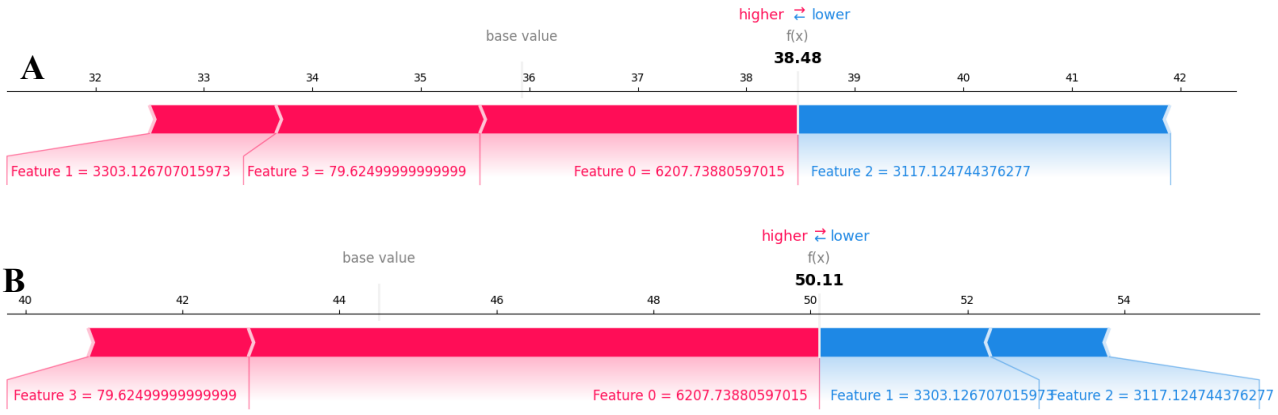


Figure L-3: SHAP force plots for a specific instance representing individual feature contribution to get the final prediction of stresses; (A) σ_z, and (B) σ_x.

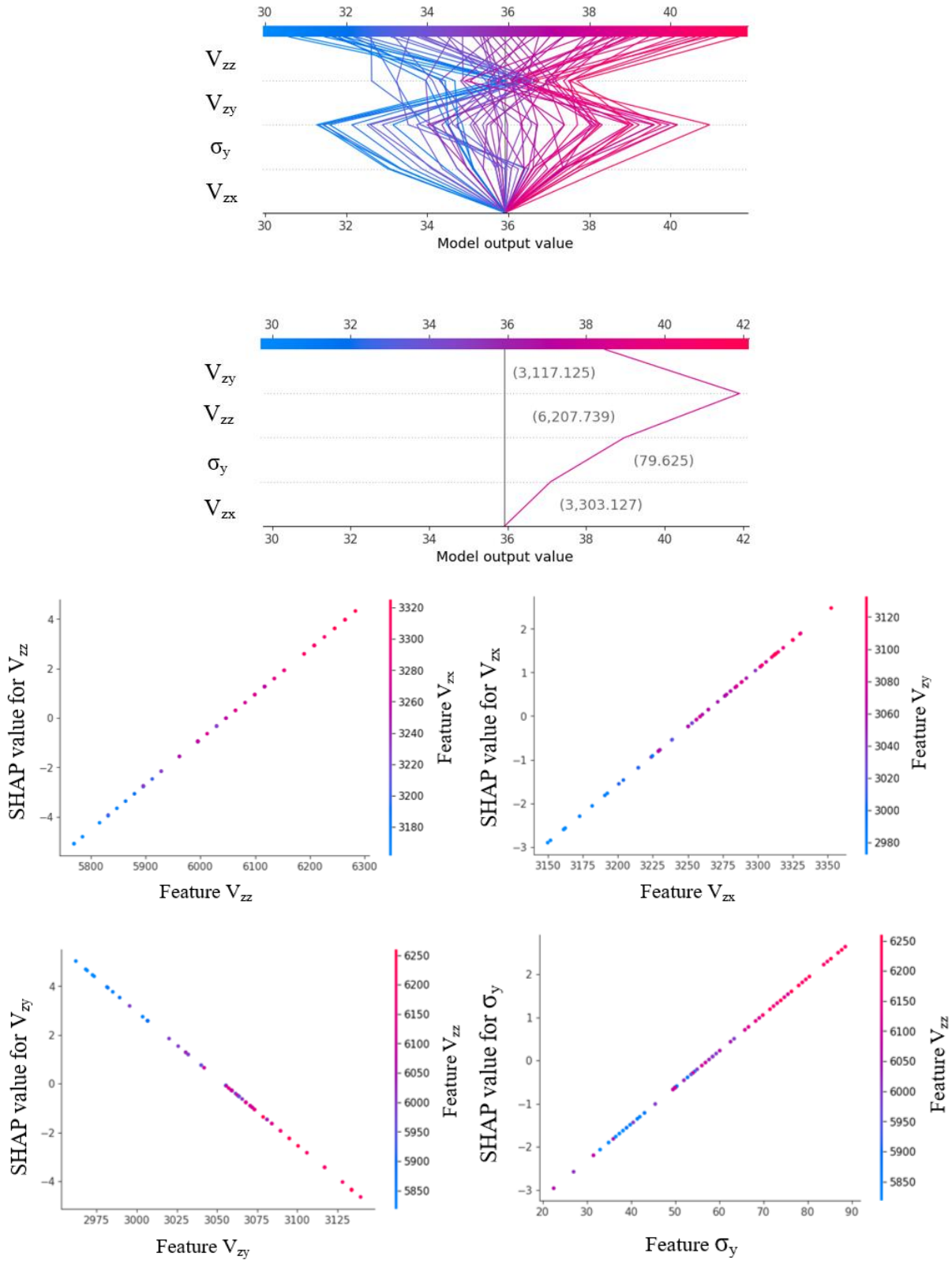


Figure L-4: SHAP decision and dependence plots for σ_z model; (A) decision plot for a few instances, (B) decision plot for a specific instance, (C) dependence plots for features V_{zz}, V_{zx}, and V_{zy}.

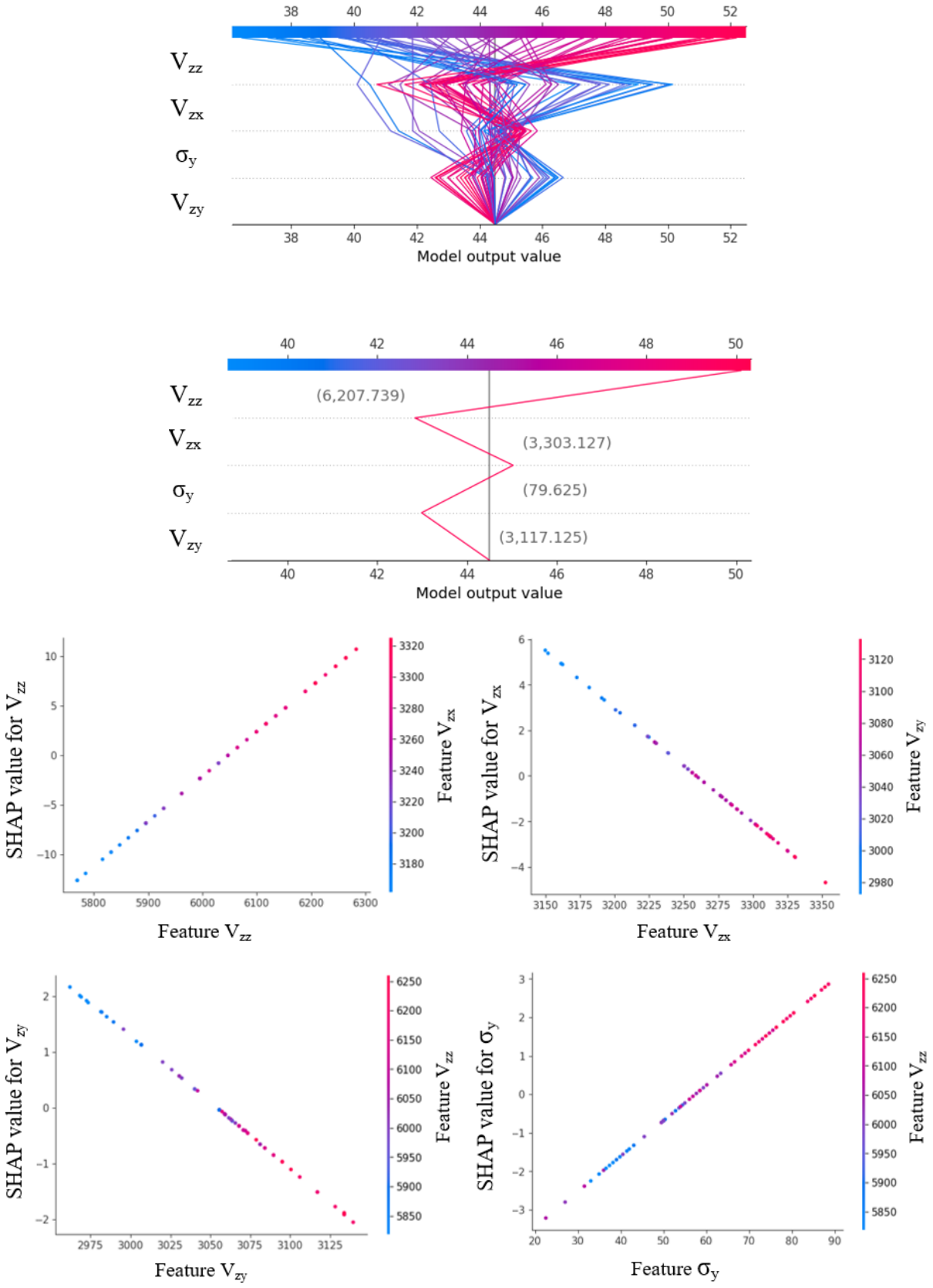


Figure L-5: SHAP decision and dependence plots for σ_x model; (A) decision plot for a few instances, (B) decision plot for a specific instance, (C) dependence plots for features V_{zz}, V_{zx}, and V_{zy}.