Title: Integrating Advanced Data Imputation Techniques and Machine Learning Models for Optimized Geopolymer Concrete Mix Design Prediction

1. Introduction

Geopolymer concrete (GPC) has emerged as a sustainable alternative to Portland cement (PC) concrete, driven by superior mechanical and durability properties [1]. Furthermore, it can reduce the carbon footprint by reducing CO_2 emissions during PC manufacturing and also by providing an added value to fly ash, providing an alternative to disposal in in landfills, in which up to 50% of fly ash can be disposed [2]. Unlike PC concrete, which relies on calcium silicate hydrates as a binding agent, GPC utilizes aluminosilicate sources, primarily fly ash, activated by alkaline activators, to form a polymeric network that binds the aggregates together [3].

Traditional methods for designing geopolymer concrete (GPC) mixes depend on empirical formulas, trial-and-error procedures, and extensive laboratory testing. Although these methods have provided acceptable results, they tend to be labor-intensive, time-consuming, and resource-demanding [4]. These limitations arise because traditional mix design approaches often fail to capture the intricate relationships and synergies between the various GPC components, such as fly ash, alkali activators, aggregates, and additional admixtures which collectively influence the fresh and hardened properties of the concrete [5]. For example, fly ash, a key ingredient in GPC, varies significantly in its chemical composition depending on its source, and this variation can alter the setting time, workability, and compressive strength of the final product [6]. Similarly, the ratio of alkali activators, including sodium or potassium hydroxide and silicate, can dramatically impact the reactivity of the geopolymer binder and, consequently, the mechanical properties and durability properties of the concrete [7]. These complex interactions between the GPC constituents cannot easily modeled or predicted using traditional empirical methods, leading to suboptimal designs that may not perform as expected

under varying environmental conditions or due to material availability [8]. Moreover, the variability in material properties such as the particle size distribution, surface area, amorphous content of the fly ash, and concentration of alkalis can further complicate the mix design process [9, 10]. In response to these challenges, researchers have increasingly turned to machine learning techniques to develop more efficient and accurate methods for designing GPC mixes. These data-driven models can account for the complex, nonlinear relationships between the input variables and the resulting concrete properties, offering the potential for more precise and optimized mix designs [11].

Designing an effective GPC mix using machine learning models is a challenging task due to the sensitivity of the GPC to both the chemical and physical characteristics of the components. The content of key chemical components such as SiO_2, Al_2O_3, Fe_2O_3, and CaO influence the reactivity and strength development of the geopolymer matrix [12]. Similarly, physical attributes including Brunauer-Emmett-Teller (BET) surface area, amorphous content, and particle size distribution significantly impact the workability, setting time, and mechanical performance of GPC. Variations in these properties can lead to substantial changes in the overall behavior of the concrete, making the mix design process highly intricate.

In addition, to this complexity, developing accurate predictive models for GPC mix design is also hindered by the lack of comprehensive datasets. While data on major oxides is generally available, more detailed physical properties data, such as BET surface area, particle size and amorphous content, are often missing. For instance, Junninen, et al. [13] emphasize that the absence of critical parameters can impede the creation of reliable predictive models, limiting the understanding of how these properties affect concrete performance.

The absence of details in GPC datasets is a significant challenge, as machine learning models depend on complete data to effectively capture the non-linear and complex relationships between input variables and output properties like compressive strength and durability. Missing

data can arise due to various factors such as measurement errors, the unavailability of testing equipment, or high testing costs. These missing data points can severely bias predictions and reduce the generalizability of the models, particularly in cases where the missing values affect critical nonlinear relationships between input and output variables [14].

One of the common approaches to deal with the missing values is to perform data imputation. However, the imputation procedure can be affected by the data missing mechanism. Missing data can typically be categorized into three mechanisms [15]. One of the mechanisms is missing completely at random (MCAR) where the data is missing independent of both observed and unobserved values. The probability of a value being missing is uniform across the dataset [16]. MCAR satisfy Eq. 1. In the equation, R is the binary matrix indicating which values are missing, Y is the underlying complete data, and ψ are the parameters of the missing data model.

$$\Pr(R = 0|Y, \varphi) = \Pr(R = 0|\varphi) \quad \text{Eq. 1}$$

The second mechanism is missing at random (MAR) where the absence of data is related to observed variables but not to the missing values themselves. In MAR the probability of missing data is constant only within specific observed groups (e.g., specific values of a feature). In other words, the missing values are related to some part of the underlying values. MAR obeys Eq. 2, where Y_{Obs} is the observed fraction of data.

$$\Pr(R = 0|Y, \varphi) = \Pr(R = 0|Y_{Obs}, \varphi) \quad \text{Eq. 2}$$

Finally, if neither the MCAR nor the MAR assumptions hold, the mechanism is in the missing not at random (MNAR) class. In other words, the etiology of missing data is unknown and, therefore, may depend on the unobserved data. MNAR can have two different settings: dependance on the missing values themselves or on other features that are outside of the dataset. Several studies have critically examined the performance of different imputation methods based on the missing data categories. Lyngdoh, et al. [17] employed MICE imputation in comparison to mean, median and KNN data imputation for missing data in ordinary Portland

cement concrete mix design and found that MICE preserved the dataset's structure and variability. However, Pereira, et al. [18] demonstrated that KNN imputation outperformed simple imputation methods such as mean/median in datasets with complex interactions, but it struggled with computational efficiency in high-dimensional data. Furthermore, the study noted that KNN method, MICE method and Soft Imputation method can effectively handle missing data, even with MNAR and a missing rate up to 80% without compromising the quality of the data. ANN imputation, on the other hand, has gained popularity due to its ability to handle complex, nonlinear patterns, though its requirement for large datasets and the risk of overfitting in small or sparse datasets remains a challenge [19]. Each imputation method has its strengths and weaknesses, and the choice of an appropriate method depends on the category of the missing data, the complexity of the dataset, and the need for computational efficiency.

K-Nearest Neighbors (KNN) Imputation technique fills in missing values by averaging the K nearest neighbors, based on a distance metric such as Euclidean distance. While KNN can handle complex relationships between variables, it can be computationally expensive and may struggle with large datasets [10]. Multiple Imputation by Chained Equations (MICE) method, the missing values 'fills in' (imputes) in the dataset were filled through an iterative series of predictive models. In each iteration, each specified variable in the dataset is imputed using the other variables in the dataset. These iterations should be run until it appears that convergence has been met (tolerance). The soft imputation (SoftImp) approach leverages the relationships between observed and missing data to generate estimates for missing values, thus preserving the overall structure of the dataset. The process begins with an initial estimation of missing values, often using simple methods like mean or median imputation. Subsequently, an iterative algorithm refines these estimates by applying low-rank matrix approximation, commonly through singular value decomposition (SVD). Artificial Neural Network (ANN) imputation models the relationships in the data and can capture complex, nonlinear interactions.

The aim of this study is to develop a predictive model for geopolymer concrete mix designs that can accurately estimate the optimal mix proportions based on the chemical and physical properties of fly ash. This is achieved by addressing the challenges of missing data by implementing and comparing multiple data imputation techniques. Additionally, it seeks to evaluate the performance of various machine learning algorithms in predicting the mix design. This research contributes to the field of geopolymer concrete by introducing a novel approach that integrates advanced data imputation techniques with machine learning models to predict mix designs by incorporating physical and chemical properties of fly ash. The study not only addresses the issue of missing data but also optimizes model performance through extensive hyperparameter tuning.

2. Methodology

The flowchart illustrates the methodology adopted in this study. The process begins with data collection, followed by data preprocessing and feature selection as depicted in Figure 1. Next, the selected input variables were examined for missing values, with data imputation applied to address these gaps. Each imputation technique was fine-tuned through hyperparameter adjustment and subjected to a statistical evaluation to ensure the quality of the imputed data. Furthermore, the parameter correlation will be analyzed before and after the data imputation to ensure the reliability of the imputed data.

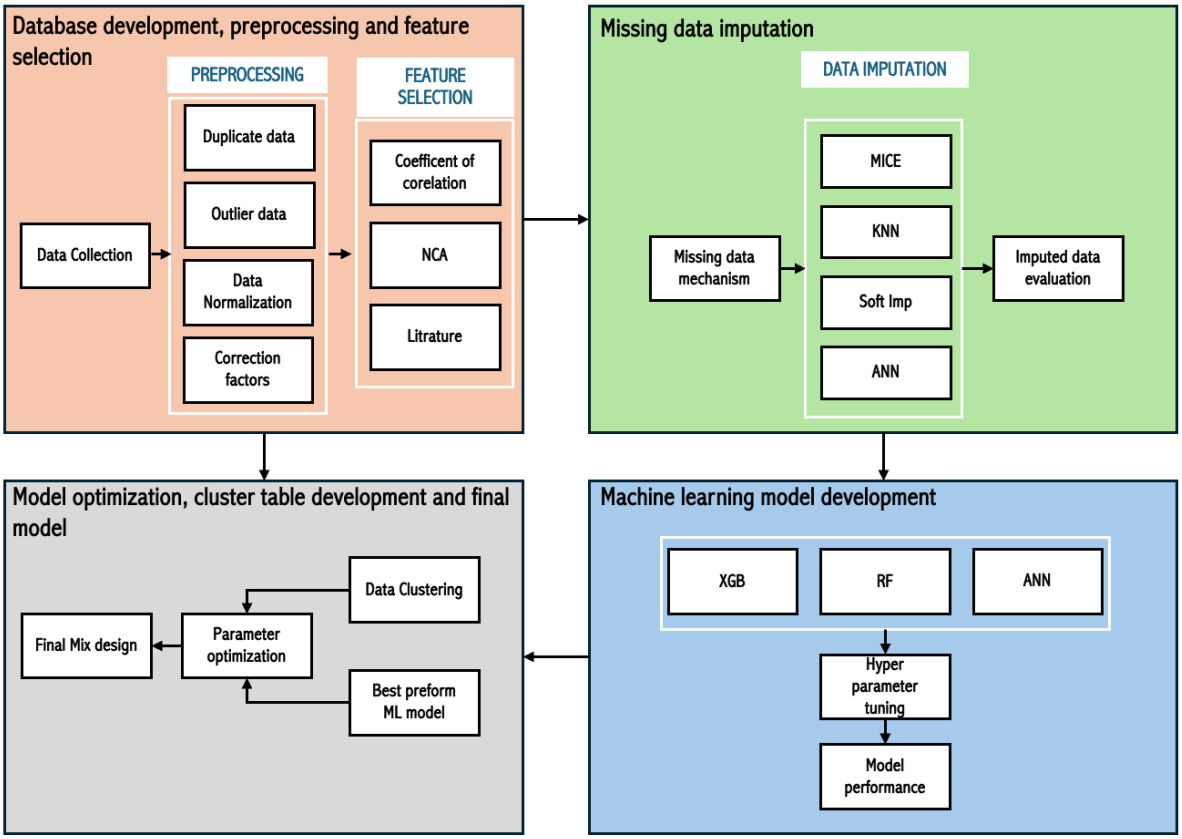


Figure 1 Overview of the methodology

The optimized dataset from each imputation method was then used to develop several machine learning models, with the resulting outputs were compared to assess the accuracy of the datasets. The imputation technique and machine learning model that performed best was adopted for use in the final mix design development with extensive hyper parameter optimization.

2.1. Database development, data pre-processing, and feature selection.

The data set used in the research was sourced from peer-reviewed literature articles published from 1990 to 2022 on low-calcium fly-based geopolymer concrete. Input variables affecting the compressive strength of geopolymer concrete were selected based on neighborhood component analysis and Pearson correlation coefficients. Duplicate data detection and removal, outlier detection and treatment, and data normalization were carried out as preprocessing steps to obtain a more stable and representative data set. Furthermore, a few additional steps were

taken to ensure the compressive strength data did not depend on the shape or size of the specimen. Moreover, data with 7-day compressive strength were converted into 28-day compressive strength using a regression model. More details on database development, feature selection, and data preprocessing can be found in the paper xxxx.

Table 1 delineates the input parameters and their summary statistics. Furthermore, Figure 2 shows the relationship between parameters, offering a visual depiction of these input parameters. Accordingly, three input variables, representing physical and mineralogical properties of fly ash, have missing values.

Table 1 Summary statistics of the input parameters

Input parameters		count	mean	std	min	25%	50%	75%	max	missing
Chemical Oxide of fly ash	SiO_2	226	226.0	36.3	177.7	196.3	210.1	238.7	340.4	0
	Al_2O_3	226	110.6	16.46	43.33	99.55	108.4	122.4	158.3	0
	Fe_2O_3	226	42.6	22.0	5.4	19.2	51.3	60.4	70.9	0
	CaO	226	10.7	6.2	0.8	7.1	9.7	10.9	30.9	0
Physical and mineralogical properties of fly ash	45 microns passing %	122	0.8	0.1	0.7	0.7	0.8	0.8	0.9	104
	BET surface area	62	1528	1085	310	773	1095	1876	5095	164
	Amorphous %	50	66.9	6.5	55.4	60.0	66.3	71.8	79.9	176
Na_2SiO_3 Solids	SiO_2 %	226	42.9	17.9	14.1	30.2	34.6	53.7	107.3	0
	Na_2O %	226	21.2	9.2	7.0	15.1	17.0	27.2	53.6	0
Solid_(NaOH)_weight		226	22.0	7.8	10.1	16.5	19.4	25.1	48.4	0
Total water		226	121.4	36.2	82.0	97.2	106.5	131.9	231.6	0
Time		226	28.5	13.2	12.0	24.0	24.0	24.0	96.0	0
Temp		226	69.9	11.7	60.0	60.0	61.5	80.0	105.0	0
Strength		226	40.6	14.2	10.4	31.0	39.6	50.9	76.4	0

Based on future selection and literature review, 13 input parameters were selected as the influential parameters for the compressive strength model prediction.

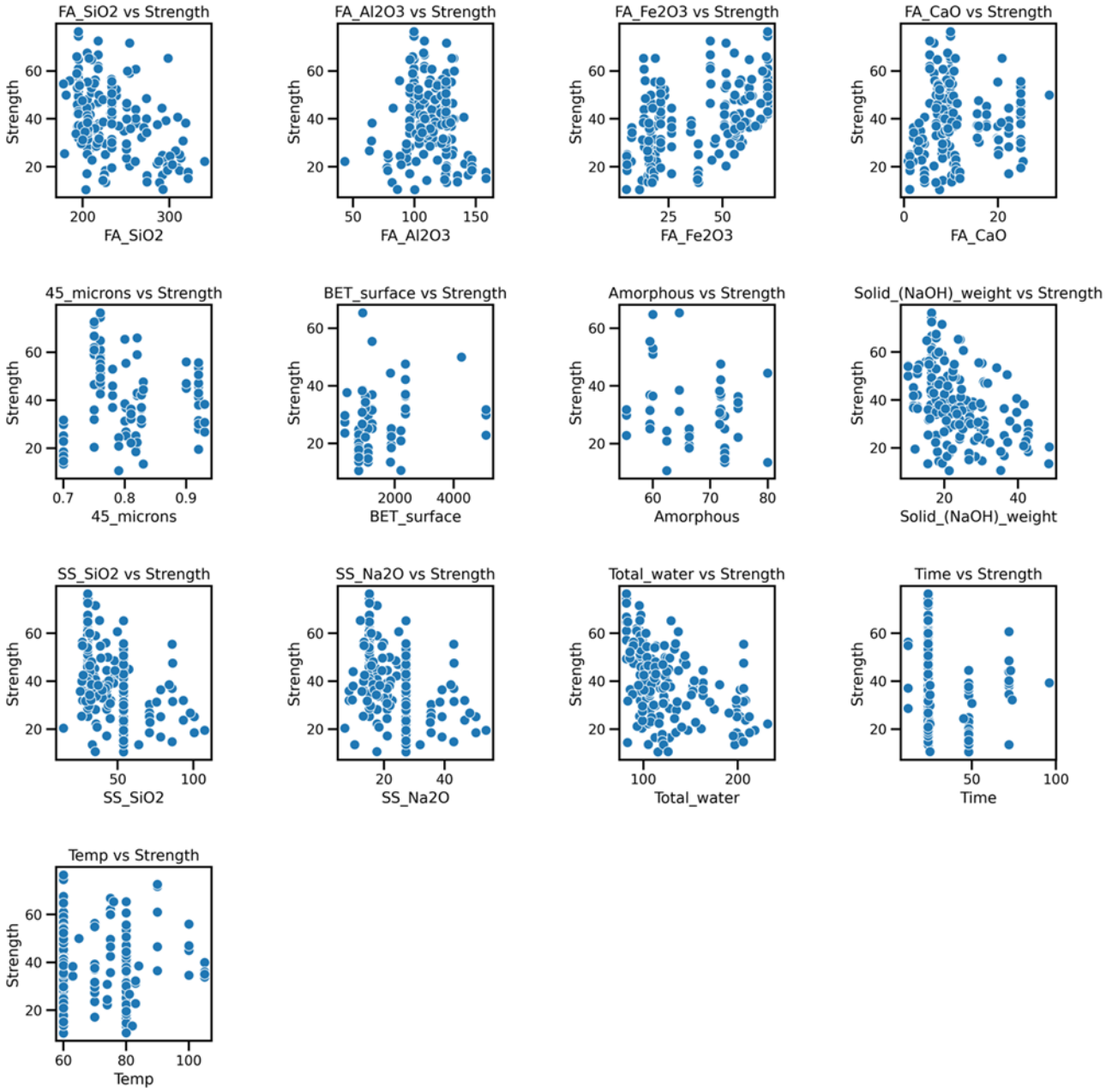


Figure 2 Initial data visualization.

2.2. Missing data mechanisms, missing data identification, and imputation

Missing data in the database were handled through data imputation. The steps shown in Figure 3 were used to identify and impute the missing values in the data set. Accordingly, the database was evaluated for missing values and data missing mechanisms. All missing data mechanisms were considered, namely missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). Little's MCAR test was used to rule out the MCAR with

MAR and MNAR cases. Based on the statistical significance (p-value), the acceptance or rejection of the null hypothesis was decided. If the significance is less than 0.05, the null hypothesis is rejected, and the data is not MCAR. To rule out the difference between MAR and MNAR, logistic regression was used. In this case a binary variable was developed for each of the missing variables (e.g., 1 for missing and 0 for non-missing) and used as a dependent variable. All other variables without missing data were considered as predicted. Logistic regression is used to identify the significance of each variable with missing data. If the significance is less than 0.05, then the data is not MAR and is MNAR.

Several data imputation techniques, namely K-nearest neighbor (KNN), multiple imputation using chain equations (MICE), soft imputation (Soft Imp) using singular value decomposition, and artificial neural networks (ANN), were used to impute missing data, as shown in Figure 3. In the current study, KNN was used with a varying number of neighbors, between 1 and 20, to identify the optimum number of neighbors for the imputation. For the MICE imputation method, Bayesian-Ridge regression was used as the estimator, while the maximum iterations were selected as 50. A tolerance value of 0.001 was used with 10 selected as the number of nearest neighbors. For the current study, SVD was used with both mean and median replacement methods while the rank of the SVD method varies between 1 to 12 since the number of variables considered in the study was 13. The convergence threshold was maintained at 0.0001 while the maximum number of iterations was limited to 100.

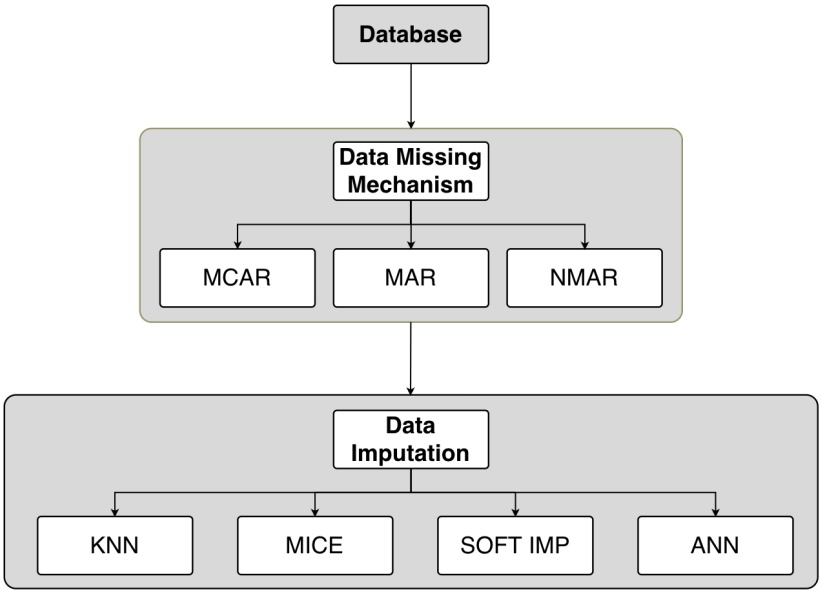


Figure 3 Data Imputation

As the final method of data imputation three, ANN models were used to generate missing values as shown in the Figure 4. Three ANN models were trained using complete data sets. For instance, first ANN model used complete data set with one missing input parameter "45 micron passing percentage" as output. Data sets without missing data were used as the training data and data set with missing values were used as test data.

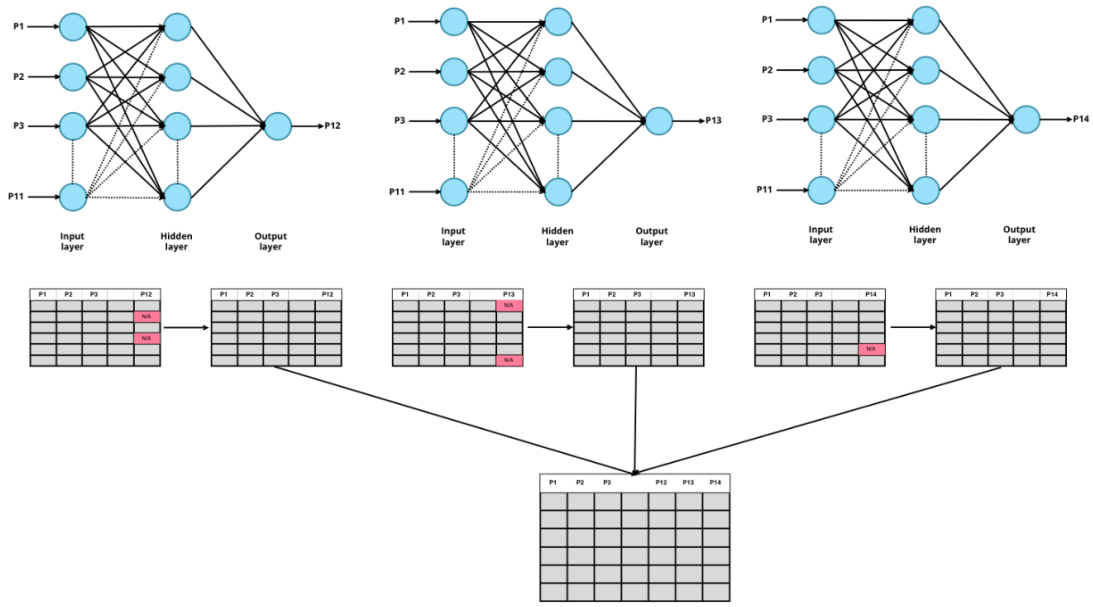


Figure 4 ANN model architecture for missing data imputation

For each of the parameters with missing values, an ANN model was developed and filled the missing values. Finally, full data sets were combined to get the final complete data set as shown in the Figure 4.

Data imputation was carried out in two steps as shown in Figure 5. In step 1 only two parameters were imputed resulting in a final dataset with 122 data points whereas in step 2 three parameters were imputed resulting in a final dataset with 226 data points. This enabled the identification of the performance of data imputation based on the percentage of missing data points.

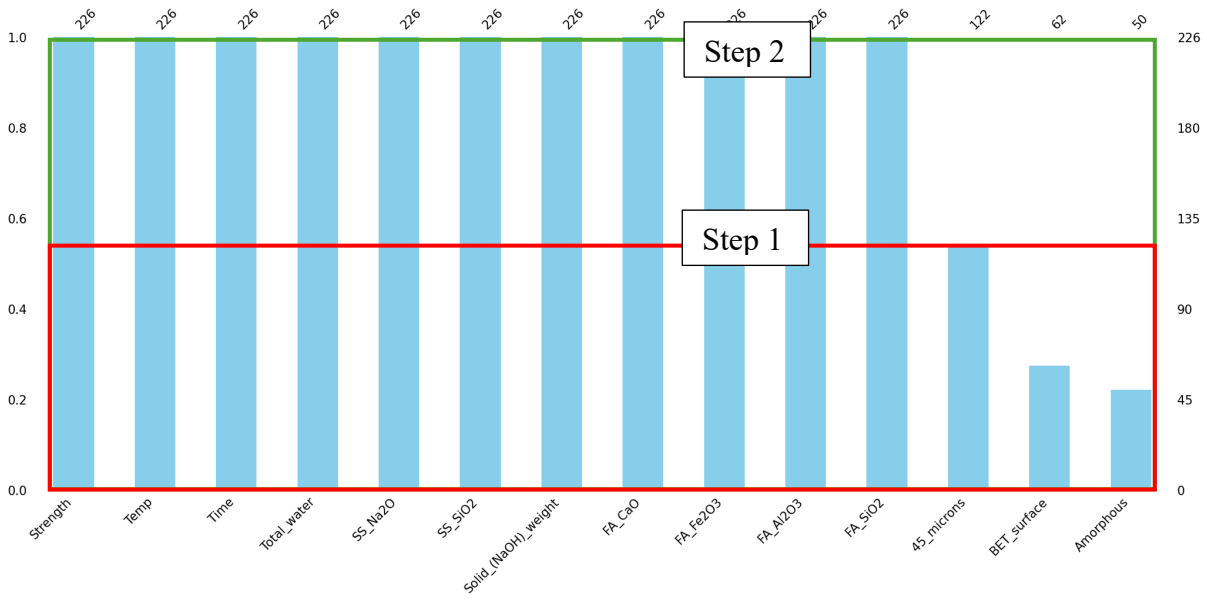


Figure 5 Database visualization with missing values

Three methods were used to evaluate the accuracy of the imputed data. First, statistical parameters were compared between imputation methods and the original data set, and if they yielded similar results, the imputed values were more likely to represent the actual missing values. The second approach is to compare the parameter correlation of the original data set and the imputed data set by developing a difference matrix. To identify differences between two correlation coefficient matrices, the signs at their corresponding positions were compared. A match is represented by 1, and a mismatch is indicated by -1. The difference is shown in a

heatmap, where matching signs are highlighted in one color and differing signs in another. If the difference matrix has the same color, it suggests that the correlation relationships in the augmented dataset are consistent with those in the original dataset. The third approach is to develop several machine learning models by using the imputed data as the input parameters to predict the compressive strength of GPC and compare the performance among the models. Consistent model performance suggests that the imputed values are likely reasonable.

2.3. Machine learning model development, hyperparameter tuning and validation

Three machine learning (ML) models, namely Artificial Neural Network (ANN), Random Forest (RF), and Extreme Gradient Boost (XGB) were used with extensive hyperparameter tuning. Initial hyperparameter selection was carried out randomly to identify the feasible initial values. Then an extensive grid search was conducted around the optimum values gained from the random search for all the ML models using the range of hyperparameters shown in Table 2.

Table 2 Hyperparameters value ranges for ML models

Model	**Hyperparmeter**	**Range**
ANN	Hidden neurons	2 to 50
	Learning rate	0.00000001 to 0.9
RF	Number of estimators	20 to 2000
	Maximum features	“auto” and “sqrt”
	Maximum depth	10 to 110
	Minimum sample split	2,5,20
	Minimum samples for leaf	1,2,4
	Bootstrap	“True” and “False”
XGB	Number of estimators	20 to 2000
	Maximum depth	1 to 100
	Learning Rate	0.01 to 0.2
	Colum sample by tree	0.4, 1.0
	Subsample	0.4, 1.0
	Gamma	0 to 1000
	Lambda	1 to 1000
	alpha	0 to 1000

Model performance was assessed using statistical functions shown in Eq 3 to 6 where y_a is the actual compressive strength, y_p is the predicted compressive strength, n is number of samples in the data set and $\overline{y_a}$ is the mean value of the actual compressive strength values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_a - y_p)^2} \quad \text{Eq. 3}$$

$$R = \frac{n\sum y_a y_p - \sum y_a \sum y_p}{\sqrt{[n\sum {y_a}^2 - (\sum y_a)^2]\left[n\sum {y_p}^2 - \left(\sum y_p\right)^2\right]}} \quad \text{Eq. 4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_a - y_p)^2}{\sum_{i=1}^{n}(y_a - \overline{y_a})^2} \quad \text{Eq. 5}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_a - y_p| \quad \text{Eq. 6}$$

For the model validation, two methods were used. Stratified K-fold cross-validation was used with 5 fold to ensure the performance of the model does not depend on the data used. By using stratified K-fold cross-validation compared to K-fold cross-validation same sort of data distribution was provided to both the train and the test data splits. Furthermore, a SHAP analysis of the model was conducted to ensure the model behaves in a similar manner observed in the laboratory experiments in literature.

3. Results and discussion

3.1 Missing data mechanism

Figure 6 shows the output results of Little's MCAR test. Based on the value of statistical significance, which is less than 0.05, null hypothesis will be rejected. Thus, the data is not MCAR.

EM Correlations[a,b]

	FA_SiO2	FA_Al2O3	FA_Fe2O3	FA_CaO	@45_microns	BET_surface	Amorphous	Solid_NaOH_weight	SS_SiO2	SS_Na2O	Total_water	Time	Temp
FA_SiO2	1												
FA_Al2O3	.089	1											
FA_Fe2O3	-.654	-.100	1										
FA_CaO	-.200	.276	-.140	1									
@45_microns	.151	-.266	-.405	.627	1								
BET_surface	-.484	-.263	.719	-.192	-.367	1							
Amorphous	.083	-.164	-.120	-.170	.226	-.444	1						
Solid_NaOH_weight	.350	.307	-.420	.041	.012	-.457	.113	1					
SS_SiO2	.371	.129	-.576	.174	.204	-.491	.086	.437	1				
SS_Na2O	.343	.141	-.537	.186	.200	-.464	.042	.434	.990	1			
Total_water	.254	.086	-.484	.173	.187	-.398	.216	.503	.786	.759	1		
Time	.368	.062	-.429	.029	.113	-.363	.214	.049	.083	.021	.178	1	
Temp	.152	.045	-.385	.167	.369	-.313	.542	.190	.400	.369	.412	.104	1

a. Little's MCAR test: Chi-Square = 292.511, DF = 43, Sig. = .000

Figure 6 Results of Little's MCAR test

According to Figure 7, the significance of CaO, amorphous content, NaOH solid weight, SiO_2 and Na_2O in Na_2SiO_3, total water and curing time is less than 0.05, thus it was concluded that the data is MNAR.

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means							
						Significance		Mean	Std. Error	95% Confidence Interval of the Difference	
		F	Sig.	t	df	One-Sided p	Two-Sided p	Difference	Difference	Lower	Upper
FASiO2	Equal variances assumed	.572	.450	2.041	224	.021	.042	9.82841143	4.81599928	.33795047	19.31887240
	Equal variances not assumed			2.038	217.246	.021	.043	9.82841143	4.82211645	.32429103	19.33253184
FAAl2O3	Equal variances assumed	.381	.538	2.172	224	.015	.031	4.731721063	2.178820449	.438113604	9.025328523
	Equal variances not assumed			2.159	212.413	.016	.032	4.731721063	2.191704814	.411443451	9.051998676
FAFe2O3	Equal variances assumed	6.158	.014	-.399	224	.345	.691	-1.17726962	2.952874887	-6.99623724	4.641698001
	Equal variances not assumed			-.405	223.690	.343	.686	-1.17726962	2.907104709	-6.90608510	4.551545864
FACaO	Equal variances assumed	22.193	<.001	-.163	224	.435	.870	-.136561477	.835888245	-1.78377202	1.510649061
	Equal variances not assumed			-.168	212.394	.433	.866	-.136561477	.810631779	-1.73447561	1.461352658
BETsurface	Equal variances assumed	2.887	.094	-3.304	60	<.001	.002	-930.4697	281.5926	-1493.7388	-367.2006
	Equal variances not assumed			-3.590	38.301	<.001	<.001	-930.4697	259.1845	-1455.0257	-405.9137
Amorphous	Equal variances assumed	18.322	<.001	-2.943	48	.002	.005	-7.84909	2.66707	-13.21158	-2.48660
	Equal variances not assumed			-8.041	43.000	<.001	<.001	-7.84909	.97614	-9.81766	-5.88052
Solid_NaOH_weight	Equal variances assumed	13.937	<.001	-2.299	224	.011	.022	-2.39941532	1.04353064	-4.45580825	-.343022395
	Equal variances not assumed			-2.352	220.505	.010	.020	-2.39941532	1.02034088	-4.41028337	-.388547272
SS_SiO2	Equal variances assumed	50.413	<.001	-3.357	224	<.001	<.001	-7.88069990	2.34750458	-12.5067182	-3.25468165
	Equal variances not assumed			-3.534	177.215	<.001	<.001	-7.88069990	2.22992846	-12.2813315	-3.48006830
SS_Na2O	Equal variances assumed	50.845	<.001	-3.138	224	<.001	.002	-3.77925609	1.20443148	-6.15272196	-1.40579021
	Equal variances not assumed			-3.299	179.878	<.001	.001	-3.77925609	1.14564651	-6.03989143	-1.51862074
Total_water	Equal variances assumed	48.762	<.001	-3.175	224	<.001	.002	-15.0473884	4.73894498	-24.3860053	-5.70877144
	Equal variances not assumed			-3.322	189.387	<.001	.001	-15.0473884	4.52999095	-23.9831086	-6.11166814
Time	Equal variances assumed	74.516	<.001	3.682	224	<.001	<.001	6.314	1.715	2.935	9.693
	Equal variances not assumed			3.504	143.492	<.001	<.001	6.314	1.802	2.752	9.876
Temp	Equal variances assumed	1.616	.205	-2.056	224	.020	.041	-3.213	1.563	-6.293	-.133
	Equal variances not assumed			-2.041	210.631	.021	.043	-3.213	1.575	-6.317	-.109
Strength	Equal variances assumed	5.192	.024	-2.330	224	.010	.021	-4.37582792	1.87796863	-8.07657349	-.675082361
	Equal variances not assumed			-2.362	223.966	.010	.019	-4.37582792	1.85244091	-8.02627132	-.725384528

Figure 7 Independent sample test considering 45-micron passing as a target

3.2 Missing Data identification and Missing data imputation

3.2.1 Evaluation of Imputation Methods for Step 1 Data imputation.

In this section the accuracy of different imputation methods for missing data in geopolymer concrete datasets were assessed, focusing on BET surface area and amorphous content as shown in Figure 8. The analysis compares five imputation methods: ANN as an imputation method, KNN, MICE, and SDV using both mean and median. The imputation was performed on a dataset of 122 records.

BET Surface Area

The original BET surface area dataset showed a skewness of 1.90, a mean of 1528 m^2/kg, and a standard deviation of 1085 m^2/kg, with values ranging from 310 to 5095 m^2/kg.

- **ANN Method**: The imputed dataset exhibited a significant reduction in skewness (-0.02), suggesting a nearly symmetric distribution, which deviates from the original positively skewed data. The mean was overestimated at 3075 m^2/kg, with a larger standard deviation of 1511 m^2/kg. The minimum (890 m^2/kg) and maximum (5095 m^2/kg) were within acceptable ranges, but the method overestimates the mean and spreads the values over a wide range.
- **KNN Method**: The skewness decreased to -0.22, moving away from the original distribution. The mean was 2541 m^2/kg, which is significantly higher than the original 1528 m^2/kg, though the standard deviation decreased to 862 m^2/kg. The minimum (890 m^2/kg) and maximum (5095 m^2/kg) remained within the original range.
- **MICE Method**: This method resulted in a slight positive skewness (0.08), with a mean of 2217 m^2/kg and a large standard deviation of 1497 m^2/kg. However, the minimum value (-2030 m^2/kg) was negative, which is not physically possible, making this method unsuitable for BET surface area imputation. The maximum value (5930 m^2/kg) was also

higher than the original maximum, adding to the inaccuracy of this method for this variable.

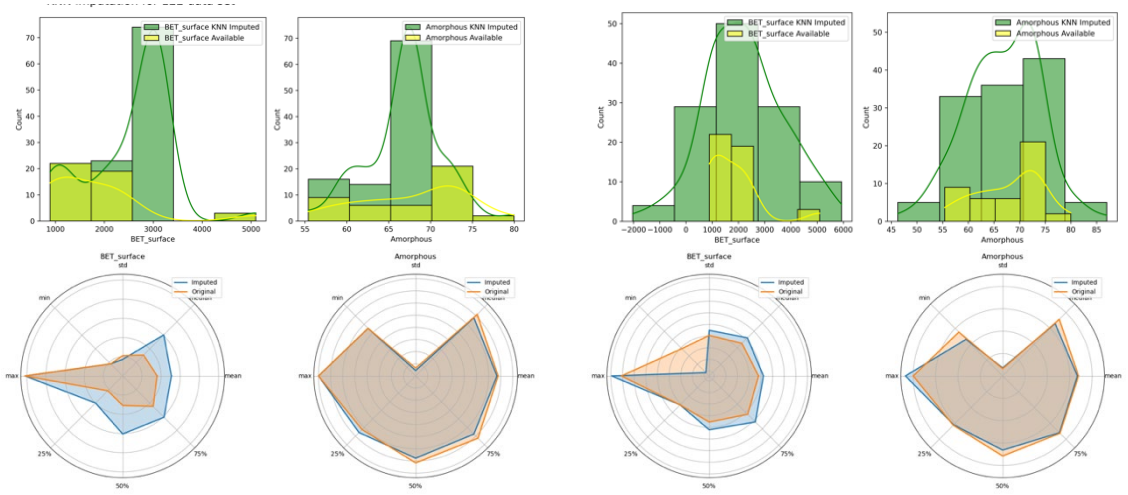
- **SDV (Mean):** The skewness increased to 3.3, which closely resembled the original dataset's positive skew. The mean (1798 m^2/kg) was a slight overestimation compared to the original, but the standard deviation decreased to 629 m^2/kg. The minimum (890 m^2/kg) and maximum (5095 m^2/kg) were accurate.
- **SDV (Median):** This method resulted in the highest skewness (3.87), with a mean of 1633 m^2/kg, which was closest to the original mean of 1528 m^2/kg. The standard deviation (641 m^2/kg) was lower, and the minimum (890 m^2/kg) and maximum (5095 m^2/kg) were within the original range.

Among the imputation methods for BET surface area, ANN Imputation, KNN imputation, SDV (Median) and SDV (Mean) provided the more accurate results, maintaining the range of values closest to the original dataset compared to the MICE method.

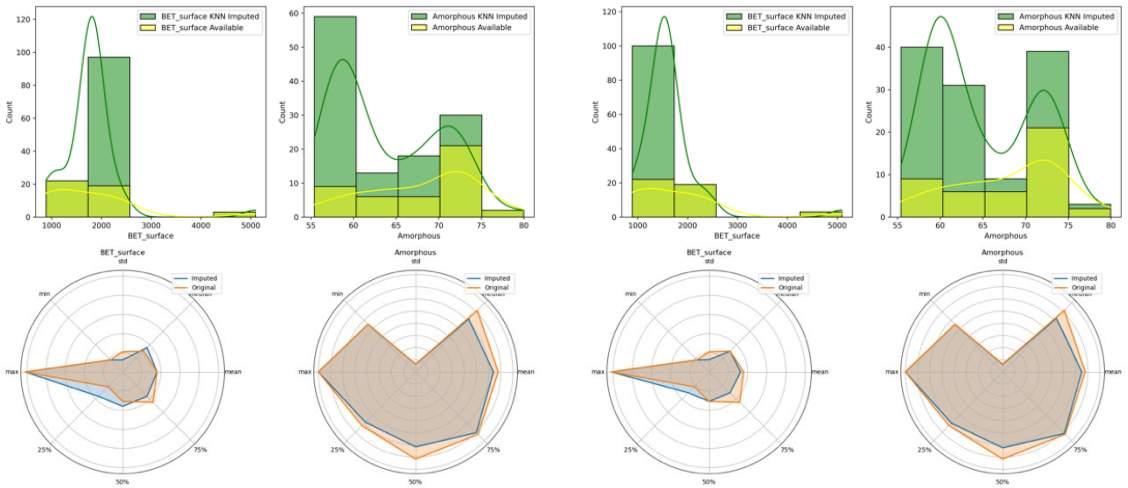
Amorphous Content

The original dataset for amorphous content had a skewness of -0.04, a mean of 66.9%, and a standard deviation of 6.5%. The values ranged from 55.4% to 79.98%.

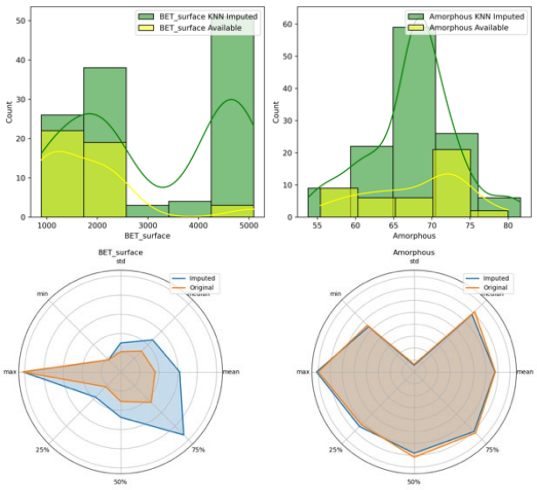
- **ANN Method:** The imputed data exhibited a slight negative skew (-0.19) and a mean of 67.5%, which is slightly higher than the original mean. The standard deviation was reduced to 5.6%, and the minimum (53.7%) and maximum (81.6%) were within an acceptable range.
- **KNN Method:** The skewness (-0.12) was closer to the original distribution, and the mean (66.88%) was almost identical to the original value of 66.9%. The standard deviation (4.44%) was lower, indicating a tighter clustering of the imputed values. The minimum (55.4%) and maximum (79.98%) perfectly matched the original range.



(a) KNN imputtaion (b) MiCE imputation



(c) SVD Imputation (Mean) (d) SVD Imputation (Median)



(e) ANN imputation

Figure 8 Data histograms and summary statistics of step 1 data imputation

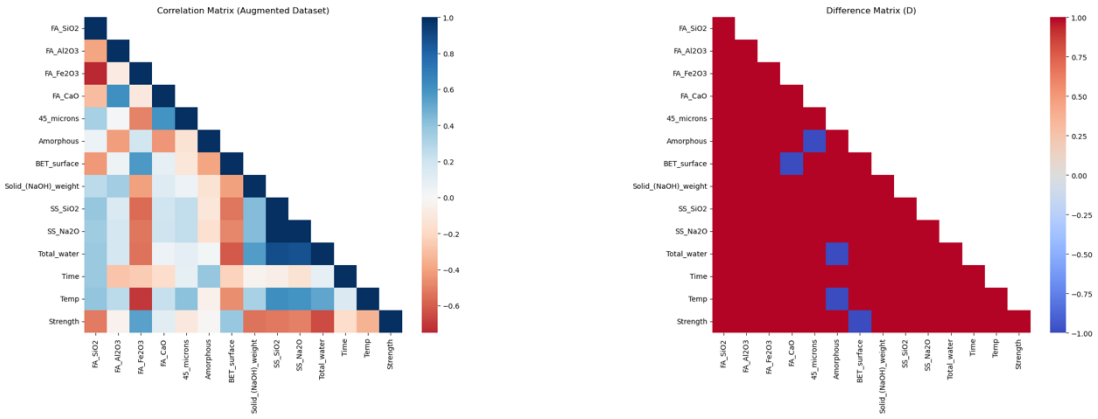
- **MICE Method:** This method produced a skewness of -0.19, similar to ANN, but the mean (66.7%) was slightly lower than the original. The standard deviation (7.2%) was

higher, reflecting greater variability in the imputed data. The minimum (46.26%) was outside the original range, making this method less reliable for amorphous content.

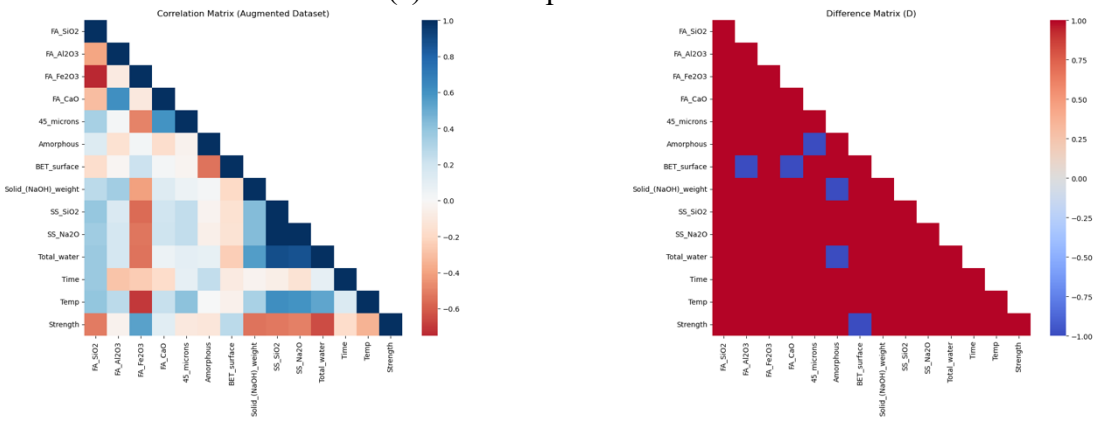
- **SDV (Mean):** The skewness shifted to 0.50, indicating a positive skew, which deviated from the original distribution. The mean (63.9%) was an underestimation, and the standard deviation (6.31%) was close to the original. The minimum (55.4%) and maximum (79.98%) matched the original values.
- **SDV (Median):** This method resulted in a skewness of 0.44, a mean of 65.03%, and a standard deviation of 6.81%. The minimum (55.4%) and maximum (79.98%) were within the original range.

For amorphous content, all imputation methods provided a closes match to the original data, both in terms of skewness, mean, and range, making it the most accurate imputation method for this variable.

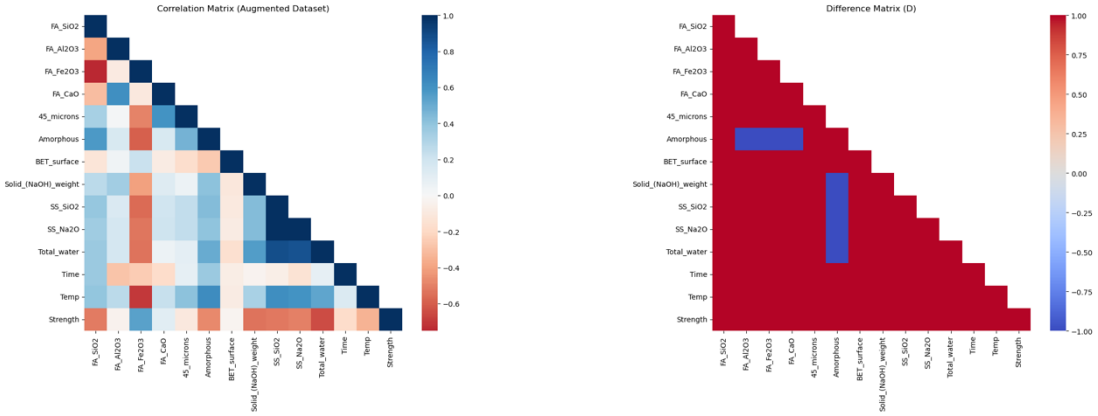
Figure 9 shows the D matrices for each data imputation technique. The left column indicates correlation matrix of imputed data set and on the right column shows the difference matrix for each of the imputation methods considered in Step 1 Data Imputation. The difference matrix values, expressed as percentages of the total number of entries (105), revealed the following. A discrepancy of 4.8% was observed for the correlation of KNN imputed data, indicating the closest alignment with the original data correlations. MICE showed a 5.7% discrepancy, while discrepancies of 6.7% were recorded for both SVD Mean and ANN. The highest discrepancy of 10.5% was exhibited by the SVD Median method, suggesting the greatest deviation from the original correlations.



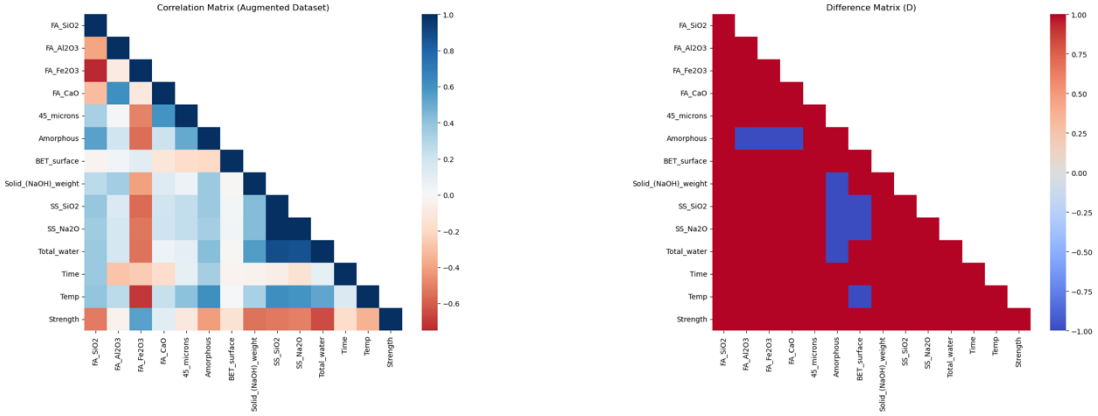
(a) KNN imputation method



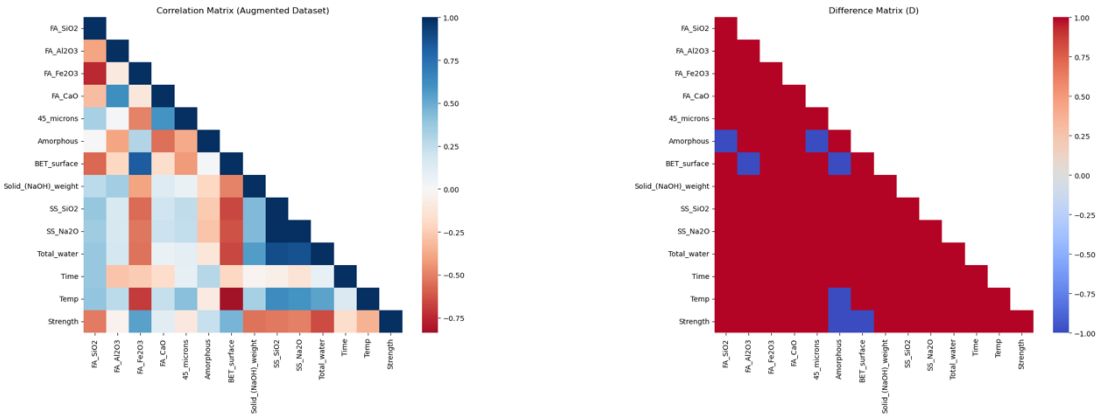
(b) MICE imputation method



(c) SVD (Mean) imputation method



(d) SVD (Median) imputation method



(e) ANN imputation method

Figure 9 D-matrices for step 1 data imputation

Table 3 and Figure 10 provide comprehensive comparison of three machine learning models ANN, Extreme Gradient Boosting (XGB), and Random Forest (RF) across four different imputation methods: KNN, SVD mean, SVD median, and ANN-based imputation for Step 1 Data Imputation. The models were evaluated based on their performance on training and test datasets using three metrics: R^2, RMSE, and MAE.

For the ANN model, the results show that imputation methods have a profound impact on the model performance. ANN-based imputation produced the highest R^2 value of 0.84 on the training set, demonstrating superior data imputation capability compared to KNN (R^2 = 0.64) and SVD methods (R^2 = 0.75 and 0.72 for SVD mean and SVD median, respectively). The corresponding RMSE and MAE values for ANN imputation (5.74 and 4.16) were also the lowest, indicating that this method yields more accurate predictions. Similarly, on the test set, the ANN model paired with ANN imputation outperformed all other methods with an R^2 of 0.80, highlighting its generalization ability. However, the model's RMSE (7.35) and MAE (5.32) suggest that while it generalizes well, some prediction errors persist, which may be due to the complexity of the dataset. The SVD mean imputation method also performed reasonably well on the test set, but the ANN-based method consistently delivered the best results.

The XGB model showed strong results overall, but like ANN model, it benefitted most from ANN-based imputation. On the training set, the XGB model with ANN imputation achieved

the highest R^2 (0.91) and the lowest RMSE (4.71) and MAE (3.23), outperforming other imputation methods. KNN also yielded decent results ($R^2 = 0.89$), but it fell short compared to ANN imputation. When evaluated on the test set, the XGB model again displayed the optimum performance with ANN imputation, achieving an R^2 of 0.66. However, there was a notable decrease in the R^2 value between the training and test sets, and the RMSE increased from 4.71 to 7.68. The SVD methods provided moderate performance, but they were consistently outperformed by the ANN imputation method. These results suggest that while XGB is a powerful model, its performance, especially with large datasets and missing values, can be significantly improved through advanced imputation techniques like ANN.

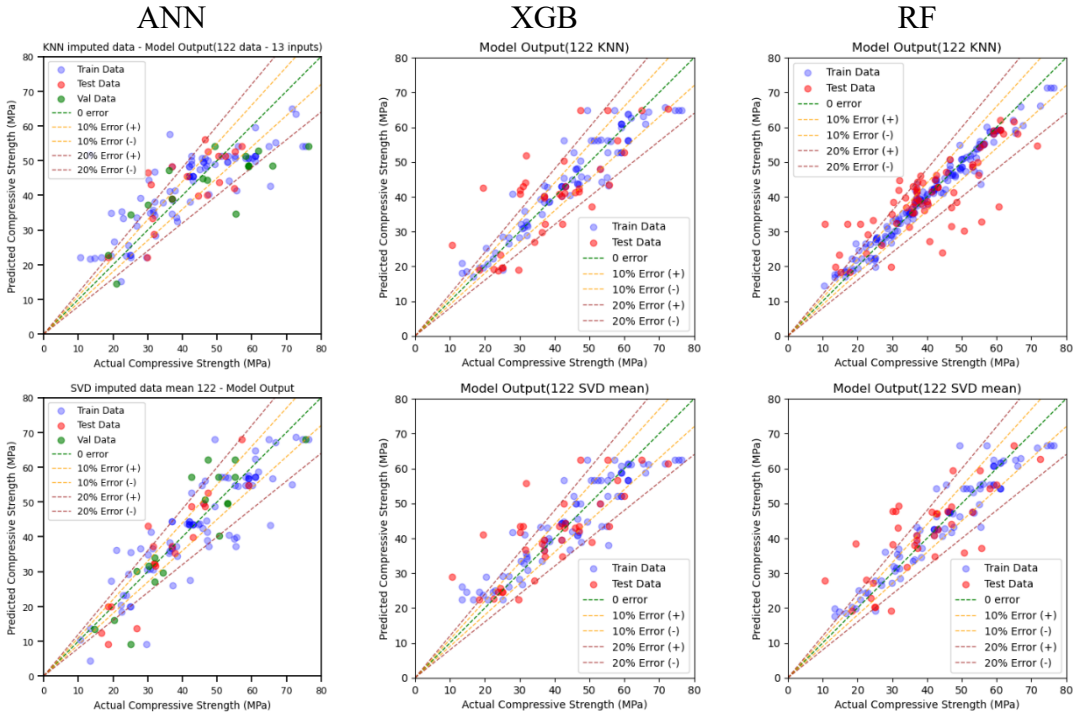
The RF model exhibited the strongest overall performance, particularly when paired with ANN imputation. In the training set, RF with ANN imputation produced an R^2 of 0.95, the highest among all models and methods, with an exceptionally low RMSE of 3.34 and MAE of 2.26. This demonstrates that RF, when provided with well-imputed data, is highly effective at capturing complex patterns. The performance in the test set remained robust, with an R^2 of 0.74 and relatively low RMSE (6.71) and MAE (4.80). This indicates that the RF model with ANN imputation generalizes better than the other imputation methods. KNN and SVD imputation methods, on the other hand, consistently resulted in lower R^2 values and higher RMSE and MAE values, both in training and test sets, further underscoring the effectiveness of ANN imputation in enhancing the RF model's performance.

Across all three models, ANN-based imputation consistently outperforms traditional imputation methods such as KNN and SVD. The superior performance of the ANN imputation method is evident in both training and test sets, regardless of the machine learning model used. This result highlights the ability of machine learning-based imputation methods to handle missing data more effectively by capturing complex relationships in the dataset that simpler methods may overlook. KNN, which relies on proximity-based data reconstruction, and SVD,

which uses low-rank approximations, seem inadequate for this dataset compared to ANN-based imputation, which better preserves the integrity of the data and leads to improved model predictions.

Table 3 Model performance of dataset with step 1 data imputation.

ML model	Data set		Imputed method			
			KNN	SVD mean	SVD median	ANN
ANN	**Train**	R^2	0.64	0.75	0.72	0.84
		RMSE	9.14	7.50	8.30	5.74
		MAE	6.92	5.32	5.24	4.16
	Test	R^2	0.49	0.75	0.66	0.80
		RMSE	7.52	6.13	8.22	7.35
		MAE	6.12	4.56	5.26	5.32
XGB	**Train**	R^2	0.89	0.83	0.84	0.91
		RMSE	5.02	6.24	6.2	4.71
		MAE	3.65	4.85	4.85	3.23
	Test	R^2	0.54	0.58	0.58	0.66
		RMSE	9.28	8.92	8.93	7.68
		MAE	7.37	6.47	6.5	5.41
RF	**Train**	R^2	0.89	0.90	0.90	0.95
		RMSE	5.13	4.74	4.76	3.34
		MAE	3.84	3.51	3.51	2.26
	Test	R^2	0.61	0.57	0.52	0.74
		RMSE	8.62	9.04	9.55	6.71
		MAE	5.36	6.88	7.24	4.80



SVD (Median)

ANN Imputed dataset

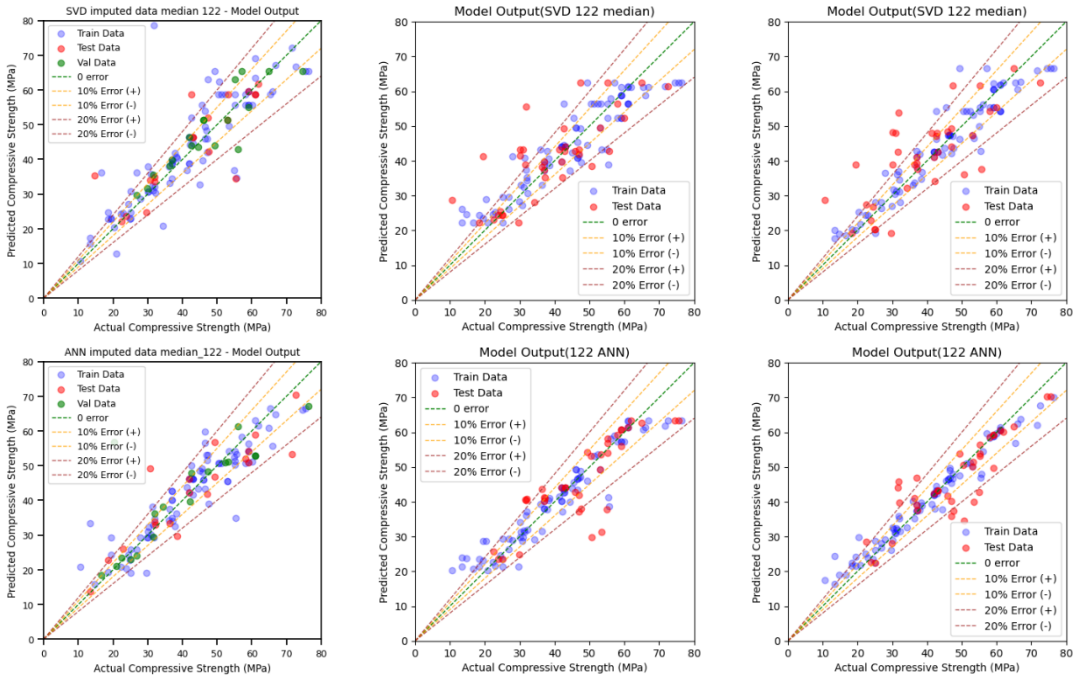


Figure 10 ML model performance matrix for step 1 data imputation

3.2.2 Evaluation of Imputation Methods for Step 2 Data Imputation

For the step two Data Imputation the imputation was applied to a dataset containing 226 records for three key variables: 45-micron particle size, BET surface area, and amorphous content.

45-Micron Particle Size

The original dataset for the 45-micron particle size distribution had a skewness of 0.64, a mean of 0.80, and a standard deviation of 0.06. The minimum and maximum values ranged between 0.70 and 0.93, respectively. After imputation, the following observations were made:

- **ANN Method:** The imputed data showed reduced skewness (0.28) with a slightly lower mean of 0.79 and an increased standard deviation of 0.07. However, the minimum value (0.59) was significantly lower than the original minimum (0.70).
- **KNN Method:** This method yielded the best results, with a skewness of 0.74, a mean of 0.80, and a standard deviation of 0.06. The minimum and maximum values (0.70 and 0.93) were identical to the original dataset.

- **MICE Method:** While the mean (0.80) closely matched the original data, the skewness shifted to -0.006, indicating a significant alteration in the distribution. Moreover, the minimum (0.52) and maximum (1.05) were outside the original and practical ranges.
- **SDV (Mean):** The skewness (0.69), mean (0.79), and standard deviation (0.07) were acceptable, though the minimum value (0.65) was lower than the original. The maximum value (0.96) was slightly higher than the original.
- **SDV (Median):** Similar to the mean-based SDV method, the skewness (0.79) and mean (0.79) were acceptable, but the minimum (0.64) was lower than the original value. The maximum (1.01) was overestimated compared to the original range.

BET Surface Area

The BET surface area in the original dataset had a highly skewed distribution (skewness: 1.90), a mean of 1528 m^2/kg, and a standard deviation of 1085 m^2/kg. The minimum and maximum values ranged from 310 to 5095 m^2/kg.

- **ANN Method**: The skewness of the imputed data reduced drastically to -0.002, indicating a shift to a nearly symmetric distribution, which is not consistent with the original. The mean increased to 2385 m^2/kg, and the standard deviation rose to 1182 m^2/kg, indicating an overestimation of the BET surface area. The minimum and maximum values remained consistent with the original data.
- **KNN Method:** The skewness (-0.42) was similarly reduced, indicating a deviation from the original positive skewness. The mean (2363 m^2/kg) was overestimated, but the standard deviation (1056 m^2/kg), minimum (310 m^2/kg), and maximum (5095 m^2/kg) remained close to the original values.
- **MICE Method:** This method performed reasonably well, maintaining a positive skewness (0.37) closer to the original. The mean (1710 m^2/kg) was within an acceptable

range, though the standard deviation increased to 1385 m^2/kg. However, the minimum value (-1499 m^2/kg) was negative, which is physically impossible, making this method unsuitable for this variable.

- **SDV (Mean):** This method provided the best match to the original data, with a skewness of 3.63, a mean of 1528 m^2/kg, and a reduced standard deviation of 565 m^2/kg. The minimum and maximum values matched the original data perfectly, but the reduction in variability suggests some smoothing in the imputation process.
- **SDV (Median):** The skewness (4.52) was much higher than the original, and the mean (1213 m^2/kg) was underestimated. The standard deviation (597 m^2/kg) was also reduced, but the minimum and maximum values were consistent with the original data.

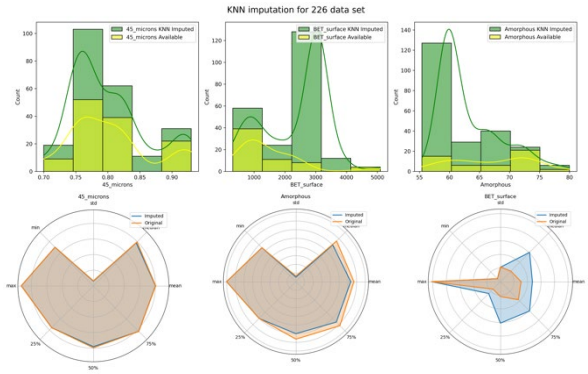
Amorphous Content

The original dataset for amorphous content showed a skewness of -0.04, a mean of 66.9%, and a standard deviation of 6.5%. The minimum and maximum values were 55.4% and 79.98%, respectively.

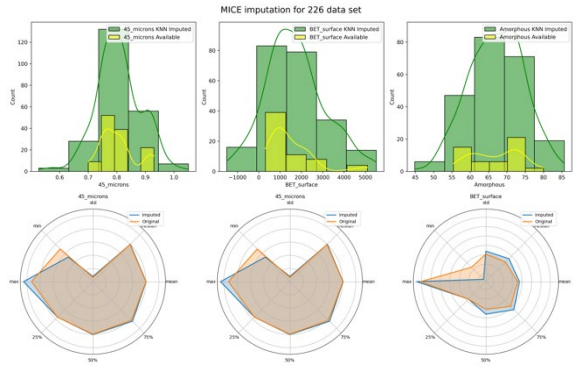
- **ANN Method:** The skewness shifted to 0.37, with a mean of 61.94%, which is an underestimation compared to the original. The minimum value dropped to 43.53%, which is outside the original range, and the maximum (98.19%) was overestimated.
- **KNN Method:** The skewness increased to 1.2, suggesting a positive skew. The mean was 63.2%, and the standard deviation reduced to 5.18%. The minimum and maximum values matched the original data.
- **MICE Method**: The skewness (-0.10) was close to the original, and the mean (67.1%) was well within the acceptable range. However, the minimum value (44.8%) was too low, and the maximum (85.67%) was slightly overestimated.

- **SDV (Mean):** The skewness (0.65) increased, and the mean (64.68%) was slightly underestimated. The standard deviation (6.38%) and the maximum (80.22%) were acceptable, though the minimum (54.7%) was slightly lower than the original.
- **SDV (Median):** The skewness (0.78) was higher than the original, and the mean (64.39%) was underestimated. The minimum (53.38%) was lower than the original, but the maximum (83.84%) was within an acceptable range.

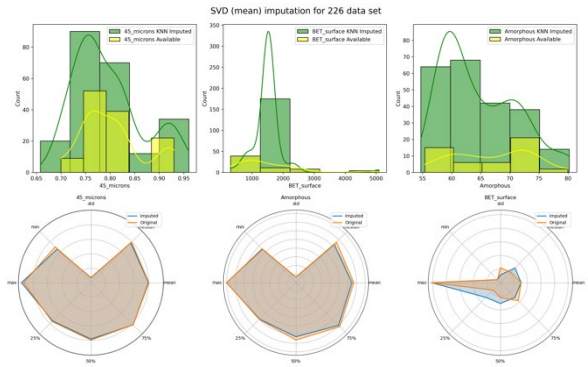
Each imputation method demonstrated varying degrees of accuracy across the three variables. The KNN method proved to be the most consistent across all variables, especially for the 45-micron particle size and BET surface area. SDV (Mean) provided the best match for BET surface area, while MICE was the most accurate for amorphous content, though care should be taken with this method due to occasional outlier imputation.



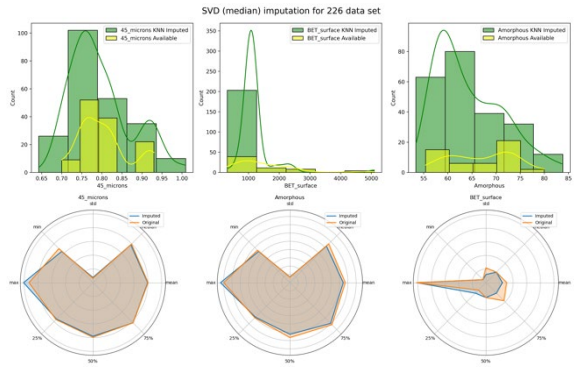
(a)



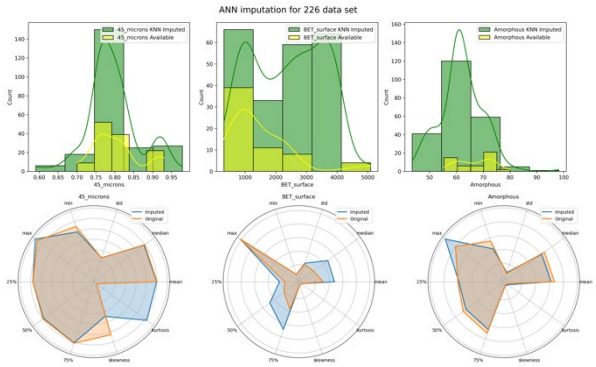
(b) MICE imputation



(c) SVD Imputation (Mean)



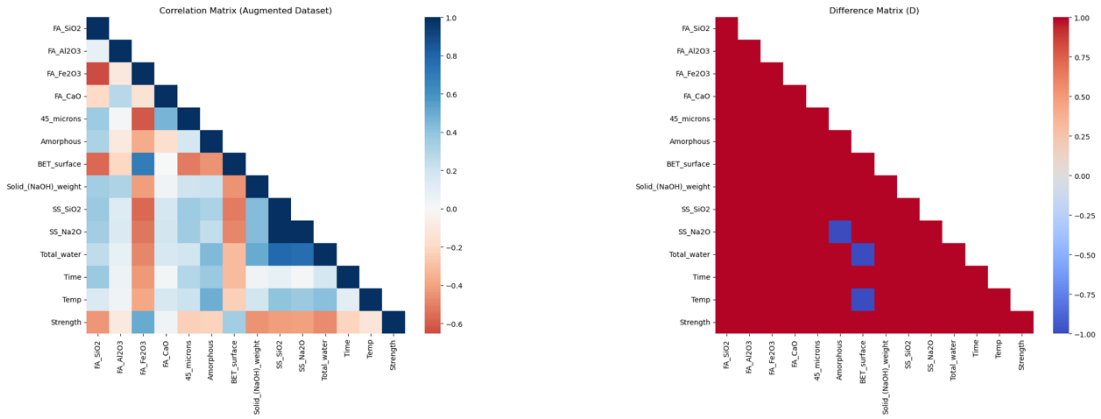
(d) SVD Imputation (Median)



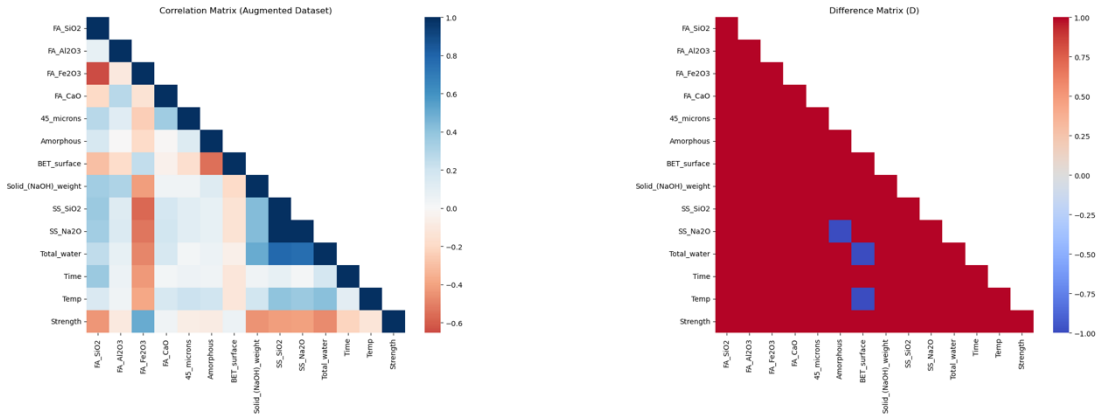
(e) ANN imputation

Figure 11 Data histograms and Summary statistics of step 2 data imputation

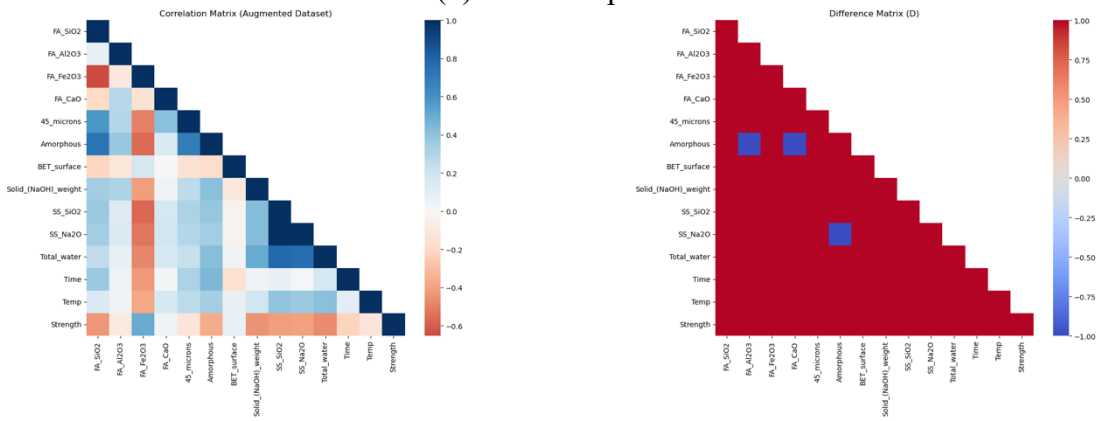
Figure 12 shows the correlation matrix of imputed data set in the left column and difference matrix on the right column for each of the imputation methods considered in Step 2 Data Imputation. The difference matrix values, expressed as percentages of the total number of entries (105), revealed the following. A discrepancy of 2.68% was observed for the correlation of KNN imputed, SVD (mean)and MICE imputed data, indicating the closest alignment with the original data correlations. SVD (median) showed a 7.63% discrepancy, while the highest discrepancy of 9.52% was exhibited by the ANN imputation method, suggesting the greatest deviation from the original correlations.



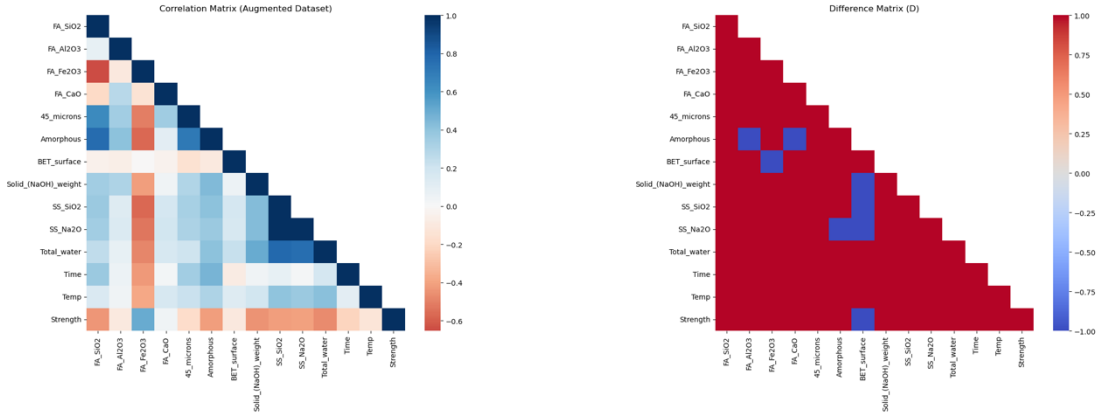
(a) KNN Imputation



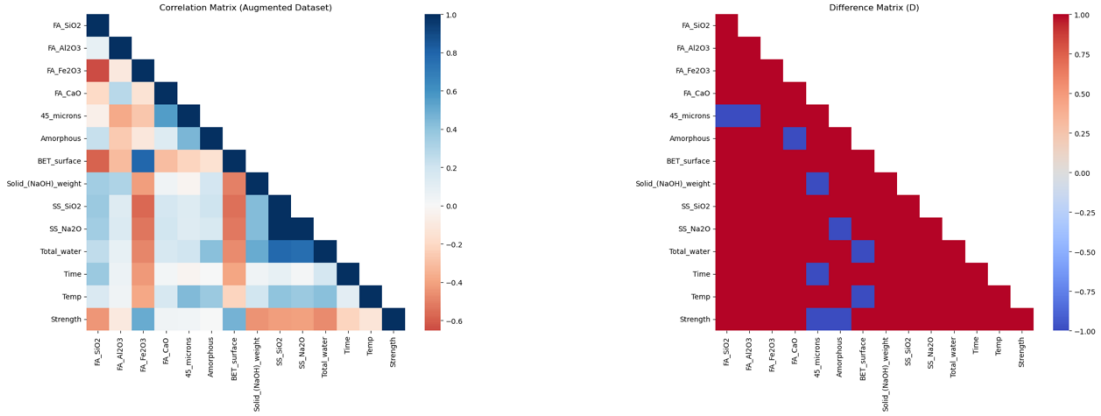
(b) MICE Imputation



(c) SVD (Mean) Imputation



(d) SVD (Median) Imputation



(e) ANN Imputation

Figure 12 D-matrices for step 2 data imputation

The analysis of Table 4 and Figure 13 demonstrates the performance of ANN, XGB, and RF models across different imputation methods: KNN, SVD mean, SVD median, and ANN for Step 2 Data Imputation. The results are evaluated on the training and test datasets using R^2, RMSE, and MAE as metrics.

For the ANN model the training set results indicate that SVD mean imputation is optimum with an R^2 of 0.72, followed closely by KNN and ANN imputation methods, both achieving an R^2 of 0.63. Interestingly, SVD median imputation performs significantly worse with an R^2 of only 0.34, suggesting that it struggles to capture the relationships in the data effectively. This is further reflected in the RMSE and MAE values, where SVD median produces the highest RMSE (11.32) and relatively high MAE (6.27). With respect to the test data set, however, ANN imputation excels, yielding an R^2 of 0.76, which is the highest among all methods. The RMSE (7.93) and MAE (5.83) values also support its strong performance. SVD mean follows closely behind with an R^2 of 0.74 and RMSE of 7.31, but the slightly higher MAE of 5.82 suggests it is less accurate than ANN-based imputation. These results again highlight that ANN-based imputation is superior for both training and test sets in the ANN model, ensuring better generalization.

For the XGB model ANN imputation clearly stands out as the best-performing method on the training set, achieving an almost perfect R^2 of 0.99. This result, combined with the exceptionally low RMSE of 1.32 and MAE of 0.54, indicates that the model is highly optimized when paired with ANN imputation. By contrast, other imputation methods such as KNN, SVD mean, and SVD median produce similar but lower R^2 values (around 0.85) and higher RMSE and MAE scores, indicating less effective handling of the missing data. However, on the test set, the performance of the XGB model declines for all imputation methods. ANN imputation still leads with an R^2 of 0.65 and the lowest RMSE of 8.17 and MAE of 5.58, though these values suggest some degree of overfitting on the training data. KNN and SVD methods hover

around an R^2 of 0.50 to 0.52, with higher RMSE values exceeding 10, indicating poorer generalization ability compared to ANN-based imputation.

The RF model similarly benefits the most from ANN-based imputation. On the training set, the model achieves a high R^2 of 0.96 and the lowest RMSE (2.71) and MAE (1.99), outperforming other methods like KNN, SVD mean, and SVD median, which show R^2 values of around 0.85 to 0.91. These alternative imputation methods lead to higher RMSE and MAE values, indicating less efficient imputation of missing data compared to the ANN-based approach. On the test set, RF with ANN imputation continues to demonstrate superior performance, with an R^2 of 0.62 and RMSE of 8.57, outperforming the KNN, SVD mean, and SVD median methods, which deliver R^2 values around 0.50 and higher RMSE values exceeding 10. The improvement in MAE with ANN imputation (6.11) compared to the other methods (around 7.27 to 7.55) further emphasizes the effectiveness of ANN imputation in preserving data structure for accurate predictions.

In summary, ANN-based imputation provides the most consistent improvements in both training and test datasets, delivering better R^2 values and lower RMSE and MAE scores. KNN and SVD methods generally perform adequately but are less reliable than ANN-based imputation, particularly for test data, where they show lower R^2 values and higher prediction errors. These findings suggest that ANN imputation is a more robust method for handling missing data in this context, leading to improved model performance and better generalization to unseen data.

Table 4 Model performance of dataset with step 2 data imputation.

ML model	Data set		Imputed dataset			
			KNN	SVD mean	SVD median	ANN
ANN	**Train**	R^2	0.63	0.72	0.34	0.63
		RMSE	8.78	7.33	11.32	7.91
		MAE	6.2	5.63	6.27	5.71
	Test	R^2	0.38	0.74	0.66	0.76
		RMSE	9.19	7.31	8.73	7.93
		MAE	6.59	5.82	6.42	5.83
XGB	**Train**	R^2	0.86	0.85	0.85	0.99
		RMSE	5.22	5.39	5.36	1.32
		MAE	3.83	4.06	4.05	0.54

	Test	R^2	0.51	0.52	0.49	0.65
		RMSE	10.17	10.01	10.40	8.17
		MAE	7.61	7.43	7.83	5.58
RF	**Train**	R^2	0.90	0.91	0.85	0.96
		RMSE	4.38	4.14	5.41	2.71
		MAE	3.28	3.07	4.13	1.99
	Test	R^2	0.52	0.50	0.49	0.62
		RMSE	10.04	10.31	10.4	8.57
		MAE	7.27	7.42	7.55	6.11

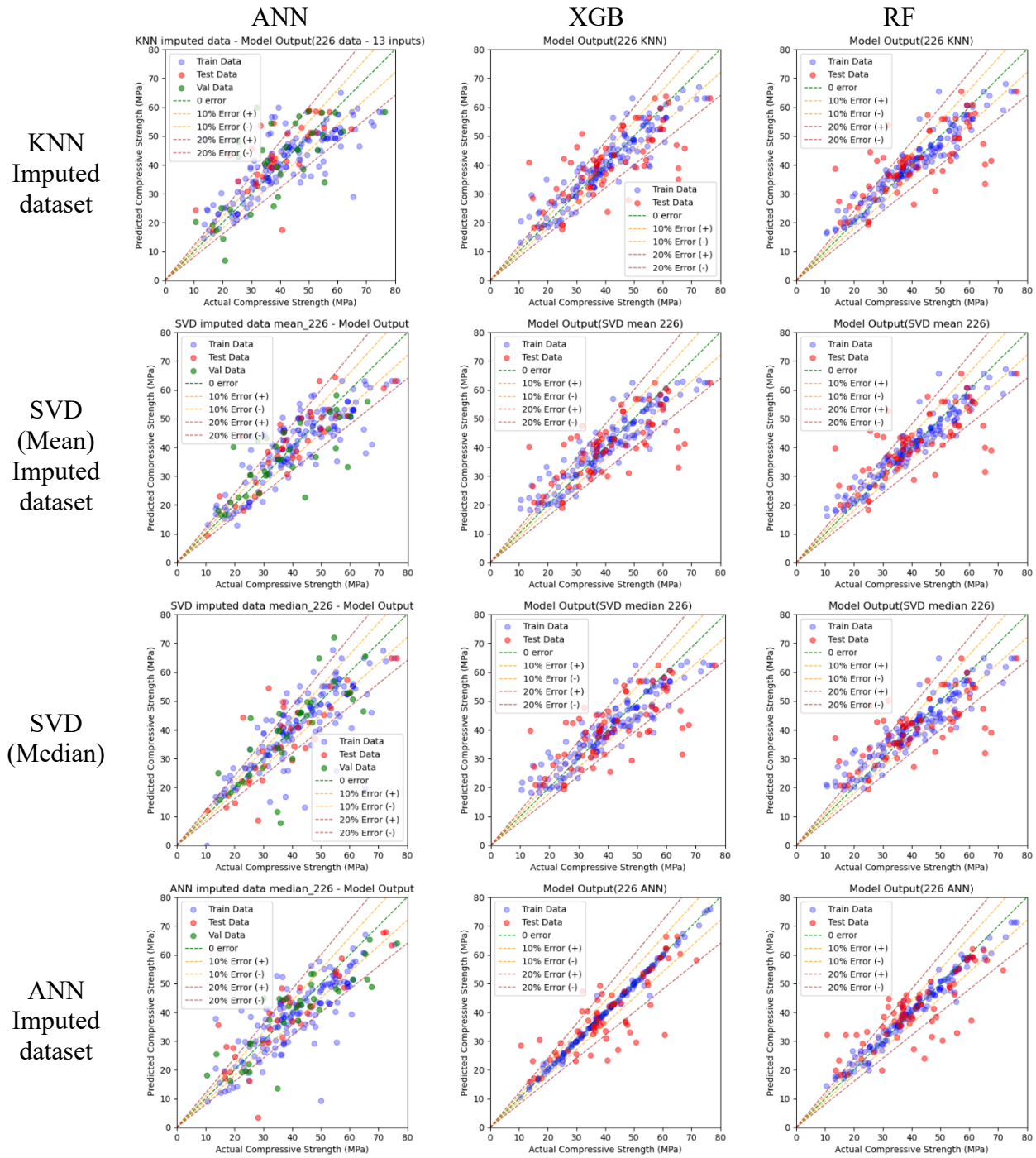
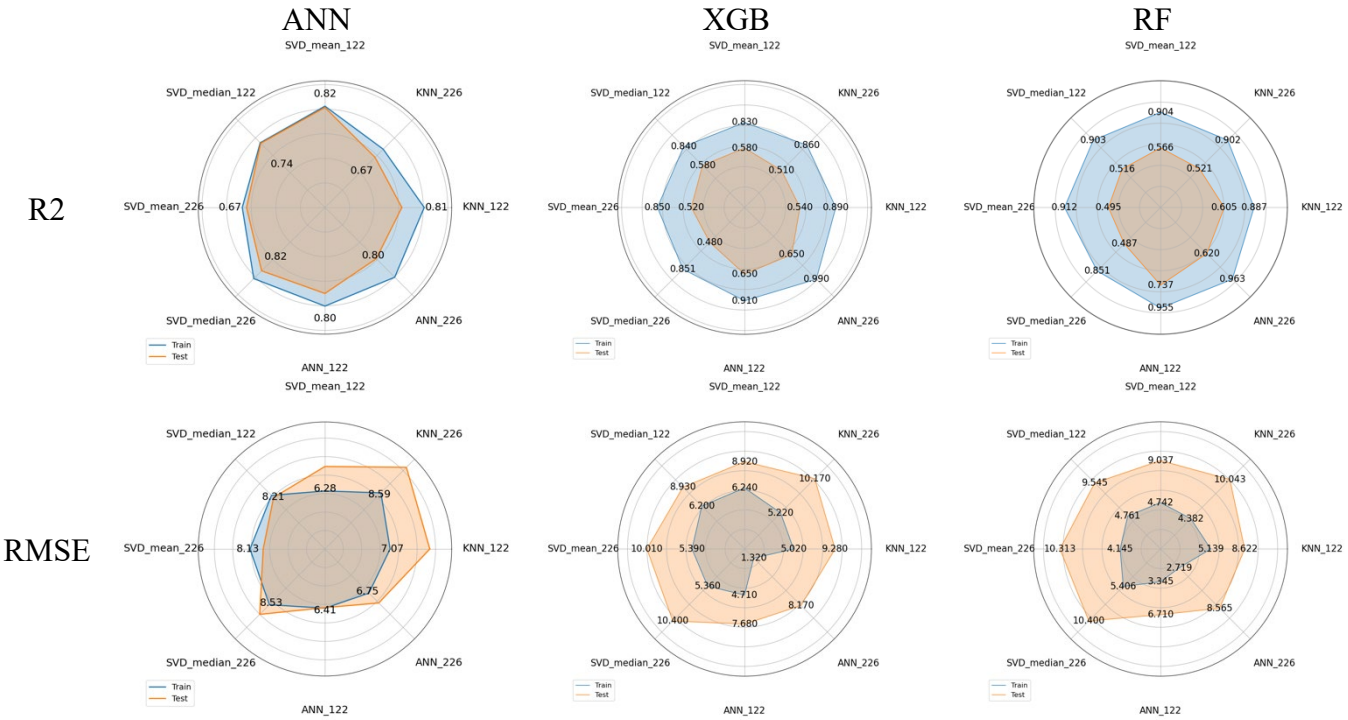


Figure 13 ML model performance matrix for step 2 data imputation

The performance of different data imputation methods for Step 1 and Step 2 Data Imputation, including KNN, SVD mean, SVD median, and ANN, was evaluated utilizing three machine learning models: ANN, XGB, and RF shown in Figure 14.

For the ANN model, Step 1 ANN imputation yielded the best performance, with an R^2 of 0.84 in the training set and 0.80 in the test set. The RMSE and MAE values were also the lowest in Step 1, demonstrating strong predictive power. While Step 2 showed slight improvements in some metrics for the test set, the overall performance did not surpass the results from Step 1, suggesting that ANN imputation (Step 1) was the most reliable method. A similar trend was observed in the XGB model, where Step 1 ANN imputation achieved an R^2 of 0.91 in the training set and 0.66 in the test set, along with the lowest error metrics (RMSE and MAE). In Step 2, despite improvements in the training metrics, the test set performance showed signs of overfitting, confirming that Step 1 provided better generalization.



MAE

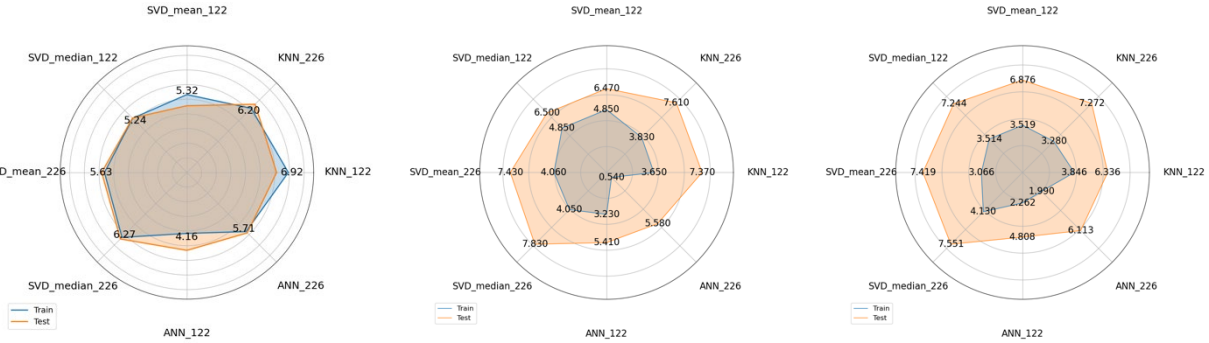
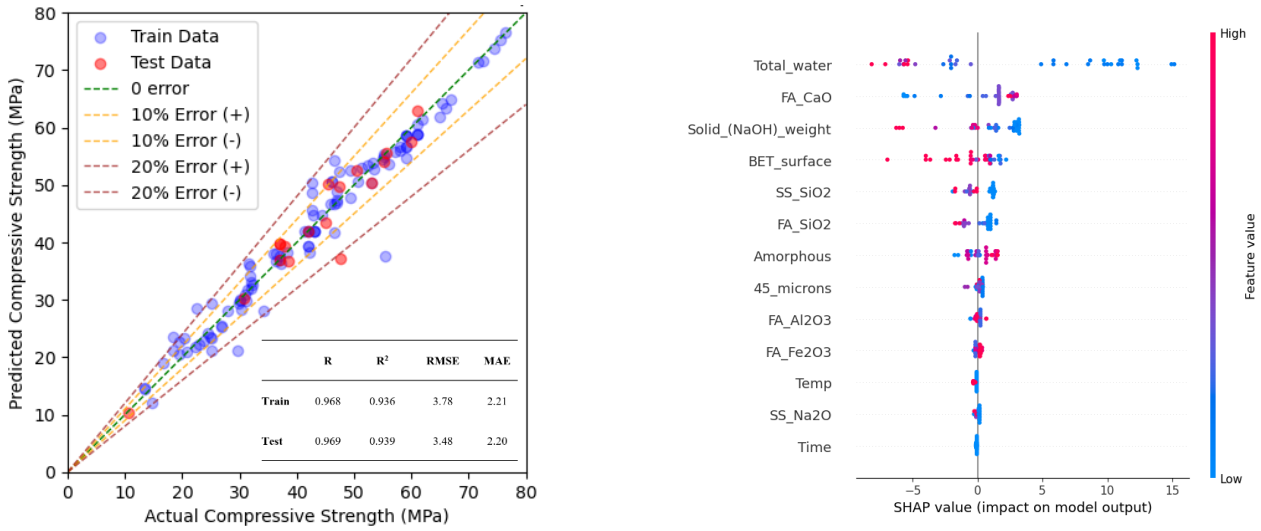


Figure 14 Model performance Step 1 and Step 2 data imputation

In the RF model, Step 1 ANN imputation again showed superior results, with the highest R^2 (0.95 in the training set and 0.74 in the test set) and the lowest RMSE and MAE values. Step 2, while improving training performance, exhibited a decline in test results, further indicating that ANN imputation from Step 1 was the optimal choice for the RF model as well.

In summary, ANN imputation from Step 1 consistently demonstrated the best balance between training accuracy and generalization to the test set across all models and overall highest in the ANN model.

The performance of the final ANN model is shown in Figure 15 (a) and the corresponding SHAP analysis is shown in Figure 15 (b). Based on performance the model predicts the compressive strength of GPC with an accuracy of 94%. In terms of actual compressive strength, the model with make predictions with an error less than 3.5 MPa.



(a) (b)

Figure 15 (a) Final ANN model performance (b) Model SHAP analysis

Based on the SHAP analysis, lower water content yields higher compressive strength, and vice versa. Furthermore, total water content had the greatest effect on the compressive strength of geopolymer concrete. These results align with the findings of Amran, et al. [20] and Hardjito, et al. [21]. This reduction in strength may be due to several factors. Firstly, an increase in water content can dilute the activator solution, directly influencing the dissolution rate of fly ash particles and the extent of the geopolymerization reaction. Lower concentrations of the activator result in slower reaction kinetics and lower final strength [22]. Additionally, excess water in the geopolymer mixture increases porosity as it evaporates during curing, leaving behind voids in the hardened concrete. This porosity significantly contributes to the reduction of compressive strength [23]. According to Provis and Bernal [24], excess water weakens the bond between fly ash particles and the geopolymer matrix. Without sufficient calcium, the water does not contribute to strength development and instead reduces the material's integrity.

The CaO content in fly ash is the second most influential parameter; as the CaO content increases, the formation of C-S-H (calcium silicate hydrate) also increases, which significantly improves the compressive strength of geopolymer concrete [25]. Furthermore, CaO can raise the alkalinity of the geopolymer mixture, leading to improved dissolution of silica and alumina from the fly ash, enhancing polymerization and strength development [26]. Additionally, the presence of CaO contributes to a denser microstructure in the geopolymer matrix, reducing porosity and improving load-bearing capacity [24]. The SHAP values for NaOH solid content indicate that up to a moderate NaOH concentration, compressive strength increases. However, higher NaOH solid contents have a negative effect on compressive strength. This could be because high concentrations of NaOH increase the alkalinity of the geopolymer mixture, which may result in excessive gel formation. This disrupts the balance between the gel and solid phases, weakening the structure [26]. Furthermore, at high NaOH concentrations, the reactivity of fly ash may cause agglomeration rather than a uniform distribution in the matrix, negatively

affecting load transfer between particles [27]. Regarding BET surface area, lower values up to a moderate range improve compressive strength. However, as the BET surface area increases, compressive strength decreases. A higher BET surface area indicates a greater number of fine particles, which can increase the water demand of the mixture. This higher water content can lead to increased porosity, thus reducing overall compressive strength. Additionally, effects such as increased porosity, reduced particle interaction, and accelerated reaction kinetics could contribute to the reduction of compressive strength in geopolymer concrete as the BET surface area increases [24, 27]

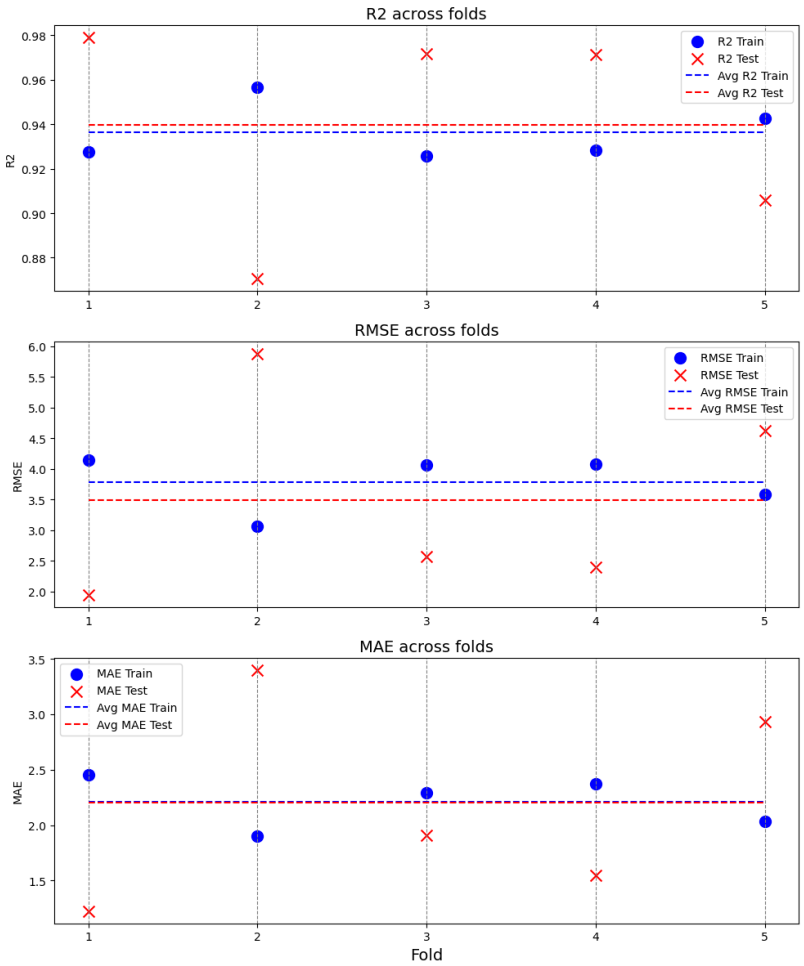


Figure 16 K-fold cross validation results for final model.

The results from the five-fold cross-validation provide a comprehensive overview of the performance of the predictive model across various training and testing subsets (Figure 16). In

terms of training performance, the R^2 values for the training sets are consistently high, ranging from 0.93 to 0.96 across the folds, with an average R^2 of 0.94. This indicates that the model deliniates approximately 94% of the variance in the training data, suggesting a strong fit. The RMSE values for the training sets vary between 3.06 and 4.14, with an average RMSE of 3.78. While these values are relatively low, indicating that the model predictions are close to the actual values, there is some variability across folds. Additionally, the MAE for the training sets range from 1.90 to 2.45, with an average of 2.21. This suggests that, on average, the model predictions have an inaccuracy of about 2.2 units, which is reasonably low and indicates effective predictive performance.

When analyzing the testing performance, the R^2 values for the test sets show more variability, ranging from 0.87 to 0.97, with an average of 0.94. The high average R^2 indicates that the model generalizes well to unseen data, but the lower value in Fold 2 (0.87) suggests that there may be challenges in some specific subsets of data. The RMSE values for the test sets range from 1.94 to 5.87, with an average RMSE of 3.48. The presence of a higher RMSE in Fold 2 indicates that there could be specific features or characteristics in the test data that the model struggled to predict accurately. Similarly, the MAE for the test sets ranges from 1.22 to 3.40, with an average of 2.20. This variability reflects differing prediction accuracy across the test folds, with some folds proving to be significantly more challenging for the model. In conclusion, the results indicate that the model generally performs, demonstrating a strong ability to predict compressive strength with acceptable error.

Summary and conclusion

1. The evaluation of imputation methods for geopolymer concrete datasets showed that ANN-based imputation outperformed other techniques for MNAR data. The ANN model achieved R^2 of 0.84 (train) and 0.80 (test), while RF performed best overall with R^2 of 0.95 (train) and 0.74 (test). XGB also benefitted from ANN imputation,

highlighting its superior capability for handling missing data compared to KNN and SVD methods.

2. SHAP analysis reveals that water content significantly enhances the compressive strength (CS) of GPC. It also emphasizes that the sensitivity of total water content to CS is extremely high, acting as a critical factor in determining CS, similar to the water-to-cement ratio in OPC concrete.
3. SHAP analysis further demonstrates that the presence of CaO in fly ash plays a crucial role in improving strength and reducing porosity. While moderate solid NaOH content can enhance CS, excessive amounts may disrupt the gel structure, weakening the concrete.
4. By incorporating the chemical and physical properties of fly ash, the compressive strength of geopolymer concrete can be predicted with 94% accuracy. Additionally, k-fold cross-validation provides a more reliable estimate of model performance by using multiple splits of the dataset, ensuring that performance metrics (e.g., R^2, RMSE, MAE) are not biased by a single train-test split.

4. References

[1] B. Zhang, "Durability of sustainable geopolymer concrete: a critical review," *Sustainable Materials and Technologies,* p. e00882, 2024.

[2] C. Luan, A. Zhou, Y. Li, D. Zou, P. Gao, and T. Liu, "CO2 avoidance cost of fly ash geopolymer concrete," *Construction and Building Materials,* vol. 416, p. 135193, 2024.

[3] P. Duxson and J. L. Provis, "Designing precursors for geopolymer cements," *Journal of the american ceramic society,* vol. 91, no. 12, pp. 3864-3869, 2008.

[4] D. Hardjito, S. E. Wallah, D. M. Sumajouw, and B. V. Rangan, "Factors influencing the compressive strength of fly ash-based geopolymer concrete," *Civil engineering dimension,* vol. 6, no. 2, pp. 88-93, 2004.

[5] J. Davidovits, *Geopolymer chemistry and applications*. Geopolymer Institute, 2008.

[6] X. Y. Zhuang *et al.*, "Fly ash-based geopolymer: clean production, properties and applications," *Journal of cleaner production,* vol. 125, pp. 253-267, 2016.

[7] M. T. Ghafoor, Q. S. Khan, A. U. Qazi, M. N. Sheikh, and M. Hadi, "Influence of alkaline activators on the mechanical properties of fly ash based geopolymer concrete cured at ambient temperature," *Construction and Building Materials,* vol. 273, p. 121752, 2021.

[8] Sindhunata, J. Van Deventer, G. Lukey, and H. Xu, "Effect of curing temperature and silicate concentration on fly-ash-based geopolymerization," *Industrial & Engineering Chemistry Research,* vol. 45, no. 10, pp. 3559-3568, 2006.

[9] L. N. Assi, E. E. Deaver, and P. Ziehl, "Effect of source and particle size distribution on the mechanical and microstructural properties of fly Ash-Based geopolymer concrete," *Construction and Building Materials,* vol. 167, pp. 372-380, 2018.

[10] C. Gunasekara. "Influence of properties of fly ash from different sources on the mix design and performance of geopolymer concrete " RMIT University. (accessed 2022-08-08, 2022).

[11] M. Rathnayaka, D. Karunasinghe, C. Gunasekara, K. Wijesundara, W. Lokuge, and D. W. Law, "Machine learning approaches to predict compressive strength of fly ash-based geopolymer concrete: A comprehensive review," *Construction and Building Materials,* vol. 419, p. 135519, 2024.

[12] J. Temuujin, A. Van Riessen, and R. Williams, "Influence of calcium compounds on the mechanical properties of fly ash geopolymer pastes," *Journal of hazardous materials,* vol. 167, no. 1-3, pp. 82-88, 2009.

[13] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen, "Methods for imputation of missing values in air quality data sets," *Atmospheric environment,* vol. 38, no. 18, pp. 2895-2907, 2004.

[14] G. E. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied artificial intelligence,* vol. 17, no. 5-6, pp. 519-533, 2003.

[15] D. B. Rubin, "Inference and missing data," *Biometrika,* vol. 63, no. 3, pp. 581-592, 1976.

[16] S. Van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in R," *Journal of statistical software,* vol. 45, pp. 1-67, 2011.

[17] G. A. Lyngdoh, M. Zaki, N. A. Krishnan, and S. Das, "Prediction of concrete strengths enabled by missing data imputation and interpretable machine learning," *Cement and Concrete Composites,* vol. 128, p. 104414, 2022.

[18] R. C. Pereira, P. H. Abreu, P. P. Rodrigues, and M. A. Figueiredo, "Imputation of data Missing Not at Random: Artificial generation and benchmark analysis," *Expert Systems with Applications,* vol. 249, p. 123654, 2024.

[19] S. J. Choudhury and N. R. Pal, "Imputation of missing data with neural networks for classification," *Knowledge-Based Systems,* vol. 182, p. 104838, 2019.

[20] Y. M. Amran, R. Alyousef, H. Alabduljabbar, and M. El-Zeadani, "Clean production and properties of geopolymer concrete; A review," *Journal of Cleaner Production,* vol. 251, p. 119679, 2020.

[21] D. Hardjito, S. E. Wallah, D. M. Sumajouw, and B. V. Rangan, "On the development of fly ash-based geopolymer concrete," *Materials Journal,* vol. 101, no. 6, pp. 467-472, 2004.

[22] H. Xu and J. Van Deventer, "The geopolymerisation of alumino-silicate minerals," *International journal of mineral processing,* vol. 59, no. 3, pp. 247-266, 2000.

[23] P. Nath and P. K. Sarker, "Effect of GGBFS on setting, workability and early strength properties of fly ash geopolymer concrete cured in ambient condition," *Construction and Building materials,* vol. 66, pp. 163-171, 2014.

[24] J. L. Provis and S. A. Bernal, "Geopolymers and related alkali-activated materials," *Annual Review of Materials Research,* vol. 44, no. 1, pp. 299-327, 2014.

[25] M. S. Reddy, P. Dinakar, and B. H. Rao, "A review of the influence of source material's oxide composition on the compressive strength of geopolymer concrete," *Microporous and Mesoporous Materials,* vol. 234, pp. 12-23, 2016.

[26] P. Duxson, A. Fernández-Jiménez, J. L. Provis, G. C. Lukey, A. Palomo, and J. S. van Deventer, "Geopolymer technology: the current state of the art," *Journal of materials science,* vol. 42, pp. 2917-2933, 2007.

[27] J. L. Provis and J. S. J. Van Deventer, *Geopolymers: structures, processing, properties and industrial applications*. Elsevier, 2009.

Appendix

KNN Pseudocode

Class KNNImputer:
  Method __init__(self, k):
    Input:
      k: Number of neighbors
    Initialize:
      self.k = k

  Method fit_transform(self, data):
    Input:
      data: Dataset with missing values
    Output:
      imputed_data: Dataset with imputed values

    Initialize:
      imputed_data = copy of data

    For each row in data:
      For each missing value in row:
        1. Identify rows without missing values in the same column.
        2. Calculate the distance between the current row and each row without missing values.
        3. Select the k rows with the smallest distances.
        4. Estimate the missing value using the mean of the k nearest neighbors.
        5. Replace the missing value in imputed_data with the estimated value.

    Return imputed_data

MICE Pseudocode

Class MICEImputer:
The MICEImputer class implements the MICE algorithm using linear regression to impute missing values.
Method __init__(self, num_iter)
    Input:
      - num_iter: Number of iterations (chained equations).
    Initialization:
      - self.num_iter = num_iter: Stores the number of iterations.
Method fit_transform(self, data)
    Input:
      - data: Dataset with missing values.

Output:
- imputed_data: Dataset with imputed values.

Steps:

3. Initialization:
   - Create a copy of the input data and store it in imputed_data.
   - Identify columns with missing values (missing_columns).
4. Multiple Iterations (chained equations):
   - For each iteration (from 1 to num_iter):
     - For each column with missing values (col in missing_columns):
       1. Split data into X (features) and y (target column col):
          - X consists of rows without missing values in col.
          - y consists of values in col for rows without missing values in col.
       2. Train a linear regression model:
          - Fit the model using X and y.
       3. Predict missing values:
          - For each row with missing value in col:
            - Use the trained model to predict the missing value based on other columns (X).
       4. Update imputed_data:
          - Replace missing values in col with predicted values.
5. Return the imputed_data with all missing values imputed after num_iter iterations.

SVD Pseudocode

Class SVDimputer:

The SVDimputer class implements the SVD imputation method.

Method __init__(self, rank)

- Input:
  - rank: Desired rank for the low-rank approximation.
- Initialization:
  - self.rank = rank: Stores the rank parameter.

Method fit_transform(self, data)

- Input:
  - data: Dataset with missing values.
- Output:
  - imputed_data: Dataset with imputed values.
- Steps:
  1. Initialization:
     - Create a copy of the input data and store it in imputed_data.
     - Identify rows and columns with missing values (missing_rows, missing_cols).
  2. Perform SVD:
     - Compute the SVD of the data matrix:
     -

Python code

U, S, Vt = svd(imputed_data)

3. Impute missing values:
    - For each missing entry (i, j):
        - Calculate the approximation using the low-rank components:

Python code
imputed_data[missing_rows[i], missing_cols[j]] = U[missing_rows[i],: self.rank] @ np.diag(S[:self.rank]) @ Vt[:self.rank, missing_cols[j]]

4. Return the imputed_data with all missing values imputed using SVD.

ANN Pseudocode

Class ANNImputer:

The ANNImputer class implements the ANN-based imputation method.

Method __init__(self, hidden_units, epochs, batch_size)

- Input:
    - hidden_units: List specifying the number of units in each hidden layer.
    - epochs: Number of training epochs.
    - batch_size: Batch size for training.
- Initialization:
    - Initialize ANN architecture with specified hidden layers.
    - Store training parameters (epochs, batch_size).

Method fit_transform(self, data)

- Input:
    - data: Dataset with missing values.
- Output:
    - imputed_data: Dataset with imputed values.
- Steps:
    1. Preprocess data:
        - Handle missing values (e.g., impute initial values, scale data).
        - Split data into features (X) and target (y) where y corresponds to columns with missing values.
    2. Build ANN model:
        - Initialize an ANN model with input layer matching X.shape[1] and hidden layers specified by hidden_units.
        - Output layer matches the number of columns with missing values.
    3. Train the model:
        - Train the ANN model using X and y for epochs with batch_size.
    4. Predict missing values:
        - Use the trained ANN model to predict missing values in data.
    5. Return the imputed_data with all missing values imputed using the trained ANN model.