An Explainable Prediction Framework for Engineering Problems: Case Studies in Reinforced Concrete Members Modeling

Amirhessam Tahmassebi[a], Mehrtash Motamedi[b], Amir H. Alavi[c], Amir H. Gandomi[d,*]

[a]*Department of Scientific Computing, Florida State University, Tallahassee, FL, USA*
[b]*Department of Civil Engineering, University of British Columbia, Vancouver, Canada*
[c]*Department of Civil and Environmental Engineering, University of Pittsburgh, Pittsburgh, PA, USA*
[d]*Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, Australia*

Abstract

Purpose

Engineering design and operational decisions depend largely on deep understanding of applications that requires assumptions for simplification of the problems in order to find proper solutions. Cutting-edge machine learning algorithms can be used as one of the emerging tools to simplify this process, while not all the engineers have sufficient grasp of the machine learning principles. In this paper, we have proposed a novel scalable and interpretable machine learning framework to automate this process and fill the current gap.

Design/methodology/approach

The essential principles of the proposed pipeline are mainly (1) scalability, (2) interpretibility, and (3) robust probabilistic performance across engineering problems. The lack of interpretibility of complex machine learning models prevents their use in various problems including engineering computation assessments. Many consumers of machine learning models would not trust the results if they cannot understand the method. Thus, the SHAP Additive exPlanations (SHAP)

*Corresponding author
Email addresses: atahmassebi@fsu.edu (Amirhessam Tahmassebi), mmotamedi@civil.ubc.ca (Mehrtash Motamedi), alavi@pitt.edu (Amir H. Alavi), gandomi@uts.edu.au (Amir H. Gandomi)

approach is employed to interpret the developed machine learning models.

Findings

The proposed framework can be applied to a variety of engineering problems including seismic damage assessment of structures. The performance of the proposed framework is investigated using two case studies of failure identification in reinforcement concrete (RC) columns and shear walls. In addition, the reproducibility, reliability, and generalizability of the results were validated and the results of the framework were compared to the benchmark studies. Clearly, the results of the proposed framework outperformed the benchmark results with high statistical significance.

Originality/value

Although, the current study reveals that the geometric input features and reinforcement indices are the most important variables in failure modes detection, better model can be achieved with employing more robust strategies to establish proper database to decrease the errors in some of the failure modes identification.

Keywords: Explainable Machine Learning, Automated Framework, Gradient Boosting, Failures in RC Member

---

1. Introduction

Engineering design and operational decisions depend largely on engineers' understanding of applications that requires assumptions for simplification of the problems in order to find solutions. The simplification process is often done with combination of computational methodologies, engineering resources, and field data. Adding robust optimization techniques and machine learning algorithms to the current equation boosts the level of overall accuracy in decision-making and design performance improvement to solve challenging engineering problems and explore interpretibility of the solutions [1]. Machine learning as an applied scientific discipline has numerous advantages in real-world engineering problems and applied sciences; the most fundamental its advantage is that a machine

learning algorithm can learn from empirical data where modeled phenomena are hidden, non-evident, or not very well explained. Machine learning algorithms in civil engineering first used for testing different existing tools on simple problems and gradually were applied to harder problems. Recently, numerous studies show that universal nonlinear machine learning algorithms including artificial neural networks, fuzzy logic, support vector machine, decision trees, and random forests can be used as adaptive tools for solving complex practical classification and regression problems in engineering along with general properties of statistical learning from data and the mathematical theory of generalization from experience [2, 3].

Emerging as one of the most contemporary machine learning techniques, gradient boosting has shown success in various areas including stock price prediction [4], traffic speed forecast [5], Alzheimer diagnosis [6], and health monitoring systems [7]. In addition to this, gradient boosting has recently shown promising use in several engineering problems such as automatic detection of cracks from concrete surface [8], structural damage assessment for proper maintenance [9, 10], prediction of undrained shear strength [11], and safety evaluation of steel trusses [12] which opens new avenue in modeling engineering problems including seismic damage assessment of structures. The boosting principles and weak learners for the first time was proposed by Schapire [13] in 1990. Changing the distribution of the training iteratively is the main idea of boosting algorithms. This principle helps to bias the training process towards the specimens that are harder to classify. At each iteration, the boosting algorithm assigns a weight to each training instance. At the end of each boosting round, the assigned weights are getting updated adaptively. Thus, various bootstraps can be chosen from the original training set via the updated weights where play an important role as a sampling distribution. This is the main principle of the base classifiers. Gradient tree boosting, also known as gradient boosting machine (GBM) or gradient boosted regression tree (GBRT) was originally proposed by Breiman and elaborated by Friedman in 2000 [14]. The key principle of the boosting algorithms is to use some variants of weak (base) learners in a bounded size.

The most common types of weak learners for gradient tree boosting models are decision trees which the prediction error can be updated with slight modification of the weights at each round.

In this paper, we have employed one of the invariants of the boosting algorithm, named as XGBoost which is short for eXtreme Gradient Boosting [15]. XGBoost uses some modifications with respect to its predecessors such as sparsity-aware split finding, weighted quantile sketch, and parallel structure which makes this algorithm scalable which can be able to be used on high performance computing. To recapitulate, a gradient boosting algorithm, first optimizes the loss function, makes the weak learner to predict the exemplars, and uses an additive model to add weak learners to minimize the loss function. The type of loss function can be chosen based on the type of problem and use-case. For example, a squared error can be a good choice for regression problems while a logarithmic loss for classification problems as we have shown in section 3. Moreover, XGBoost includes extra implementations for the constraints that are applied on the additive model. The main benefit of employing the decision tree as an additive model is the number of degrees of freedom we would have in terms of hyper-parameters. In better words, the additive model can be changed by increasing/decreasing the number of trees (estimators), number of leaves or terminal nodes, the depth of tree or number of observations per split. In addition to this, other objectives can be chosen based on the learning quality can be applied including minimum improvement to loss, L_1 (mean-absolute-error as regularize) and L_2 (mean-squared-error as regularize) weights (the value at each leaves) regularization which would results in huge improvement in the results in comparison to classical machine learning models.

Many consumers of machine learning models will not trust the results if they cannot understand the method. While the mechanism and math of a black-box model is still a difficult concept to grasp, we hope that supplementing interpretability will go a long way in fostering trust in these methods. The basic idea of interpretability comes from the simplicity of the model: the simpler model, the more explainable. For complex models such as ensemble methods

[16] including gradient boosting, simplicity is not attainable. We can name two important elements of a complex model: (1) the information that can be algorithmically extracted, and (2) the noise [17]. The complex models often perform significantly better than the parametric models (in terms of prediction) and have achieved tremendous success in applications across many fields [18]. Cutting edge methods involving complex models have the ability to significantly improve outcomes, however the trade-off between accuracy and interpretability is a significant challenge in the field of machine learning. Hastie et al. [17] has shown multiples ways, including Friedman's partial dependence plot [19] and Pearl's back-door adjustment [20], to determine causal interpretation of black-box models. Therefore, an approach is needed to replace the complex model with an interpretable approximation of the original model.

There are several approaches to improve the explainability of a model: (1) LIME [21], (2) DeepLIFT [22], (3) Layer-Wise Relevance Propagation [23], (4) Shapley Regression Values [24], and (5) Shapley Sampling Values [25]. SHAP is a unified approach created to explain the output of any machine learning model through connecting game theory with local explanations. SHAP unifies several of the previous methods and presents the only possible consistent and locally accurate additive feature attribution method based on expectations [26, 27]. There is a myth between gradient definition (how y changes as x changes at the point x) and slope definition (how y changes as x differs from the baseline). SHAP implies that what is important here is the slope rather the gradient.

In this paper, the details of the modeling design and interpretation framework are explained in section 2. Then, section 3 presents the performance of the proposed framework using two case studies. Last, the summary and conclusions of the current study are presented in section 4.

2. Modeling & Interpretation Framework

Modeling and interpretation framework includes three main components: (1) training, (2) testing, and (3) generalization. Fig. 1 illustrates the flowchart

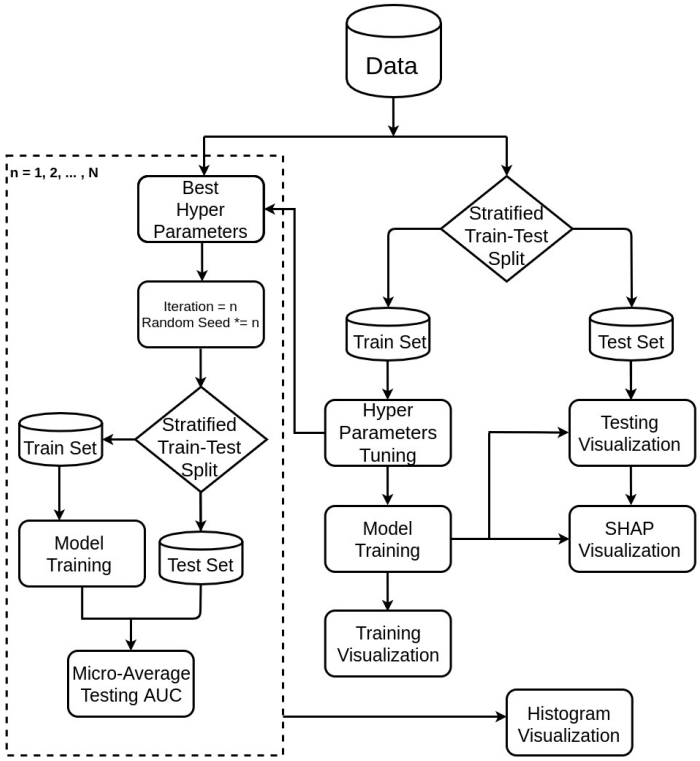


Figure 1: Modeling and interpretation framework flowchart.

of the framework. As shown, the pipeline begins with loading the database along with data preprocessing and feature encoding (no feature standardization/scaling). The database is splitted into train/test sets in a stratified fashion. This would a be crucial task to improve the generalization results due to any imbalanced classification problem. In the training step, the feature standardization/scaling can be fitted to the train data and scaler object should be applied to the test set to transform the test data into the scaled train data subspace (no fitting for test data). Next, the hyper-parameters of the XGBoost model should be tuned. Fail to tune the hyper-parameter values is one of the most common reasons for training a biased/over-fitted model.

There are various methods including exhaustive grid-search, random search, and Bayesian optimization [28]. Grid search provides an exhaustive search over

specified parameter values. All the possible combinations of the specified hyper-parameters will be checked. In contrast to grid search, the main idea of random search is not all the hyper-parameters values play an important role in the prediction results. Therefore, tying out a fixed number of parameter settings can be sampled from the specific distributions. Therefore, by random sampling, the most important hyper-parameters can be determined and the other hyper-parameters settings can be kept fixed. The learning slope will be positive while the similar outcomes would not be replicated. Checking all the combinations of the hyper-parameters also requires large enough computational time where designing a pipeline to predict what combinations are likely to work well using machine learning methodologies could help to rescue from this challenge. This can be done by predicting the regions of the hyper-parameter space that might give better outcomes and calculating the uncertainty of that prediction using Gaussian Process models for each new combination of hyper-parameters. Gaussian processes provide a simple, principled, practical, and probabilistic approach in machine learning with an essential assumption that similar inputs give similar outputs. This simple and weak prior is actually very sensible for the effects of hyper-parameters. Bayesian optimization, is a constrained global optimization approach built upon Bayesian inference and Gaussian process models to find the maximum value of an unknown function in the most efficient ways (less iterations) [29, 30].

After tuning the hyper-parameters, the model can be trained and the fitness metrics can be evaluated using both training and testing data sets. Finally, the visualization modules of the training stage including the evolution of the performance metrics on both training and testing data sets, visualization of the trained XGBoost trees, and the XGBoost feature importance can be employed. The testing stage begins with running the testing visualization modules including receiver operating characteristic (ROC) curves and confusion matrix. Finally, SHAP values can be calculated for the testing data and SHAP visualization modules can be applied on the testing data.

The main idea of generalization (as shown in Fig. 1 in the box with dashed

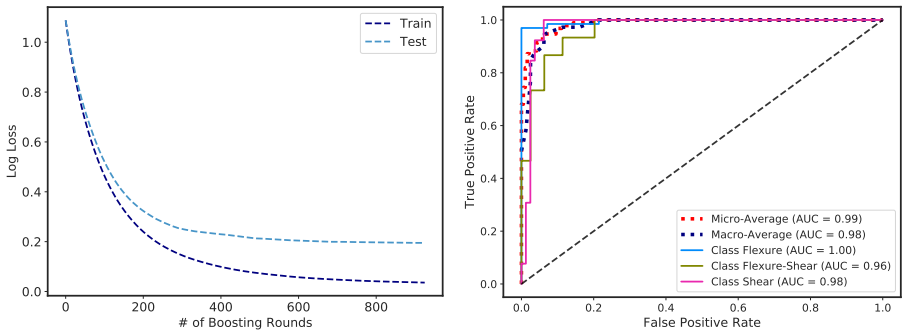
line) is to validate the reliability of the model by permuting the train/test data. This would reduce the possibility of the stochastic results by incorporating the statistical significance of the classification metrics over multiple iterations. The generalization step begins with initializing the number of the iterations ($n = 1, 2, ..., N$). Then, for each iteration, the random seed is being initialized (new random seed = random seed $\times$ n) to maximize the possibility of the coverage of the whole data in train/test sets. Consequently, the train/test split module in a stratified fashion with the new random seed along with model training module and the best set of hyper-parameters from the training component can be employed. It should be noted that we should not incorporate a dynamic module to tune the hyper-parameters at each iteration since the main idea here is to evaluate how reliable the trained model can be for different permutation of the database. For each trained model, classification metrics can be calculated based on the testing set (comes from a unique random seed). For instance, for N iteration, we would have N models, and each model can be validated over its testing set using a classification metric. Therefore, we end up with an array of metric values of size N which can be used for statistical significance tests and confidence intervals.

3. Results & Discussion

In this section, the performance of the proposed framework is investigated via two case studies (1) failure modes in RC columns, and (2) failure modes in RC shear walls. In addition to this, the reproducibility, reliability, and generalizability of the trained models is validated and the results of the proposed pipeline are compared with similar studies.

3.1. Case Study 1: Failure Modes in RC Columns

In this experimental study, the data contains 311 specimens of circular and octagonal RC columns with 3 failure modes including 217 in flexure, 50 in flexure-shear, and 44 in shear. The main features of the database to classify the



(a) Evolution (b) ROC

Figure 2: XGBoost performance curves: (a) Evolution of log-loss for the train/test sets through number of boosting rounds, and (b) ROC curves of various failure modes of RC columns.

failure modes are (1) aspect ratio (a/D), where a is the shear span length, and D is the diameter of circular columns, (2) axial load ratio ($P/f'_c A_g$), where P is the axial load on the column, f'_c is the compressive concrete strength, and A_g is the cross sectional area of the column, (3) longitudinal reinforcement index ($\rho_l f_y/f'_c$) where ρ_l is the longitudinal reinforcement ratio, and f_y is the yield strength of longitudinal reinforcement, and (4) transverse reinforcement index ($\rho_s f_{yh}/f_t$), where $\rho_s = 4A_{sp}/d_s$ is a composite factor of transverse reinforcement area A_{sp} and the distance between hoops d_s, yield strength of transverse reinforcement f_{yh}, and tensile concrete strength f_t [31, 32].

As discussed in section 2, the data is splitted into train/test sets in a stratified fashion with 70% as training and 30% as testing (66 specimens with flexure, 15 specimens with flexure-shear, and 13 specimens with shear failure modes). The hyper-parameters of the trained XGBoost model was chosen using Bayesian optimization. Fig. 2a illustrates the evolution of the multi-class-logarithmic-loss (mlogloss) as the chosen evaluation metric over the number of boosting rounds for both training and testing sets. As seen, the mlogloss values decay for both training and testing sets through number of boosting rounds. This fact validates the trained model as a just-right fitted model and vanishes any chance

Table 1: Classification results of trained model in prediction of testing set for various failure modes of RC columns.

Failure Mode	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Flexure	0.96	0.98	0.97	0.98
Flexure-Shear	0.82	0.60	0.69	0.60
Shear	0.80	0.92	0.86	0.92
Macro-Average	0.86	0.84	0.84	0.83
Micro-Average	0.91	0.91	0.91	0.91

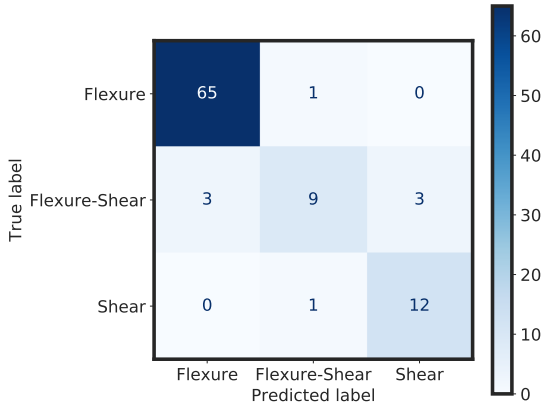


Figure 3: Confusion matrix of the classification results of trained model over testing set for various failure modes of RC columns.

of over-fitting. In addition to this, Fig. 2b depicts the ROC curves with area under curve (AUC) of various failure modes of RC columns. An ROC curve presents false positive rate (1-specificity) versus true positive rate (sensitivity or recall) under different classification thresholds. The true positive rate is the proportion of positive cases that are correctly classified while the false positive rate is the proportion of negative cases that are incorrectly classified as positive. The performance can be evaluated through how well a model separates the true positive rate from the false positive rate. The area under the ROC curve provides a straightforward measure where an AUC of 1.0 represents a perfect model and

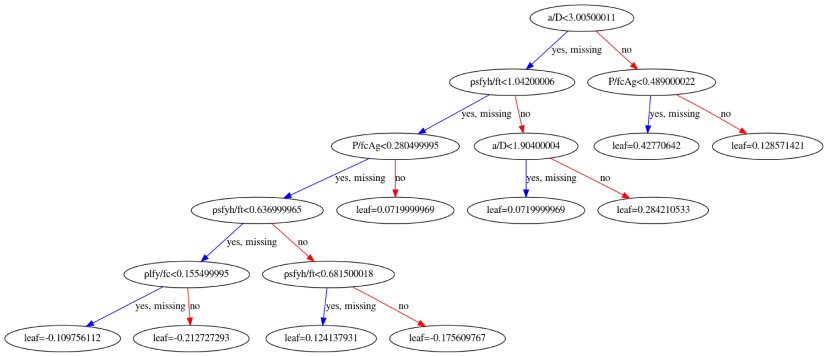


Figure 4: First trained tree of the XGBoost model for RC columns.

an AUC of 0.5 represents a worthless (stochastic) model. The closer the AUC to 1.0, the better the model. As shown in Fig. 2b, the solid lines present the curve for each of the failure modes separately and the red and blue dashed lines present micro-average (weighted-average) and macro-average (numeric-average) AUCs, respectively.

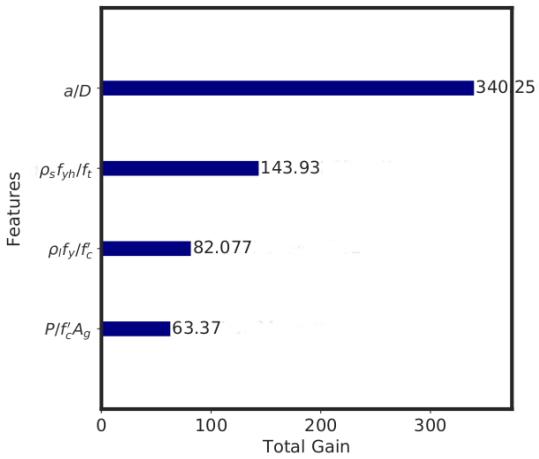
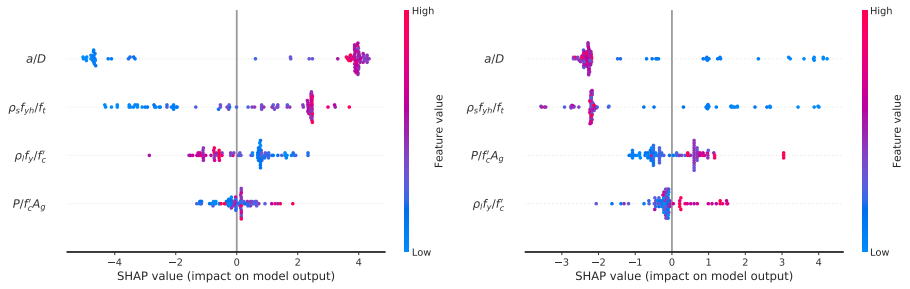


Figure 5: Feature importance of the trained XGBoost model for RC columns.

Table 1 presents the classification results of the trained model applied on the testing set for various failure modes of RC columns. This table includes the report of precision, recall, f1-score, and accuracy for all three modes separately. In addition to this, macro-average, and micro-average calculation of

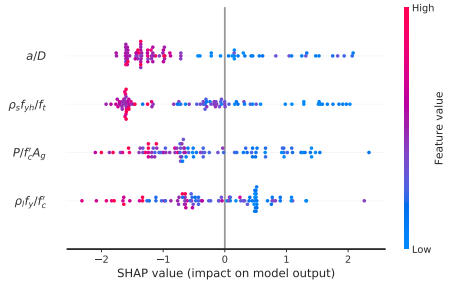
the classification metrics are also presented. The trained XGBoost model has the best performance in predicting of flexure modes with a precision of 0.96, recall of 0.98, and accuracy of 0.98. This could be due to extra presence of the flexure modes in the training data (typical imbalanced classification problem statement). However, the model showed lower recall in prediction of flexure-shear failure mode, and lower precision in prediction of shear failure mode in the testing set. Additionally, the shear and flexure-shear failure modes both have close precision around 0.80, while their recall values are 15% off from each other with the fact that the shear and flexure-shear failure modes both cover the same fraction of the train/test sets (around 16%). This would rise the fact that how important is to use a stratified fashion for train/test splits to minimize any possibility of over-fitting. This would also increase the chance of training a generalizable model with even small number of specimens in any database. Fig. 3 presents the confusion matrix of the classification results of the testing set. The diagonal elements represent failure modes that are predicted correctly. Among 66 specimens in flexure class, 1 case was incorrectly classified as flexure-shear, and among 13 specimens with shear class, 1 case was incorrectly classified as flexure-shear. In principle, the trained model has good ability to distinguish between flexure and shear cases and it did not classify any of the specimens in these classes as the other class, while among 15 specimens with flexure-shear, 3 specimens were incorrectly classified as flexure, and 3 specimens were incorrectly classified as shear (60% accurate). Therefore, increasing the number of specimens with flexure-shear or adding more features with the ability to decrease the marginal error between flexure and shear would improve the model performance. Mangalathu et al. [31] have also previously noted that it is often arduous to properly establish the decision boundaries between flexure-shear and other modes of failure.

One of the most important aspects of the training a model is how the features contribute in the training process which can be used as a metric to measure the relative impact of the features in the trained model. Fig. 5 illustrates the feature importance of the trained model based on the total gain metric. The



(a) Flexure

(b) Shear



(c) Flexure-Shear

Figure 6: SHAP summary plots of the testing set for various failure modes of RC columns: (a) Flexure, (b) Shear, and (c) Flexure-Shear.

gain implies the relative contribution of the corresponding feature to the trained model calculated by taking each feature's contribution for each tree in the model. A higher value of this metric when compared to another feature implies that the corresponding feature has more impact for generating a prediction. In principle, total gain is the total improvement in evaluation metric (mlogloss here) brought by a feature with respect to all features to the branches it is on. In fact, before adding a new split on a feature X to the branch, there were some wrongly classified elements, after adding the split on this feature, there are two new branches, and each of these branches would be more accurate (one branch saying if your observation is on this branch, then it should be classified as 1, and the other branch saying the exact opposite and it should be classified as 0). As shown in Fig. 5, aspect ratio has the most impact in the model and its total gain

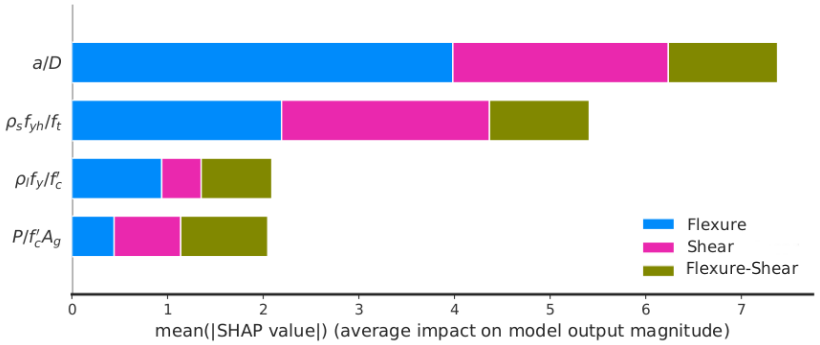
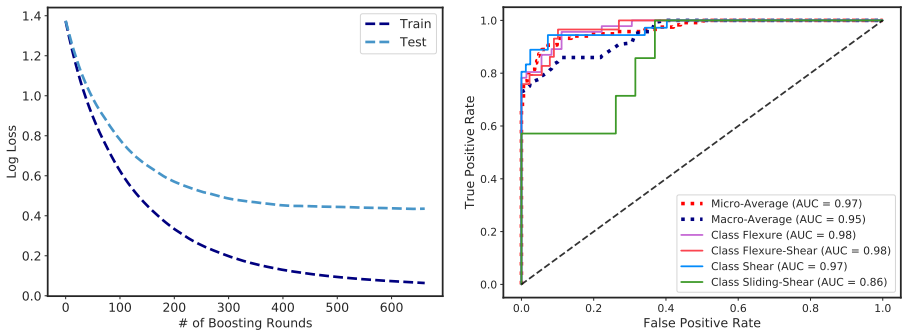


Figure 7: SHAP summary plot of the testing set with absolute impact of the features for various failure modes of RC columns.



(a) Evolution (b) ROC

Figure 8: XGBoost performance curves: (a) Evolution of log-loss for the train/test sets through number of boosting rounds, and (b) ROC curves of various failure modes of RC shear walls.

is more than two times more than the total gain of the transverse reinforcement index which is the next important feature in the model. This can also be seen by looking at the first couple trained trees. Fig. 4 illustrates the first trained tree of the XGBoost model. As seen, the first split is over aspect ratio, and second splits over transverse reinforcement index and axial load ratio, respectively.

In addition to the XGBoost feature importance plot, SHAP can be a great tool in order to reveal the interpretation of each single predictions of the trained model. Fig. 6 presents the SHAP summary plots of the trained XGBoost

model for (a) flexure, (b) shear, and (c) flexure-shear failure modes using tree explainer which is a combination of the feature importance with considering the feature effect. For each of the features, the SHAP values (each dot in the summary plot) and their impacting contribution (range and distribution) to the model (high as red, low as blue) are shown. The density of the dots in the summary plot indicates the real distribution of the exemplars in the testing data set. As seen, the aspect ratio is the important feature which indicates the importance of the geometry of the columns. This result is along with what we have already seen in Fig. 5 (importance over the training set) which indicates the fact that model is properly trained and the possibility of over-fitting is reasonably minimized. However, the aspect ratio has shown different impacts based on the failure modes. For instance, having larger values (in the test set) for aspect ratio in the shear failure mode summary plot as shown in Fig. 6b has shown counter-predictive response in the trained model while larger aspect ratio values has shown direct response in prediction of the flexural model as depicted in Fig. 6a. Next impactful feature is the transverse reinforcement index where higher values have counter-predictive response in the trained model for shear and flexure-shear failure modes. On the contrary, it can be seen in Fig. 6a that higher transverse reinforcement index values increase the possibility of the flexural failure. The longitudinal reinforcement index and the axial load ratio are the next important features in which they have shown mixed impacts. For instance, the longitudinal reinforcement index has shown inverse response between flexure and shear failure modes. In fact, the specimens with higher longitudinal reinforcement index have shown counter-predictive responses for flexural failure (Fig. 6a) while they have predictive responses in shear failure (Fig. 6b). Fig. 6c presents the SHAP summary plot of the flexure-shear failure model. Despite the other two failure modes, the SHAP values of the flexure-shear failure mode have shown consistency with their actual values. In principle, as the feature values decrease, the predictive responses of the model increase.

While SHAP values can have both positive and negative values, for the sake of comparison, the average of absolute SHAP values are used in Fig. 7

to compare the global average impact on the model output magnitude ($I_j = \sum_{i=1}^{n} |\phi_j^{(i)}|$) for flexure (blue bar), shear (magenta bar), and flexure-shear (olive-green bar) modes of failure. The idea behind SHAP feature importance is simple: features with large absolute SHAP values are important. This would be the global impact of the features over the testing set for each failure mode. It is always recommended to compare Fig. 7 with the XGBoost feature importance (Fig. 5). As seen, the features have the same ranking in both of the figures and they share close impact/importance over the model. For instance the aspect ratio has 2.4, 4.1, and 5.4 times more total gain than transverse reinforcement index, longitudinal reinforcement ratio, and axial load ratio, respectively. Similarly, the aspect ratio has 1.5, 3.8, and 3.8 times more average impact on model output magnitude than transverse reinforcement index, longitudinal reinforcement ratio, and axial load ratio, respectively. It should be noted that the XGBoost total gain (feature importance) is calculated over the course of the training data while the SHAP summary plot is calculated based on the trained model over the course of testing data.

Table 2: Classification results of trained model in prediction of testing set for various failure modes of RC shear walls.

Failure Mode	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Flexure	0.87	0.89	0.88	0.89
Flexure-Shear	0.79	0.93	0.86	0.93
Shear	0.94	0.86	0.90	0.86
Sliding-Shear	1.00	0.57	0.73	0.57
Macro-Average	0.90	0.81	0.84	0.82
Micro-Average	0.88	0.87	0.87	0.87

3.2. Case Study 2: Failure Modes in RC Shear Walls

In this experimental study, the data contains 393 specimens of RC shear walls in which 238 of the specimens have a rectangular, 95 with barbell type,

and 60 with flanged cross sections. It should be noted that all of the presented shear walls include symmetric cross sections and continuous longitudinal reinforcement without lap splices, deformed, and straight reinforcement. Moreover, the selected specimens with RC shear walls have 4 failure modes including 152 in flexure, 122 in shear, 96 in flexure-shear, and 23 in sliding-shear. The main features of the database to classify the failure modes are (1) aspect ratio M/Vl_w calculated as the shear span length to the wall length where M is the base moment, V is the base shear, and l_w is the wall length, (2) length to thickness ratio of the wall l_w/t_w, where t_w is the thickness of the wall, (3) axial load ratio (P/f'_cA_g), where P is the axial load on the column, f'_c is the compressive concrete strength, and A_g is the cross sectional area of the column, (4) ratio of the boundary element to cross sectional area A_b/A_g, where A_b is the boundary element, (5) section, (6) web vertical reinforcement index $\rho_{vw}f_{y,vw}/f'_c$, where ρ_{vw} is the vertical reinforcement ratio of the web, and $f_{y,vw}$ is the vertical yield strength of web reinforcements, (7) web horizontal reinforcement index $\rho_{hw}f_{y,hw}/f'_c$, where similarly ρ_{hw} is the horizontal reinforcement ratio of the web, and $f_{y,hw}$ is the horizontal yield strength of web reinforcements, (8) boundary element vertical reinforcement index $\rho_{vc}f_{y,vc}/f'_c$, where ρ_{vc} is the vertical reinforcement ratio of the boundary element, and $f_{y,vc}$ is the vertical yield strength, and (9) boundary element horizontal reinforcement index $\rho_{hc}f_{y,hc}/f'_c$, where ρ_{hc} is the horizontal reinforcement ratio of the boundary element, and $f_{y,hc}$ is the horizontal yield strength [31, 32].

Similar to the first case study , the data is splitted into train/test sets in a stratified fashion with 70% as training and 30% as testing (46 specimens with flexure, 29 specimens with flexure-shear, 36 specimens with shear, and 7 specimens with sliding-shear failure modes). The section feature includes various cross section shapes including rectangular, barbell, and flanged with categorical values. The categorical values are encoded as integers for the sake of modeling. Moreover, the hyper-parameters of the trained XGBoost model was chosen using Bayesian optimization. Similarly, Fig. 8a illustrates the evolution of the mlogloss over the number of boosting rounds for both training and testing

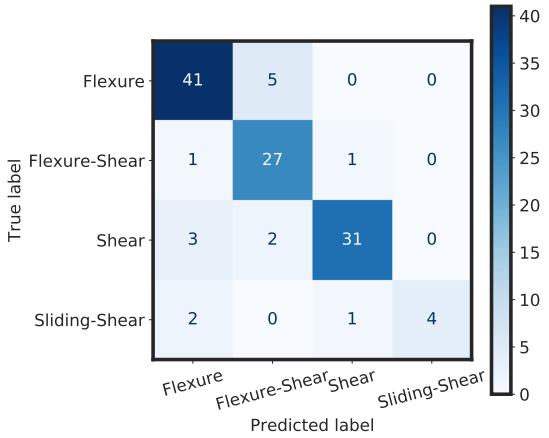


Figure 9: Confusion matrix of the classification results of the trained model over the testing set for various failure modes of RC shear walls.

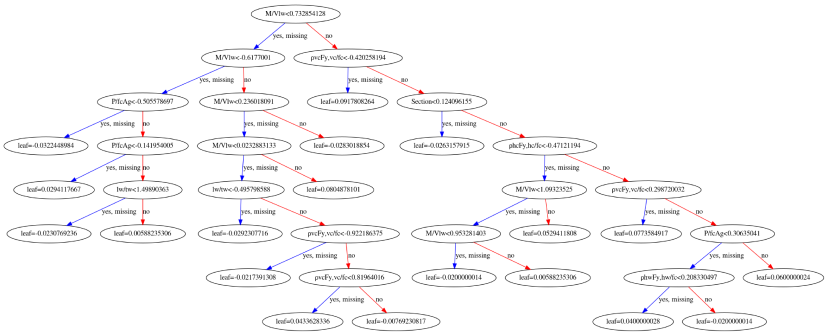


Figure 10: First trained tree of the XGBoost model for RC shear walls.

sets and Fig. 8b shows the ROC curves of various failure modes of RC shear walls, where the solid lines present the curve for each of the failure modes separately and the red and blue dashed lines present the micro-average and macro-average AUCs, respectively. As seen, the mlogloss values decay for both training and testing sets through number of boosting rounds, flexure and flexure-shear failure modes have the best AUC values (0.98), and sliding-shear failure mode has the worst performance with an 0.86 AUC among all failure modes of the RC shear walls.

Table 2 presents the results of the trained model in prediction of the testing

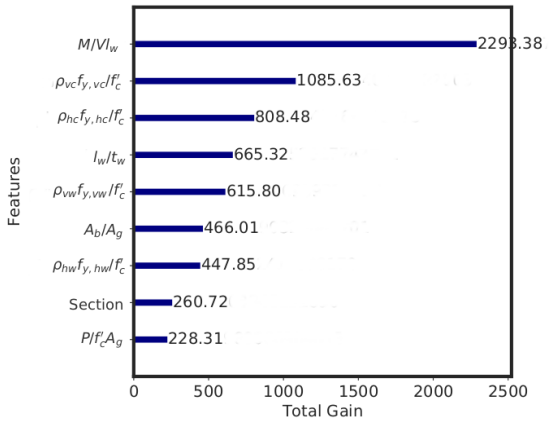
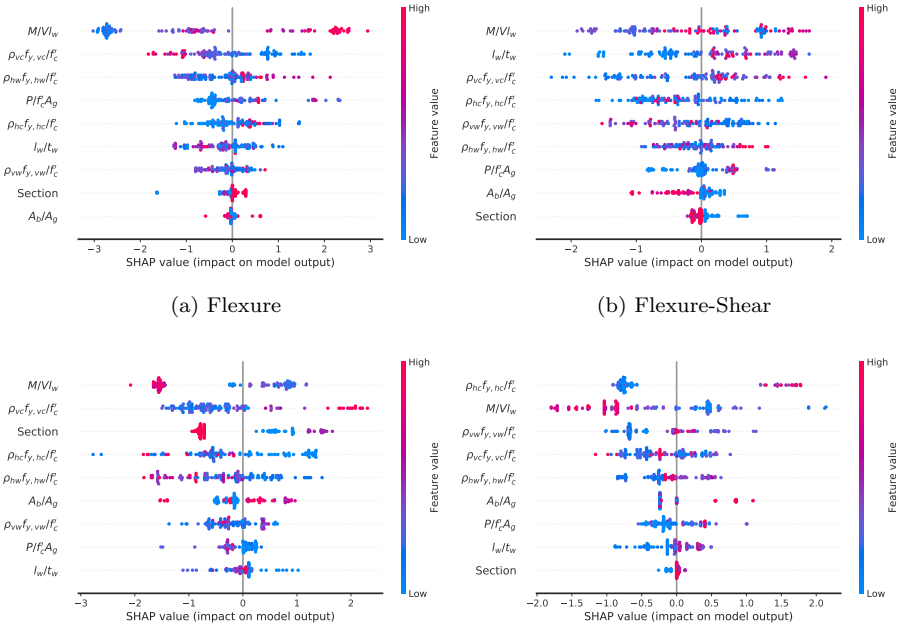


Figure 11: Feature importance of the trained XGBoost model for RC shear walls.



(a) Flexure

(b) Flexure-Shear

(c) Shear

(d) Sliding-Shear

Figure 12: SHAP summary plots of the testing set for various failure modes of RC shear walls: (a) Flexure, (b) Flexure-Shear, (c) Shear, and (d) Sliding-Shear.

set for various failure modes of RC shear walls. The trained XGBoost model has the best precision in prediction of the sliding-shear failure mode, the best

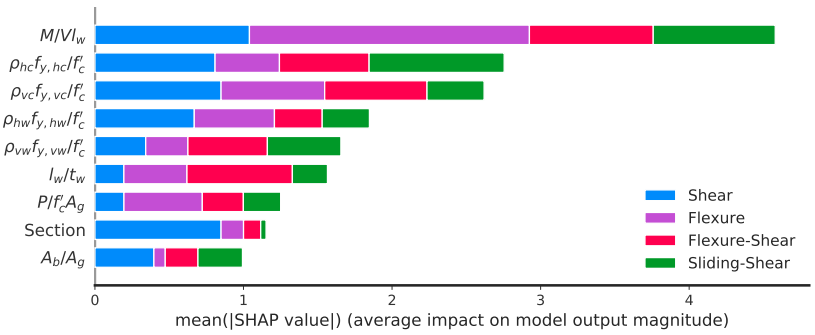


Figure 13: SHAP summary plot of the testing set with absolute impact of the features for various failure modes of RC shear walls.

recall and accuracy in prediction of the flexure-shear failure mode, and the best f1-score in prediction of the shear mode with values of 1.0, 0.93, 0.93, and 0.90, respectively. The probabilistic values can be taken from their AUC values based on the ROC curves according to the Fig. 8b where flexure, and flexure-shear modes they both have an AUC of 0.98, shear mode has an AUC of 0.97, and the sliding-shear mode has the worst AUC (0.86). Furthermore, Fig. 9 presents the confusion matrix of the classification results of the testing set. High precision and low sensitivity (recall) values of the sliding-shear failure mode can be due to the few number of specimens in the experimental database that can be a good point of attention for the future studies. Feature contribution of the trained model is presented in Fig. 11 using the XGBoost total gain metric, where similar to the RC columns case study, the aspect ratio has the highest total gain among all of the features. This implies that the geometrical features still play the most important role in RC failure assessment. Similar to the RC column results, this fact can also be seen by looking at the first couple trained trees, where the first split is over the aspect ratio. Fig. 10 illustrates the first trained tree of the XGBoost model.

Fig. 12 presents the SHAP summary plots of the trained XGBoost model for (a) flexure, (b) flexure-shear, (c) shear, and (d) sliding-shear failure modes using tree explainer. Similar to Fig. 5, the aspect ratio is the most important feature

which indicates the importance of the geometry in prediction of the failure in shear walls. The higher values of the aspect ratio showed linear correlation with the prediction of the flexure (Fig. 12a), and flexure-shear (Fig. 12b), while they have inverse effects (counter-predictive feature) on sliding-shear (Fig. 12d) and shear (Fig. 12c) failure modes. Following the aspect ratio, the boundary element vertical reinforcement index has shown mixed importance based on the testing set where higher values of the vertical reinforcement index has predictive responses in prediction of shear failure modes while they have shown counter-predictive responses in prediction of flexure failure mode. In addition to this, length to thickness ratio is the second imapactful feature in prediction of the flexure-shear failure mode while it has not shown importance in prediction of the other failure modes which can be clearly seen in Fig. 13 as the global average impact on the model output magnitude for flexure (purple bar), shear (blue bar), flexure-shear (red bar), and sliding-shear (green bar) modes of failure. The aspect ratio, horizontal and vertical boundary element reinforcement indices, following by the horizontal and vertical web reinforcement indices are the top five features with the highest global impact in prediction of various failure modes of RC shear walls. It should be noted that the SHAP explainability has numerous variations and in this study only some of them were presented, while the proposed framework does have the ability to visualize all of the available SHAP visualizations (more on `https://github.com/slundberg/shap`).

3.3. Generalization Study

Last, the validation of the generalizability of the proposed models is desired. As shown in section 2, the proposed framework has the feature to validate the results to be reproducible and statistically reliable. For example, the main train/test runs of both case studies resulted in micro-average $AUC = 0.987$, and $AUC = 0.968$ for RC columns and RC shear walls, respectively. The confidence interval (CI) for any significance level can be calculated using the following formula:

$$CI = Score \pm Z_{Score} \times \sqrt{\frac{Score \times (1 - Score)}{N}} \tag{1}$$

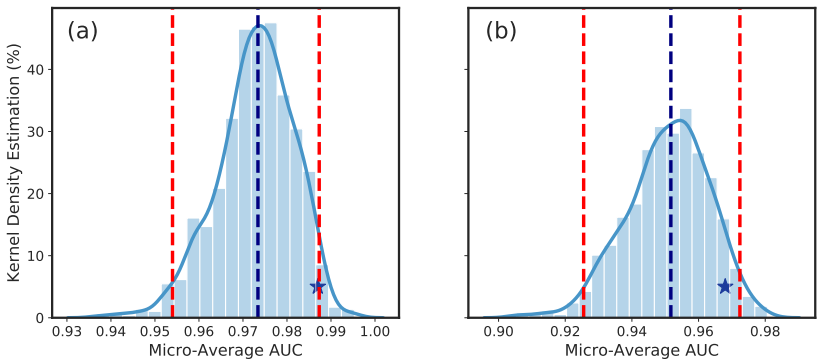


Figure 14: Histograms of the micro-average AUCs based on the generalization runs with random-stratified train/test sets for (a) RC columns, (b) RC shear walls. The stars illustrate the related performance of the trained models presented in Section 3. The figures share the same y-axis scale for the sake of comparison.

where Z_{Score} for 95% significance level is 1.96, and N is the testing sample size. Therefore, the CI for the micro-average AUC of RC columns (Score=0.987 and N=94) would be $[0.965, 1.00]$, and similarly the CI for the micro-average AUC of RC shear walls (Score=0.968 and N=118) would be $[0.937, 1.00]$. The presented CIs are true based on the central limit of theorem [1] if we have normal distributions for the scores which can be acquired if we have big enough sample size. To simulate this theorem and validate the generalizability of the models, 1000 different runs with random-stratified train/test sets as defined in section 2 employed. Fig. 14 illustrates the histograms of the micro-average AUCs based on the 1000 generalization runs with random-stratified train/test sets for (a) RC columns, (b) RC shear walls, where the light-blue solid line, navy dashed line, and red dashed lines present the kernel density estimation, median (50^{th} percentile), lower and upper confidence intervals, respectively. In addition to this, the scatter point with star marker depicts the micro-average AUC resulted based on the prediction of each trained model over the testing set. The generalization confidence intervals with 95% statistical significance level for RC columns and RC shear walls are $CI = [0.9539, 0.9873]$ and $CI = [0.9255, 0.9723]$, respectively. As shown, the micro-average AUCs of the testing sets (star markers) are inside

the experimental confidence intervals for both case studies which proves the fact that the presented models are statistically generalizable with 95% significance level. Moreover, the micro-average AUCs of the main models are in the right tail of the distributions which can be due to the tuned hyper-parameters of the XGBoost resulted from the Bayesian optimization. In addition to this, the performance of two case studies can be compared via the spread of two distributions. As shown in Fig. 14, the histograms for RC shear walls (Fig. 14(b)) has a wider spread. This can be due to higher number of failure modes in RC shear walls study and relatively close number of specimens in the study. This can be used as a useful notes to engineers to take into account before designing experiments.

3.4. Comparative Study

As a comparative study, we have outperformed the results of the best models presented by Mangalathu et al. [31] based on the same experimental database. They have employed random forests as the best models with micro-average testing accuracy of 84% and 86% for RC columns, and RC shear walls, respectively. However, as we reported in Table 3, the presented XGBoost models have improved the micro-average testing accuracy for 7% and 2% for RC columns, and RC shear walls, respectively. In addition to this, we have compared the overall precision and recall of the models, where the models presented by the current framework outperformed the random forests models. It should be noted that the testing size in both studies was set to 30% of the data. The proposed framework showed robust performance in multi-label imbalanced classification while the results presented by Mangalathu et al. did not handle this issue. In principle, Mangalathu et al. did not pay attention to the importance of the stratification of the imbalanced classes and this was ended up with misleading results. In fact, in imbalanced classification problems the crucial goal is to find a trade-off to predict the low prevalence class as well as the high prevalence class, while training a model that detects the high prevalence class is not a challenge and the performance would be still reasonably high. However, stratification

Table 3: Comparison of the classification results of the proposed framework with the model presented by Mangalathu et al. [31]. All the metrics are evaluated as on the testing data and they all reported as micro-average.

Database	Model	Accuracy	Precision	Recall
RC Columns	XGBoost	**0.91**	**0.91**	**0.91**
	Random Forest	0.84	0.86	0.84
RC Shear Walls	XGBoost	**0.88**	**0.87**	**0.87**
	Random Forest	0.86	0.86	0.86

helps to prevent this issue for any test scenarios and prevents possible risks and damages. It is crucial to keep the prevalence of each class in both training and testing sets. Therefore, their models were not generalizable enough since they did not learn all the classes equally. Moreover, boosting algorithms change the training data distribution iteratively with the goal of predicting the specimens that are harder to classify. This feature would enable the proposed framework to outperform the random forest models that are based on constructing parallel decision trees, while XGBoost is a result of a sequence of decision trees.

4. Summary & Conclusions

In this paper, we have proposed an scalable-interpretable modeling framework which can be applied to a variety of engineering problems. The pipeline was applied on two case studies: (1) failure modes in RC columns, and (2) failure modes in RC shear walls. The results of the pipeline for both studies were compared to the benchmark study. Clearly, the results of the proposed pipeline outperformed the results that were presented in the benchmark study and passed the reliability validation tests with high statistical significance. This would give the experimental domain experts enough insights to plan studies that will help establishing better experimental database to reveal the flaws of

the current models that would help creating better machine learning models in the future.

In addition, we have discussed why it is beneficial to use gradient boosting models, as well as some of the explainability complications involved in such models. Through our case studies we have illustrated that by using SHAP values, these models can be interpretable, at least in the area of feature importance. As the availability of data increases, so does the opportunity for machine learning algorithms to discover solutions to real-world problems. Many consumers of machine learning models will not trust the results if they cannot understand the method. While the mechanism and math of a black-box model is still a difficult concept to grasp, we hope that supplementing predictions with understandable feature importance results will go a long way in fostering trust in these methods. If this trust cannot be gained, the benefit of cutting-edge methods in machine learning is largely lost.

Acknowledgements

The authors would like to thank Trace Smith for the careful revision of the final version of the manuscript.

Conflict of Interest

The authors declare that they have no conflict of interest.

Appendix

1. Precision = True Positive/(True Positive + False Positive)
2. Recall = True Positive/(True Positive + False Negative)
3. F1-Score = 2×True Positive/(2×True Positive + False Positive + False Negative)
4. Accuracy = (True Positive + True Negative)/(True Positive + True Negative + False Positive + False Negative)

References

[1] R. O. Duda, P. E. Hart, D. G. Stork, Pattern classification, John Wiley & Sons, 2012.

[2] Y. Reich, Machine learning techniques for civil engineering problems, Computer-Aided Civil and Infrastructure Engineering 12 (4) (1997) 295–310.

[3] P. C. Deka, A Primer on Machine Learning Applications in Civil Engineering, CRC Press, 2019.

[4] R. M. Nabi, S. Soran Ab M, H. Harron, A novel approach for stock price prediction using gradient boosting machine with feature engineering (gbm-wfe), Kurdistan Journal of Applied Research 5 (1) (2020) 28–48.

[5] X. Zhan, S. Zhang, W. Y. Szeto, X. Chen, Multi-step-ahead traffic speed forecasting using multi-output gradient boosting regression tree, Journal of Intelligent Transportation Systems 24 (2) (2020) 125–141.

[6] D. Liu, G. Yang, Y. Li, J. Wu, F. Lv, Gradient boosting tree for 1h-mrs alzheimer diagnosis, International Journal of Data Mining and Bioinformatics 23 (1) (2020) 12–29.

[7] A. Tahmassebi, J. Martin, A. Meyer-Baese, A. H. Gandomi, An interpretable deep learning framework for health monitoring systems: A case study of eye state detection using eeg signals, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2020, pp. 211–218.

[8] P.-j. Chun, S. Izumi, T. Yamane, Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine, Computer-Aided Civil and Infrastructure Engineering (2020).

[9] P.-j. Chun, T. Yamane, S. Izumi, N. Kuramoto, Development of a machine learning-based damage identification method using multi-point simultaneous acceleration measurement results, Sensors 20 (10) (2020) 2780.

[10] P.-j. Chun, I. Ujike, K. Mishima, M. Kusumoto, S. Okazaki, Random forest-based evaluation technique for internal damage in reinforced concrete featuring multiple nondestructive testing results, Construction and Building Materials 253 (2020) 119238.

[11] W. Zhang, C. Wu, H. Zhong, Y. Li, L. Wang, Prediction of undrained shear strength using extreme gradient boosting and random forest based on bayesian optimization, Geoscience Frontiers (2020).

[12] V.-H. Truong, Q.-V. Vu, H.-T. Thai, M.-H. Ha, A robust method for safety evaluation of steel trusses using gradient tree boosting algorithm, Advances in Engineering Software 147 (2020) 102825.

[13] R. E. Schapire, The strength of weak learnability, Machine learning 5 (2) (1990) 197–227.

[14] J. Friedman, T. Hastie, R. Tibshirani, et al., Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), The annals of statistics 28 (2) (2000) 337–407.

[15] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[16] A. Choudhury, T. Konnur, P. Chattopadhyay, S. Pal, Structure prediction of multi-principal element alloys using ensemble learning, Engineering Computations (2019).

[17] Q. Zhao, T. Hastie, Causal interpretations of black-box models, Journal of Business & Economic Statistics (just-accepted) (2019) 1–19.

[18] T. Hastie, R. Tibshirani, J. Friedman, J. Franklin, The elements of statistical learning: data mining, inference and prediction, The Mathematical Intelligencer 27 (2) (2005) 83–85.

[19] J. H. Friedman, J. J. Meulman, Multiple additive regression trees with application in epidemiology, Statistics in medicine 22 (9) (2003) 1365–1381.

[20] J. Pearl, Interpretation and identification of causal mediation., Psychological methods 19 (4) (2014) 459.

[21] M. T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you?: Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, 2016, pp. 1135–1144.

[22] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 3145–3153.

[23] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PloS one 10 (7) (2015) e0130140.

[24] S. Lipovetsky, M. Conklin, Analysis of regression in game theory approach, Applied Stochastic Models in Business and Industry 17 (4) (2001) 319–330.

[25] E. Štrumbelj, I. Kononenko, Explaining prediction models and individual predictions with feature contributions, Knowledge and information systems 41 (3) (2014) 647–665.

[26] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.

[27] S. M. Lundberg, G. G. Erion, S.-I. Lee, Consistent individualized feature attribution for tree ensembles, arXiv preprint arXiv:1802.03888 (2018).

[28] J. Snoek, H. Larochelle, R. P. Adams, Practical bayesian optimization of machine learning algorithms, in: Advances in neural information processing systems, 2012, pp. 2951–2959.

[29] A. Tahmassebi, ideeple: Deep learning in a flash, in: Disruptive Technologies in Information Sciences, Vol. 10652, International Society for Optics and Photonics, 2018, p. 106520S.

[30] A. Tahmassebi, T. Smith, Slickml: Slick machine learning in python (2021). URL `https://github.com/slickml/slick-ml`

[31] S. Mangalathu, S.-H. Hwang, J.-S. Jeon, Failure mode and effects analysis of rc members based on machine-learning-based shapley additive explanations (shap) approach, Engineering Structures 219 (2020) 110927.

[32] S. Mangalathu, H. Jang, S.-H. Hwang, J.-S. Jeon, Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls, Engineering Structures 208 (2020) 110331.