Predicting Shear Strength In RC Deep Beams Using Machine Learning

Posted on March 21, 2025

Reinforced concrete deep beams are one of the key structural components that are essential in a variety of applications including bridges, tall buildings, and marine structures. They are capable of creating strut-like compression components between applied loads and supporting structures through their distinctive load and support setup, thus being effective load-distribution elements in various structures. According to ACI code 318-18 [1], beams are classified as deep beams for flexion, with the overall depth ratio of the single beams being less than 1.25, and the overall depth ratio of the continuous beams being more than 2.5. Furthermore, because the depth beam is high in shear capacity and the depth ratio (a/d 2.0) is low, it is commonly used in high-rise buildings, bridges, and naval structures. Their differences in behaviour, including high-stress points and uneven strain distributions, question the accuracy of Bernoulli's theory [5] [6]. RC deep-distance damage location is usually determined by empirical methods or thumb rules, while the strung-and-tie method offers an alternative approach [7], [8], and [9]. Most design codes of RC deep beams are based on empirical or semi-empiric formulas. However, the accuracy and reliability of these equations are limited by the scope of experimental data used for the evaluation.

Although these empirical methods tend to be cautious in the design process, they can lead to potentially dangerous results [10]. Collins et al. [11] Using extensive RC beam data, shear strength estimation techniques were studied for NZS 3101-2006, ACI 318-18 and EC2-1-2004. The results show that many beams are too optimistic and even consider safety margins. Furthermore, the contemporary design codes adopted a design methodology incorporating mechanical models of rods and cables. It is still difficult to estimate the shear capability of reinforced concrete beams, and some current models estimate conservative estimates, including safety factors. The complexity of failures and many factors that affect failures make it difficult to determine the ability of lay cutting. As a result, the curiosity for predicting the shear strength of RC deep-dish beams through machine learning (ML) methods is increasing and it has the potential to improve accuracy and effectiveness. In several research projects, machine learning technologies are used to predict the strength of reinforced concrete beams.

A major research project is exploring the use of artificial neural networks (ANNs) to predict the final cutting strength of RC deep beams. Scientists collected 111 experimental data sets, analyzed 10 input factors, including geometric and material properties, and calculated the final thickness. ANN performance is evaluated using traditional methods such as ACI, strut-and-tie, and Mau-Hsu. The results show that artificial neural networks (ANNs) surpass conventional techniques and are much more precise, with an average deviation strength of 0.99. Similarly, Wakjira and colleagues[13] have proposed a modified cut design formula for reinforced concrete pillars without a stirrup using machine

learning and genetic algorithms. The model considers factors such as concrete strength, cross-section size, angle ratio and internal reinforcement to estimate cut performance. The proposed model was compared with the ACI 318/14 method [14] and demonstrated better safety, accuracy and cost-effective prediction capabilities.

Although ML applications are not common in RC deep beams, Mohammadizadeh et al. [15] It has been studied to predict the use of adaptive neuroflash intuition (ANFIS) and meta-heuristic algorithms. The proposed model takes into account various factors, such as effective concrete depth, compression strength, beam length, and reinforcement ratio. The results showed that the ANFIS model and meta-heuristic algorithms are more accurate than other methods. Another research project proposed a combination model, the SVR-GA, for predicting the shear resistance of the deep RC beam [16]. The beams are analyzed by measuring, mechanical properties, and material characteristics. In the testing phase, the SVR-GA model showed higher determination coefficients (R2) than the traditional SVR, ANN, and gradient-building decision trees (GBDTs). In addition, Kaveh and his colleagues [17] created group machine-learning algorithms based on gradient-based decision trees to predict the final cutting capabilities of FRP beams without shocks. The proposed XGBoost model is more precise and general than experimental equations and other machine learning models. Almasabha et al. SFRC Deep Blade Shear Strength Experiments predict shear strength using ML models such as LightGel-Band Machine (LightGBM), XGBoost, and Genetic Expression Programming (GEP). LightGBM and XGBoost achieved high accuracy levels with R2 values of 97.25 and 94.66 per cent, respectively. Further research is needed to verify that the ML model is a reliable replacement for the mechanism model. It is necessary to conduct more research to understand model interpretation, especially in relation to the cutter mechanism. Thus, in order to improve the application of ML technologies, it is necessary to understand a clear methodology coupled with the cutting behaviour of RC deep beams, focusing on quantitative evaluation. The aim of the study is to fill the research gap by predicting the shear strength of RC deep beams using data and mechanism-based machine learning models. The outcome model is expected to provide more accurate forecasts and become a useful tool for comparable obstacles. This paper consists of: 1) explaining the research reasons and methods used, including the research of ML models and suggested interpretation techniques; 2) selecting input characteristics and summarising the experimental data; 3) assessing the results of the created ML models and finding the most effective; 4) analyzing the SHAP analyses and explaining the decision-making processes of the selected models and proposed interpretation methods.

2. Research Thought And Methodology

2.1. Research Thought

This study focuses on the construction of synergistic data and mechanism-driven approaches in order to develop a predictive model for estimating the cutting strength of RC deep beams. As shown in Figure 1, the first step is to create an experimental directory, which includes selecting features and

carefully compiling data. Selective input properties are closely related to the cutting mechanism of RC deep beams. After the database was established, the second phase consisted of training and evaluation of six different ML models. The aim is to identify models that show the best prediction accuracy. Subsequently, the performance of the mainline model was compared to five mechanism models, including Matamoros and Woo (19), NZS 3101-2006, EC2 1-2004, and Russo and Russo equations. [20] to verify its ability to predict the accuracy, generalization capabilities and impact of key input features. The following task is to ensure that the decisions and expected values are consistent with the cut mechanism. This was achieved by analyzing the meaning and dependence of SHAP's qualitative discussion results (Shapley Additive ExPlanations). In reviewing these aspects, it is intended to propose models with higher performance.

2.2. Machine Learning (Ml) Models

Machine learning (ML) is a sophisticated data analysis technique that automates the creation of analysis models. As part of artificial intelligence (AI), systems are based on a principle that can get data knowledge, recognize patterns, and make informed decisions with minimal human intervention. With the increase in data volume, the importance of intelligent data analysis is expected to increase, and it will play an important role in technological development and innovation [21]. The following six ML models were chosen to predict the shear strength of RC deep beams.

2.2.1. Gradient Boosting Decision Tree (GBDT)

Gradient-upgrading Decision Tree Algorithms (GBDTs) are widely recognized as high-performance models that produce accurate predictions by using unified decision trees and gradient-upgrading techniques [22]. Larestani et al. GBDT is similar to the functional gradient descending approach, and new learners solve each step residual errors in the previous learners and minimize certain loss functions. The GBDT ensemble structure enables the integration of several weak prediction models and often uses simple form decision trees or branches. The rank tree structure of GBDT offers additional interpretability advantages. It provides insights into the decision-making process of the model. It emphasizes the importance of different characteristics [22]. This technology uses a step-by-step format to efficiently build regression solutions while minimizing over-fit problems. In addition, GBDT optimization simplifies learning processes and transforms optimization problems into common optimization problems [23]. The iterative algorithm starts with a single leaf, improves the learning rate of each node and record, and updates the function iteratively to obtain an accurate prediction. GBDT algorithms are combined with the interpretation, staging, and optimization of GBDT algorithms, and are a powerful intuitive tool for generating accurate results for various applications.

2.2.2. eXtreme Gradient Boosting (XGBoost)

The introduction of eXtreme Gradient Boosting (XGBoost) by [24] in 2016 has since been recognized

as a leading predictor of current performance, both in regression and classification tasks.

The algorithm is based on traditional gradient enhancement algorithms and includes various improvements and regulation techniques to prevent over-incorporation. XGBoost's loss function combines loss function(Ψ*) to measure the difference between the predicted label and the actual label, and Ω (f) to punish each leaf node's number of leaves and output to discourage complex models.

The XGBoost loss function merges a loss function Ψ(*) to gauge the variance between predicted and actual labels with a regularization function Ω(f) aimed at discouraging complex models. The complexity control of each leaf, defined in Eq. 1 with T representing the leaf count, □ as the penalty parameter, and ‖□‖² for the leaf node output, is crucial. Unlike traditional gradient boosting, XGBoost utilizes the Taylor series in a second-order approach to convert objective functions into quadratic minimization problems.

Equation 2 shows a transformed objective function, where first and second-order gradient statistics are represented by gi and hi in the loss function, for the i-th sample respectively.

With the help of Ij, the objective function is simplified to represent the number of leaves.

In order to manage overfitting, XGBoost uses various techniques such as learning rate adjustment, enhancement of iteration control, limitation of maximum tree depth and sub-sampling training samples [25].

2.2.3. K-Nearest Neighbors (KNN)

The K-nearest neighbours (KNN) algorithm identifies the K training samples that are most similar to the target. The value of K signifies the number of nearest neighbours and is chosen based on their proximity to classify input features. By evaluating the closest neighbours, KNN groups similar training samples together. This algorithm relies on distance calculations and voting mechanisms to find the optimal K value for accurate classification [26].

2.2.4. Bagging

Bagging is a popular ensemble method [27] that involves generating multiple diverse training sets through bootstrap sampling techniques. Each of these training sets then aggregates the predictions from base learners to form the ultimate model. The key elements of bagging include bootstrap sampling and model aggregation. Bootstrap sampling entails randomly selecting n samples from the initial dataset of n samples, with replacement, to maintain independence among various training sets [25].

2.2.5. Random Forest Regression (RFR)

Leo Breiman introduced the random forest regression algorithm (RFR), a machine learning technology proficient in predicting extensive datasets. The RFR algorithm comprises four main processes: bootstrap resampling, random feature selection, out-of-bag error estimation, and the development of full-fledged decision trees. It generates multiple decision trees through bootstrap resampling, treating them as weak learners. Randomly chosen subsets of data are used to train each tree, while the remaining samples serve as out-of-bag samples for error estimation during training. Unlike pruned decision trees, random forest trees grow without pruning, leveraging new training samples to create robust prediction models [26]. Renowned for its versatility in handling large datasets and delivering accurate predictions, the RFR model finds widespread use across various machine learning applications.

2.2.6. Stacking

Stacking Ensemble Learning is an approach to the ability to fully familiarize the model based on the interconnected strengths of different base models introduced by Wolpert (28). The process is to obtain a prediction from various basic models, and then intelligently combine these predictions with meta-learners to reach a final prediction. In order to avoid overload, meta-learners do not learn directly from the output of the base model. On the contrary, the methodology of leave-one-out cross-validation is used in this approach of integrated learning. The validation folds are placed together and create a new set of data that the meta-learner can learn from.

The integration of the basic model plays an important role in stacking. The selection of meta-models depends on specific problems, but multiple linear regression ML models are often used as meta-models. The stacking model offers powerful techniques that leverage the different expertise of the base model and combine their prediction skillfully, resulting in improved prediction performance compared to the individual model.

2.3. Model Building

Python’s scikit-learn library is utilized to handle the data, which is split into training, testing, and validation sets with an 80-10-10 percentage split. Standard scalers are applied to scale the data, followed by a grid search to optimize hyperparameters. The training data is used to train the models while testing and validation are performed on unseen data.

In models like GBDT, Bagging, and XGBoost, decision trees serve as the base estimators. In contrast, the stacking model incorporates base models such as KNN, GBDT, and XGBoost, adjusting their hyperparameters via grid search. Once the best model is identified, they are adapted to scale the training data. The base model predictions are combined to form a new dataset, which becomes the

input for the metamodel—a linear regression model. Hyperparameter tuning is then carried out on the metamodel using grid search to optimize its performance. After obtaining the best meta-models, they are applied to the combined dataset of base model predictions and actual target values, resulting in the final stacked model.

2.4. Evaluation Metrics

Assessing the accuracy and efficacy of machine learning models relies significantly on evaluation metrics, offering essential insights into how well the models predict target variables. These metrics serve as valuable tools for comparing models and making informed decisions about their performance. In this context, five evaluation metrics were employed to gauge the effectiveness of the ML models:

2.4.1. Coefficient of determination (R^2)

The coefficient of determination assesses how much of the variability in dependent variables can be explained by independent variables within a model, as depicted in Eq. 4. Ranging from 0 to 1, a value of 1 signifies that the model accurately predicts the target, while 0 suggests that the model does not enhance predictions beyond the average.

2.4.2. Root mean square error (RMSE)

Root Mean Square Error (RMSE) measures the average magnitude of a dataset's prediction and actual value inconsistency and is widely used in regression models in statistics and ML. Equation. 5 shows the formula for RMSE.

2.4.3. Mean absolute error (MAE)

The mean absolute error (MAE), illustrated in Eq. 6, measures the average absolute difference between actual and predicted values. Unlike the mean squared error (MSE), it does not square the errors, making it less sensitive to outliers.

2.4.4. Median absolute error (MdAE)

It is the average value of the absolute difference between the actual target value and the predicted value. As shown in Eq. 7, it resists deviations and provides a central measurement of the central tendency of errors.

2.4.5. Mean squared logarithmic error (MSLE)

The mean squared logarithmic error (MSLE), presented in Eq. 8, is a modified version of the mean squared error that involves taking the logarithm of both predicted and actual values before computing the squared differences. This metric is often used when the target values cover a wide range, and evaluating the model's performance on a logarithmic scale is more appropriate. In the equation, yi represents the actual value, yi' stands for the predicted value, μ denotes the mean value, and n signifies the number of samples.

3. Data Collection And Preprocessing

3.1. Experimental Data Collection

Figure 2 illustrates a schematic depiction of the shear mechanism in reinforced concrete (RC) deep beams. Extensive experimental and theoretical investigations [9], [10], [30], [31], [32] have revealed that the shear strength is influenced by various components, primarily the concrete quality and the web reinforcement. Key factors affecting shear strength include the compressive strength of concrete (fc'), the shear-span ratio of the beam (a/d), the strength and proportion of longitudinal reinforcement (fy and ρl), and the ratio of vertical and horizontal reinforcement in the web (ρv and ρh). Additionally, the characteristics of both vertical and horizontal web reinforcement significantly impact the web reinforcement section. This study has identified critical parameters related to shear strength, encompassing concrete compression force, beam shear-span ratio, longitudinal reinforcement properties, and web reinforcement ratios.

The dataset consists of 587 data points sourced from previous researches [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54]. Table 1 provides a summary of the crucial parameters related to shear strength used in this investigation, including their respective units of measurement and statistical details. These key parameters serve as input features for the model, with cutting force (νexp) representing the corresponding output. Furthermore, Figure 3 presents a histogram of all six input characteristics within the dataset.

Table 2 presents the statistical distribution of both input and output parameters, offering insights into the central tendencies, variability, and ranges within the dataset. The average value of the shear-span ratio (a/d) is approximately 1.25, with a minimum of 0.23 and a maximum of 2.50. The standard deviation of 0.50 suggests a moderate level of variability around the mean. For the compressive strength of concrete (fc'), the average value is around 31.58 MPa, with a minimum of 11.7 MPa and a maximum of 89.4 MPa. The standard deviation of 15.29 indicates variation in the compression force values. The mean yield strength (fy) in the dataset is approximately 428.2 MPa, ranging from 100.00 MPa to 1000.00 MPa, with a standard deviation of 102.74 indicating a significant range of yield

strength values. Similarly, the statistical distribution of other parameters such as longitudinal reinforcement ratio (ρl), vertical web reinforcement ratio (ρv), horizontal web reinforcement ratio (ρh), and cutting force (νexp) reflects the range of each parameter.

3.2. Data Preprocessing

3.2.1. Data normalization

As the data of multiple attributes shows different ranges, they need to be standardized [55] to allow meaningful comparison and analysis. Thus, the data sets are normalized using the z-score method. Each data point is converted to the corresponding z score that represents the standard deviation from the average. This transformation ensures that the standardized data have a standard deviation of 0 and 1 regardless of the original scale of the attributes. The standard scaler equation in sci-kit-learn is as follows:

X is the original value of the data point, μ is the average value of the data set, σ is the standard deviation of the data set, and is the standard value (z score) of the data point.

3.2.2. Hyperparameters

Hyperparameters are fundamental elements in the performance of machine learning algorithms, especially in supervised learning. According to Alhakeem et al. [56], Correct selection of hyperparameter values is crucial because it has a major impact on the performance of the model. When the appropriate value is selected for these hyperparameters, the performance of the model can improve significantly. Similarly, Angoros and Mukti [57] point out that the optimal performance of algorithms is highly dependent on hyperparameters in supervised learning.

Grid search is a complete search technique that analyzes the subsets of the hyperparameters, specifying lower and upper limits, and determining the number of steps between them. This method systematically evaluates all possible combinations in a defined grid to find the optimal hyperparameter values. By thoroughly analysing each grid, grid search identifies the most effective configuration. This method is beneficial for data analysis with high accuracy [57]. Another study [58], in which different high-parameter search methods were compared, consistently showed that grid search had higher performance than other techniques for their respective data sets. This observation highlights the effectiveness of grid search in identifying optimal hyperparameter configurations that improve model accuracy and generalization. Because of the advantages of using grid searches, grid searches were used to identify the best hyperparameter configuration for each ML model. Table 3 shows the best hyperparameter obtained in the grid search of each ML model. By optimizing these hyperparameters through grid search, the model was optimized to improve performance and accuracy.

3.3. Shap Feature Selection

Feature selection is essential in ML to reduce dimension to prevent over-fitting of high-dimensional data. Minimizing characteristics improves computational efficiency and minimizes the impact of irrelevant characteristics [59]. The positive results include improved predictive accuracy and a more interpreted learning result due to the reduction in dimensionality [60]. In this study, the SHAP methodology was used to analyze both the selection of characteristics and the dependencies of characteristics. Lundberg and Lee's SHAP methodology (61), based on game theory and the Shapely value, is easy to understand. The traditional method usually relies on the comparison of target variables and feature sets to calculate correlation coefficients such as the Pearson correlation coefficient, but SHAP uses a distinct method to assess the importance of features. It defines the significance of various feature classes by averaging the absolute magnitude of the importance of each characteristic in each sample. This alternative methodology offers a comprehensive view of the importance of features and highlights the collective impact of features on the predictions for each data point. Importantly, it transcends the limitations of traditional correlation-based techniques and encapsulates the interaction and dependency of complex features [62].

4. Results And Discussion

4.1. Performance Evaluation Of MI Models

Table 4 compares the performance of six prominent ML models, namely RFR, GBDT, KNN, Bagging, XGBoost, and Stacking. Comparisons were made using evaluation metrics such as R2, RMSE, MAE, MdAE, and MSLE. In a comprehensive assessment of six ML models, stacking was the most robust choice, especially considering key performance metrics. RFR and Bagging demonstrated excellent performance with high R2 values in all datasets, showing a strong overall fit. However, both models show a slight increase in error measurements, such as RMSE and MAE, in the validation set, suggesting potential challenges for generalization. Although GBDT and XGBoost performed exceptionally well in training sets with high R2 values, errors in test sets increased considerably. This involves the risk of over-adaptation, in which the model may be too personalized to training data and limits the ability to generalize to new invisible data. KNN has been successful in tests, especially with low values of RMSE, MAE and MdAE. However, the slight increase in the error level in the validation set indicates limitations in its generalization capabilities. In contrast, stacking consistently shows high R2 values in all data sets, reflecting its superior ability to capture data variation. In the test group, stacking allowed alow RMSE of 9.248, an MAE of 7.006, an aMdAE of 6.339, and an ample of 0.003.. These values highlight the accuracy and reliability of stacking predictions. The slight increase in error metrics in the validation set is a compromise, but the overall performance and generalization of stacking exceed this concern.

The strength of stacking lies in its ensemble approach, which effectively combines the strengths of

different base models. This is evident in the consistently low MSLE in all datasets, showing its ability to handle errors and uncertainties. Stacking is adaptable to different aspects of data and is preferred for this particular dataset and predictive tasks. Furthermore, the visual representations of the comparison of prediction (pred) and experimental (exp) values are shown in Figure 4. In this illustration, the diagonal line (y = x) represents the case in which the prediction values coincide precisely with the experimental values. In particular, the upper part of the diagonal shows cases where the predicted value exceeds the experimental value, while the lower part shows cases where the predicted value is below the experimental value. The fitting lines of the ensemble models encompassing GBDT, KNN and Stacking show significant improvements over RFR, Bagging, and XGBoost. Specifically, in Figure 4 (a), the visual representation of the RFR model shows relatively poor performance compared to other models. In contrast, Figure 4 (f) shows that the stacking model is superior to the rest, showing the highest alignment between the predicted value and the experimental value. Visual analysis of fitting lines emphasizes the effectiveness of the ensemble model with Stacking distinguished as the most qualified in accurately predicting experimental values.

Finally, stacking has proved to be the best model, offering a harmonious mixture of accuracy, resiliency against the outliers and generalization capabilities. Although further improvements through normalization techniques and hyperparameter adjustments can improve its performance, stacking's overall strengths make it the best choice for prediction of this dataset.

4.2. Comparison With Mechanism Models

This section compares the performance of the stacking model with five established mechanism models incorporating design codes such as ACI 318-18 (1, NZS 3101-2006, 2), EC2-1-2004 (3) and the equations proposed by Matamoros and Wang (19), and Russo. [20]. The simplified expressions of the five model models are described in detail in Appendix A, which analyzes the accuracy of prediction and the effect of these models on key features by using the prediction value ratio of the experimental value (=pred/exp) as the evaluation metric. A value below 1.0000 indicates a conservative prediction and a value greater than 1.0000 indicates an over-prediction. If is close to 1.0000, it means a small prediction error, which indicates a good prediction accuracy.

Table 5 provides statistical information about the X value and provides insight into the prediction capability of the stacking model and the five mechanism models. The average χ value obtained by the stacking model is 1.0002 and close to 1.000. In addition, the standard deviation value of stacking is 0.025, indicating the stability and robustness of predictive precision. Fig. 55 also gives statistical detail of the χ value across the entire database, showing the dispersion of χ from the Stacking model and the five mechanism models.EC2-1-2004 and the equations proposed by Russo et al., Matamoros and Wong showed similar results characterized by right-skewed χ value distributions and a tendency towards conservative predictions(Figure 5c-e). For example, in ACI 318-18, the average, standard deviation, and variation coefficient of χ values were 1.475, 0.44, and 30.43% respectively, suggesting greater prediction accuracy, but greater variability(Figure 5 (a).NZS 3101-2006 shows results almost

identical to those of ACI318-18 as shown in Figure 5 (b). In contrast, as shown in Figure 5 (f), χ values of the stacking model cluster around 1,000, with mean, standard deviation, and variation coefficients of 1.0002, 0.025, and 2.52% respectively, indicating superior results among these models. These comparisons provide convincing evidence of the superiority of stacking models in terms of accuracy and stability of prediction.

In addition, Figure 6-9 shows the effect of key features such as a/d, fc', fy, and ρl on the prediction efficiency of the entire database. Figure 6 shows the upward trend in χ values with the increase of a/d for ACI 318-18, NZS 3101-2006, EC2 1-2004, and the equations proposed by Russo et al., Matamoros and Wong indicating the influence of a/d on prediction accuracy. When a/d is less than 1.00, these models usually show a conservative prediction tendency, whereas they tend to display overprediction, as the a/d ratio varies from 1.00 to 2.50. However, the stacking model does not show an obvious trend in its χ distribution (Fig. 6 (f)), reflecting a consistent correlation between the shear strength and a/d. Similarly, the trend to increase the χ value with fc' was observed in all five mechanisms models, indicating that fc' influenced the accuracy of the prediction of the mechanisms models. On the other hand, changes in values did not affect the prediction accuracy of the stacking model, as shown in Figure 7 (f). Figs 8 and 9 examine the influence of fy and l on the prediction accuracy of the RC deep beam. It is clear that, in the five model models of mechanisms, the distribution of χ shows rising trends when values fy and l increase. On the other hand, the prediction accuracy of the stacking model is almost unchanged by fy and l, as shown by the consistent distribution of χ around 1.00 in Figures 8 and 9.

Overall, these evaluation metrics show a significant improvement in the performance of the stacking model, characterized by fast calculations, higher prediction accuracy, and strong generalization capability. The stacking model demonstrates a lower sensitivity to important variables and correctly captures the relationship between shear strength and these key factors. However, these conclusions alone are not sufficient to establish a stacking model that replaces the mechanism model.

5. Feature Importance And Dependency Analysis

5.1. Feature Importance Analysis

Feature importance is a method used to evaluate the importance of input features to predict target variables. The assigning points to input features offers useful insights into how they contribute to the prediction model [63]. This analysis helps to understand the characteristics most relevant to making accurate predictions of target variables, as well as help to select characteristics and improve the model's interpretation. Various methods can be used to achieve interpretation. One of them is the SHAP model approach. SHAP is SHapley Additive exPlanations. To understand how the main model handles this case, it creates a substitute model in its local neighbourhood. It isolates a single instance [64]. The insight into the model's decision-making process in specific cases is obtained through this local approach, thereby improving the model's overall interpretability.

Figure 10 shows the relative importance of input characteristics for RC deep beams. The input parameter a/d is the most important, and its relative importance is 30.83%. This emphasizes the importance of geometric configuration in determining shear strength, emphasizing that the changes in the a/d ratio can have a significant impact on model predictions. The next in terms of importance is fc'. This emphasizes the crucial role that concrete material's inherent strength plays in resisting the shear forces on the beam. The results emphasize the need to carefully consider the mixture of concrete and quality control to achieve the required shear strength results in RC deep beams. fy has the third highest relative importance, highlighting the importance of the material properties of longitudinal reinforcement to influence the shear strength prediction. The strength of the reinforcement material contributes significantly to the overall structural integrity and provides resistance to shear loads. The longitudinal reinforcement ratio ρl is another influential factor. This parameter highlights the importance of appropriate distribution and quantity of longitudinal reinforcement in relation to the dimensions of the beam. The achievement of an optimal balance is crucial for improving shear strength and overall structural performance.

When it comes to the ratio of web reinforcement,ρh andρv show lower relative importance values. Although these ratios contribute to the predictions of the model, their influence is relatively small. This indicates that horizontal and vertical distributions of wet reinforcements have a modest effect on the results of cutting strength.

In summary, a comparative analysis of the importance of SHAP features highlights the complex nature of variables inducing the shear strength of RC deep beams. Although geometric considerations, longitudinal strengthening properties and concrete strength play an important role, the contribution of web reinforcement ratios is relatively lower. This precise understanding can guide engineers to prioritize design considerations and optimizations to achieve the desired shear strength performance in RC depth beam structures.

5.2. Feature Dependency Analysis

Feature dependence analysis is a technique that is used to determine the co-relation among input features and the influence they exhibit on predicting the shear strength of RC deep beams. By reviewing how variations in these characteristics affect the prediction of the model which aismeasured using SHAP value, it can be determined which characteristics are critical to the accurate prediction of shear strength.

As mentioned in Section 3.3, the feature dependency diagram shows the relationship between the SHAP value on the Y axis and the feature value on the X axis. The objective of this research is to study the practicality of the decision-making basis of the stacking model by examining the selected key features, namely fc', a/d, ρ_l and fy, determined from the previous assessment of feature importance.

6. Conclusions

The study aimed to develop predictive models to estimate the ultimate shear strength of the RC deep beam using harmonious data and mechanism-based methods. The research methodology included creating an experimental database containing 587 RC deep beams, training and evaluating six ML models, and validating the model selected against a mechanism-based model and design code. The key conclusions and conclusions of the research are summarized as follows:

Amongst the 6 machine learning models (XGBoost, Bagging, GBDT, KNN, RFR, Stacking and RFR), the Stacking model unfolded as the most vigorous and accurate estimator for shear strength in RC deep beams. It showcased superior performance in terms of R^2, *MAE*, *MdAE*, *MdAE* and *MSLE*, showcasing its capability to supervise intricate connections and generalise greatly to new data.

Five mechanism models were compared to stacking models, based on the basis of equations and codes. These mechanism models were consistently outperformed by the stacking model in prediction, showcasing higher stability, accuracy and lower sensitivity to changes in vital input variables such as compressive strength of concrete (f_c'), beam shear-span ratio (a/d), longitudinal reinforcement ratio (ρ_l) and longitudinal reinforcement strength (f_y).

It was deducted via feature importance analysis that the most influential parameters were geometric variables longitudinal reinforcement properties (f_y, ρ_l), (a/d) and concrete strength (f_c'). ρ_h and ρ_v had lesser influence as compared to other parameters. Their findings were further supported by feature importance analysis, showcasing the influence of each parameter on the prediction of shear strength.

A balanced combination of precision, robustness against outliers and ability to generalization was exhibited by stacking model. It has been shown in this study to be the optimal choice for predicting the shear strength of RC deep beams. However, further improvements and hyper-parameter tuning can improve its performance. This model provides engineers with an invaluable tool to optimize designs and predict the shear strength precisely.

In conclusion, the stacking ensemble model incorporating machine learning techniques demonstrated impressive performance in predicting the shear strength of RC deep beams. The findings urge ML modelling to be integrated as an additional tool in structural engineering practices, thereby improving accuracy and understanding for knowledge decision-making in the design and evaluation of RC deep beam structures.