Construction and Building Materials 220 (2019) 587–595

Contents lists available at ScienceDirect

Construction and Building Materials

journal homepage: www.elsevier.com/locate/conbuildmat

Prediction of concrete coefficient of thermal expansion and other properties using machine learning

Vanessa Nilsen [a], Le T. Pham [b], Michael Hibbard [c], Adam Klager [c], Steven M. Cramer [b], Dane Morgan [a,*]

[a] Department of Materials Science and Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA
[b] Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA
[c] Department of Engineering Mechanics, University of Wisconsin-Madison, Madison, WI 53706, USA

HIGHLIGHTS

- A machine learning model effectively predicted concrete coefficient of thermal expansion.
- The model was more accurate than level-2 and level-3 predictions.
- The model was also effective in predicting other concrete properties.

ARTICLE INFO

Article history:
Received 5 July 2018
Received in revised form 23 April 2019
Accepted 1 May 2019
Available online 24 June 2019

Keywords:
Concrete
Coefficient of thermal expansion
Machine learning
Random forest
Compressive strength

ABSTRACT

The coefficient of thermal expansion (CTE) significantly influences the performance of concrete. However, CTE measurements are both time consuming and expensive; therefore, CTE is often predicted from empirical equations based on historical data and concrete composition. In this work we demonstrate the application of linear regression and random forest machine learning methods to predict CTE and other properties from a database of Wisconsin concrete mixes. The random forest model accuracy, as assessed by cross-validation, is found to be significantly better than the American Association of State Highway and Transportation Officials (AASHTO) recommended prediction methods for CTE, denoted as level-2 and level-3.

1. Introduction

The objective of this study was to demonstrate the usefulness of using machine learning in predicting a range of concrete properties with an emphasis on concrete coefficient of thermal expansion (CTE). The ability to predict concrete properties using other indicator properties or descriptors can save time and costs related to making and testing materials. While prediction of concrete strengths has been extensively studied using both non-machine-learning and machine-learning techniques [1–5], there have been fewer studies on prediction of concrete coefficient of thermal expansion (CTE), and none exploring machine learning methods for this property.

Concrete CTE is an important input in pavement design, as detailed in the American Association of State Highway and Transportation Officials' (AASHTO) Mechanistic-Empirical Pavement Design Guide (MEPDG) [6], and has significant effects on slab cracking, joint faulting, and surface roughness [7]. The AASHTO's MEPDG describes three levels of design input for concrete CTE. Level-1 input is site- or project-specific, and requires testing for concrete CTE using the same materials that would be used for a specific paving project. Level-1 data is the most reliable and the most costly. For level-2 design input, the concrete CTE is estimated as the average of the CTE values of mixture components (coarse aggregate, fine aggregate, and cement paste) weighted with respect to their volumetric proportions. CTE values of the mixture components can be determined by experiments or from a database. For level-3 design, the MEPDG permits the use of typical concrete CTE, which can be obtained from a national database, according to the type of coarse aggregate. Level-2 and level-3 design approaches require less effort but are less reliable than level-1. In addition to these three design approaches, transportation agencies have begun to develop their own concrete CTE databases using their typical mix designs and materials and can use such databases to obtain CTE inputs based on project locations.

* Corresponding author.
E-mail address: ddmorgan@wisc.edu (D. Morgan).

https://doi.org/10.1016/j.conbuildmat.2019.05.006
0950-0618/

While these CTE values are not usually a level-1 input as they are not typically the exact materials used in a specific project, CTE measurements on typical local mix designs and materials are likely to be more accurate than the averages provided by level-2 or level-3 input [7]. With the development of these local databases there is an increased need for methods to predict concrete CTE that can extract the full predictive ability of CTE data. Such methods can potentially both reduce the amount of data needed for a useful database and increase the accuracy of CTE predictions for a given project based on the database. There have been a few studies on predicting concrete CTE based on mechanical models [8] or linear regression [9,10]. For example, a study by Yang and Kim analyzed the importance of concrete mix design variables using ANOVA and used these variables to fit a linear equation for predicting CTE [10]. However, Yang and Kim's study did not explore the predictive capabilities of their model through cross validation, which is a focus of this study. There have been no explorations of which we are aware on predicting concrete CTE using machine learning techniques or assessing models beyond linear regressions.

While CTE has not been studied with machine learning methods, neural networks have been widely used in predicting concrete compressive strength (CS) [1–5,11–13], and machine learning has been used widely in other civil engineering applications [14]. Neural network predictions of CS of concrete have achieved a Root Mean Squared Error (RMSE) of approximately 3 MPa on validation data with a range of 82.4 MPa, demonstrating that these techniques can be effective and warrant further study [2].

This paper details the improvements in predicting the CTE of concrete by using machine learning (ML) methods and demonstrates that this approach can result in higher accuracy than level-2 and level-3 predictions. Using a database of concrete composition and properties using materials sourced from Wisconsin [15], many mechanical properties of concrete, including CS and CTE, were predicted using ML models. As ML models used for predicting CTE were explored here for the first time, their predictions are the focus of this paper. Additional mechanical properties of the Wisconsin concrete were modeled and the results of these ML model predictions are discussed for CS in detail and summarized for all the studied properties except CTE in the Supplementary Information.

The outline of the paper is as follows: Section 2 provides an overview of an experimental program that provided the database used in the machine learning models, Section 3 describes the different methods used to predict concrete CTE, Section 4 compares results of the different methods in predicting concrete CTE, Section 5 demonstrates the potential for reducing the amount of experimental efforts by using ML models, and Section 6 demonstrates the potential of the random forest models to predict concrete CS.

2. Database

2.1. Dataset description

The dataset used was compiled as part of a study [15] that aimed to determine the mechanical properties of concrete for use in the AASHTO MEPDG. Mix proportions were varied to determine the effects on mechanical properties of the concrete, namely the CS, modulus of elasticity, modulus of rupture, splitting tensile strength, CTE, and Poisson's ratio (six properties). The mix proportions, properties of constituent materials, and the methods used to measure the concrete properties can be found in reference [15]. To investigate the effects of concrete composition, mixes were varied to include fifteen different sources of coarse aggregate, two different sources of fine aggregate, two sources of Type I ordinary Portland cement, two grades of slag cement (Grade 100 and 120), and three sources of Class C fly ash. By varying the mix proportions, the listed mechanical properties could be measured and used for approximating properties of future mixes in the MEPDG program. The six mechanical properties were predicted from a set of 10 input descriptors (note that when we use one hot encoding for categorial variables this becomes effectively 17 descriptors, as described in the Supplementary Information Table S2), given in Table 1, along with the unit of the descriptor and values or ranges in the database. A total of 110 samples are included in the dataset. Each mechanical property was measured after curing for 7, 14, 28, and 90 days, with the exception of CTE, which was measured only at an age of 28 days. Each of these curing ages was treated separately for each property such that four predictions were made for CS, modulus of elasticity, modulus of rupture, splitting tensile strength, and Poisson's ratio corresponding to the four curing ages.

3. Methods

3.1. Multivariate linear regression

This work explores two regression methods, multivariate linear regression (MVLR) and random forests, for predicting concrete properties. The MVLR is used as a baseline comparison for the ability of more advanced and nonlinear random forests approach. The regression algorithm was obtained from the Scikit-Learn toolkit in Python, and the coefficients were optimized to minimize the sum of squares error [16]. The MVLR method is not described in further detail given its wide use in many fields.

Table 1
Inputs used in the MVLR and random forest models, as well as possible values categorical variables could take or range of continuous variables.

Input Variable	Data Type	Data Categories or Range	Units
Mix Types	Categorical	Grade A-S, Grade A-F, Grade A*	–
Cement Source	Categorical	Source 1, Source 2**	–
Supplementary Cementitious Material (SCM)	Categorical	Slag, Fly Ash, None	–
Fine Aggregate Type	Categorical	Sand A (igneous), Sand B (carbonates)	–
Coarse Aggregate Type	Categorical	Crushed Stone, Glacial Granite	–
Air Entraining Admixture (AEA)	Continuous	7–30	mL/2.5 ft^3
Air Content	Continuous	3.4–6.8	%
Slump	Continuous	1–3	in.
Water-to-Cement Ratio (WCM)	Continuous	0.33–0.40	–
Water Reducing Admixture (WRA)	Continuous	0–125	mL/2.5 ft^3

* Grades A-S, A-F, and A are the terms used to refer to three different concrete mixes in Ref. [15].
** Source 1 and Source 2 are the terms used to refer to two different suppliers of concrete in Ref. [15].

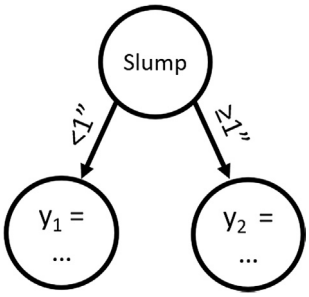

Fig. 1. A basic DT, where the slump of the concrete mix is being used to predict a concrete property.

3.2. Random forest

As random forest approaches are perhaps less familiar to some readers, a brief overview is given here. Random forest approaches are based on the method of decision trees (DT's), which are a decision-making framework based on information theory, a mathematical model for storing information in data [17]. DT's can be used to make predictions about categorical or continuous data. Depending on whether classification (categorical predictions) or regression (continuous predictions) tasks are being pursued, the exact methodology of the DT algorithm varies slightly. Since regression was used for this analysis, this discussion will focus on regression DT's. Regression DT's can be thought of as piecewise regressions, where the exact regression equation used to predict a given data point depends on the values of the features of the data point and the structure of the tree, which will be subsequently described. A basic tree using information about the slump of a concrete mix to predict some concrete property is shown in Fig. 1 to aid in the description of a DT's method. In Fig. 1, each circle in the picture is known as a node, where the uppermost node labelled Slump is known as the root node, and the lowest nodes on the tree are known as the leaf nodes. The black arrows connecting nodes are called arcs and contain information on the value of the node they originate from.

Predictions are made with a DT by starting at the root node of the tree and traversing the tree to reach the leaf nodes. Tree traversal can be thought of as asking a series of questions about the features of the data set. In this example, each non-leaf node of the tree can be thought of as a question about the dataset, and each arc from the node is a possible answer to the question being asked. Using the basic DT in Fig. 1 as an example, the root node examines the slump of concrete in the dataset. If the slump is less than 1″, the data point would travel down the left branch of the tree, and if the slump is greater than 1″, the data point would travel down the right branch of the tree. The process of asking a question about the data set and choosing an answer from the arcs is repeated until a leaf node is reached. At the leaf node, a simple regression is fit to data points in the training data set, and new predictions are made based on the result of the regression at the specific leaf node a sample falls into [18,19]. A visualization of one decision tree fit for this analysis is shown in Fig. 2.

This work uses the random forest and decision tree tools in the Scikit-Learn software package [16]. In this implementation of the algorithm, the order in which decision trees split on features is determined when training the model by calculating either the mean square error (MSE) or the mean absolute error (MAE) of the prediction value versus the actual value of the data being predicted at a given node in the tree. Features that result in a larger reduction in the error are split upon first for every possible attribute, and those that are split upon earlier are more influential on the property being predicted [16]. In addition to the criterion used to assess splits, the way a decision tree functions can also be influenced by limiting the maximum depth of the tree and the minimum number of samples per leaf node [20]. In this study, the maximum depth of the trees and minimum number of samples per leaf node were set to default values (to be provided later in this paper) and modest changes from these values did not have a significant impact on the predictive abilities of the model. These criteria are hyperparameters for decision trees and can be thought of as settings for the algorithm. DT's often make predictions by "memorizing" the training data, resulting in predictions that do not recognize real patterns in the data, a phenomenon known as overfitting [18,21].

To avoid overfitting, many decision trees can be used in conjunction with one another to make predictions. This method is known as a random forest. Random forests have fewer problems with overfitting as they average the results of many trees to mitigate the poor performance of any single tree. Random forests also have several more hyperparameters beyond the split criterion and maximum depth of a tree for single decision trees, such as the number of trees in the forest and whether or not to utilize bootstrapping for sampling trees from the forest. For this study, Scikit-Learn's implementation of a random forest was used to fit

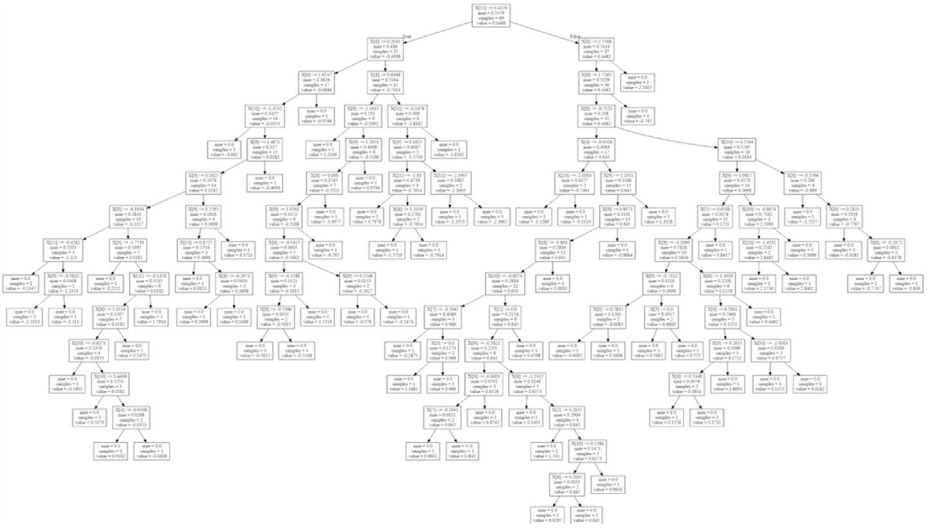

Fig. 2. A visualization of one decision tree fit for our models used for predicting concrete CTE. More trees can be found in Supplementary Information Figure S21. The meaning of the descriptors *X* can be found in Supplementary Information Table S2.

the data. The hyperparameters were set to default values (specifically, the following Scikit-Learn variables were set to the following values: *max_depth* = None (trees can be as deep as needed to fit data), *min_samples_split* = 2 (must have two data points to split), *min_samples_leaf* = 1 (must have at least one sample to be a leaf node), *min_weight_fraction_leaf* = 0.0 (a leaf node can have any weight fraction), *max_features* = None (no constraint on the number of features considered when splitting), *max_lead_nodes* = None (can have any number of leaf nodes), *min_impurity_decrease* = 0.0 (set splitting to occur when any decrease in impurity occurs), *min_impurity_split* = 10^{-7} (threshold of change in impurity below which we stop splitting and form a leaf node)), with the exception of the number of trees in the forest (*n_estimators* = 90 trees were used) and the split criterion (*criterion* = MAE was used to split on mean absolute error). Random forests demonstrate a time complexity of $O(mk\tilde{n}\cdot\log^2(\tilde{n}))$, where m is a constant, k is the number of variables randomly drawn at each node in the DT, $\tilde{n}$ is 0.632*n, and n is the number of samples [22]. In this particular study, this time complexity translates to a runtime of less than one minute required to fit a random forest with 90 DT's and perform 100 rounds of 10-fold cross validation for any given mechanical property in this data set. However, this runtime is dependent on many variables, including the size of the data set being fitted and the power of the computing device.

3.3. Cross validation

In order to assess a model's extrapolative capabilities on validation data not used in training, cross validation can be used to partition data into training and validation sets [23]. Training sets are used to train the ML model, while validation sets are portions of the data set held out for assessing a model's performance. Many varieties of cross validation exist, but most follow the pattern of leaving out random data points when training and using the left out points for validation. This study utilized k-fold cross validation to assess model performance. K-fold cross validation consists of randomly grouping the data into k subsets (also called folds). Of these k subsets, k − 1 are used to train the model (training folds) and predictions are made on the remaining fold (validation fold). The model performance on the validation fold can be assessed by the RMSE of the predicted versus true data and the R^2 [16] values of the parity plot (a plot of predicted versus true values). Here and throughout this work R^2 refers to the coefficient of determination, calculated using Scikit-Learn's r2_score() method [16]. This method computes R^2 using an intercept value of 0 and gives the best possible R^2 value of 1 [24]. The process of training using k − 1 folds, testing on the remaining validation fold, and assessing model performance was repeated until every one of the k folds in the data set has been used for testing. The entire process of splitting data, training and validating until every fold has been used for testing is a single k-fold validation. In this work, the RMSE and R^2 values were calculated for each validation fold used, and the k values of the RMSE and R^2 were averaged to give one RMSE and one R^2 for a full set of k-rounds of cross validation (i.e., for a single k-fold cross validation). The process was then repeated 100 times, with different folds chosen at random for each k-fold cross-validation, to assure the results were not biased by the characteristics of any given k-fold decomposition. The RMSE and R^2 values were averaged over these 100 full k-fold cross validations to assess performance of the model.

Parity plots were created to visually assess the performance of the model. We also explored both 2-fold and 10-fold cross validations, where 2-fold was expected to be significantly more demanding as only half the data was used to predict the other half. The RMSE value from cross validation, either a single k-fold or an average over many k-folds, is referred to as a CV-RMSE. This value can be different from the RMSE of the direct fit due to some level of overfitting.

3.4. Level 1–3 model approaches for CTE prediction

In standard pavement design, three levels of determination are used to predict the CTE for a given mix design. Level-1 uses a standardized test method experimentally measuring the CTE for the mix design, following the specifications set by AASHTO T336 [7]. Level-2 CTE determination is based on a linear combination of the CTE values for the dominant mineral composition of the coarse aggregate as well as the cement paste (which is a combination of cement and water) [25]. The linear combination is based off the percent volume of each component in the total mixture, and is given by $CTE_{mix} = \alpha_{Paste}CTE_{paste} + \alpha_{CA}CTE_{CA} + \alpha_{fA}CTE_{fA}$ [25], where α_{Paste} is the percent volume of the cement paste, α_{CA} is the percent volume of the dominant mineral composition of the coarse aggregate, and α_{FA} is the percent volume of the dominant mineral composition of the fine aggregate. From the provided database, three different dominant mineral compositions were present in the coarse aggregate (quartz, dolomite, and basalt), which have CTE values of 9.3×10^{-6}, 8.9×10^{-6}, and 7.8×10^{-6} °C^{-1}, respectively [7]. Furthermore, the water-to-cement ratio of the cement paste directly affects its coefficient of thermal expansion. For the dataset the water-to-cement ratio was held constant to 0.4, which yields a corresponding cement paste CTE of between 18 and 20×10^{-6} °C^{-1}. For the analysis in this work we used 18×10^{-6} °C^{-1}. Lastly, the level-3 determination for the concrete CTE is based off of a collection of historical data for the CTE values of the dominant mineral composition of the coarse aggregate [25].

4. CTE model comparison results and discussion

The performance of the models was based on a combination of the R^2 value and the normalized root mean square error (RMSE). The RMSE was normalized by dividing by the standard deviation ($\sigma = 0.45\times10^{-6}$ °C^{-1}) of the CTE data, as outlined in Section 3.3. This normalization gives the quality of the RMSE prediction relative to the mean value of the data. The results of the analysis are shown in Table 2. Detailed discussions of each result are subsequently provided.

4.1. Level-2 and 3 approximations

This section focuses on the ability of level-2 and 3 models to predict the concrete CTE values in the given dataset. Comparisons to the performance of level-1 measurements cannot be made as level-1 directly measures the CTE. The parity plots of the level-2 and 3 approximations are shown in Fig. 3 and Fig. 4, respectively. The level-3 prediction yields three distinct lines in the parity plot, which correspond to the three different CTE values for the dominant mineral compositions (quartz, dolomite, and basalt) that were

Table 2
Summary of results comparing the performance of all models used for predicting the concrete CTE.

Model	CV-RMSE ($\times10^{-6}$ °C^{-1})	RMSE/σ ($\sigma = 0.45\times10^{-6}$ °C^{-1})	R^2
Level-2	1.2	2.6	−5.9
Level-3	0.67	1.6	−1.4
Multivariate 10 Fold	0.46	1.0	−0.04
Multivariate 2 Fold	0.48	1.1	−0.17
Random Forest 2 Fold	0.35	0.78	0.39
Random Forest 10 fold	0.22	0.48	0.76

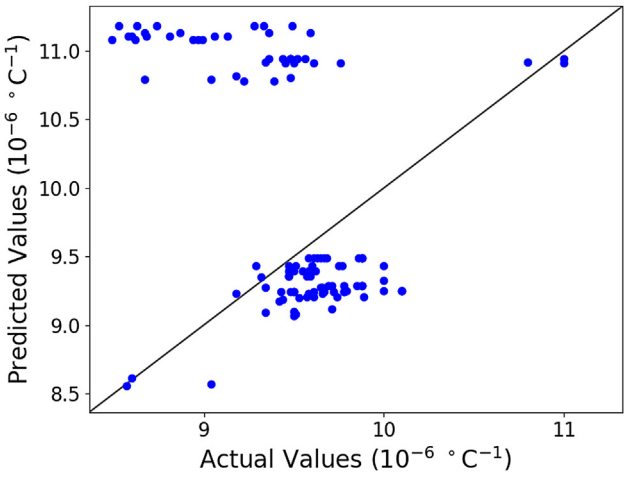

Fig. 3. Parity plot of CTE values predicted using level-2 approximation.

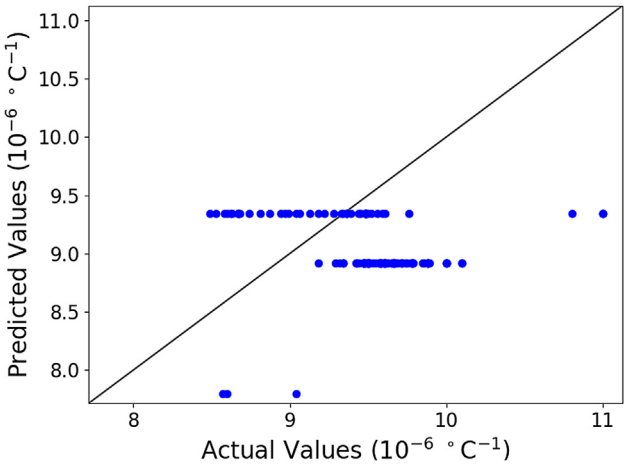

Fig. 4. Parity plot for CTE predictions using level-3 approximation.

present in the dataset. It is clear that the inclusion of a linear combination of materials in level-2 enables more variation than compared to the level-3 approximation. Instead of the three distinct lines observed for the level-3 method, the level-2 method values are more clumped around the line of best fit. However, the linear combination causes more variance in the values, and there is actually a larger spread, i.e., less predictive ability. Although level-3 only predicts with three lines, these lines were more tightly fit around the line of best fit than the level-2 predictions. The R^2 for the level-2 and 3 approximations are negative, and their RMSE values are larger than the standard deviation of the actual CTE values. These R^2 and RMSE results imply that just taking the mean value of the data set would give a better predictor than using either level model, and that level-2 and level-3 methods give a reasonable overall estimate for the typical CTE, but have essentially no ability to predict relative changes in CTE with changing mix for our data set.

4.2. Multivariate linear regression

An MVLR model was explored as a simple baseline fitting approach. Inputs to the MVLR are detailed in Table 1 and the results for the 2-fold and 10-fold cross validations are shown in Fig. 5 and Fig. 6, respectively. The MVLR performs better than the level-2 and level-3 predictions for both 2- and 10-fold cross validation on all metrics (see Table 2). However, although the best-case parity plots show some modest correlation, the mean RMSE/σ is greater than 1 and the mean R^2 is negative for both 2- and 10-fold tests, which shows better predictive abilities than the level-2 and level-3 methods but does not indicate strong predictive abilities. It is also interesting to note that several data points were predicted poorly in all the tests and stand out in the far right of the plot. This consistent failure of the model on these points suggests that the MVLR model could not capture some nonlinearities in the concrete CTE for these mixes. This result supports the need for a nonlinear regression model like random forest.

4.3. Random forest

In addition to the predictions made in Section 4.2 using MVLR, a random forest model was fitted using cross validation to predict the concrete CTE, and the results from 2- and 10-fold cross valida-

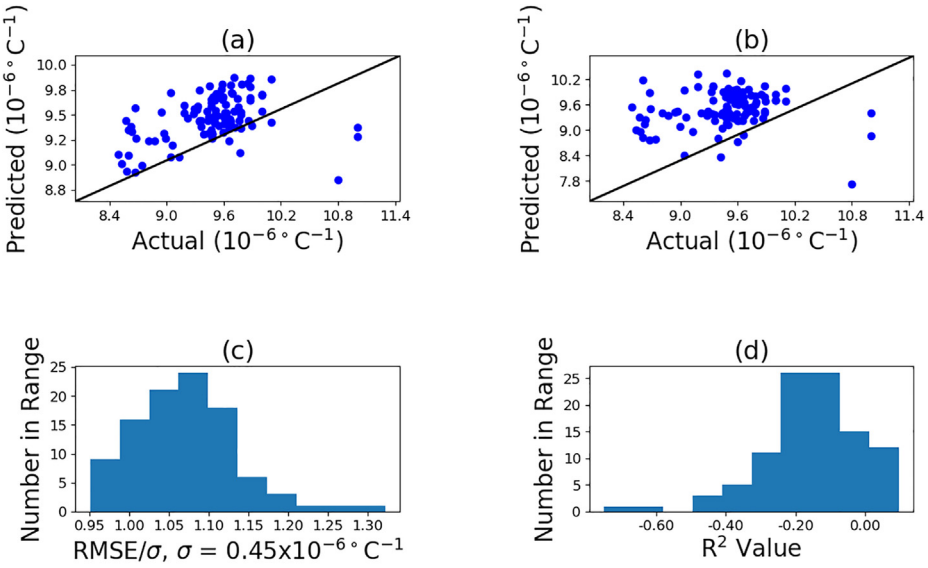

Fig. 5. Results for 100 rounds of 2-fold cross validation using multivariate linear regression. Parity plots for (a) best and (b) worst fits as determined by RMSE (45-degree lines represent perfect fits), and histograms of (c) RMSE and (d) R^2 values.

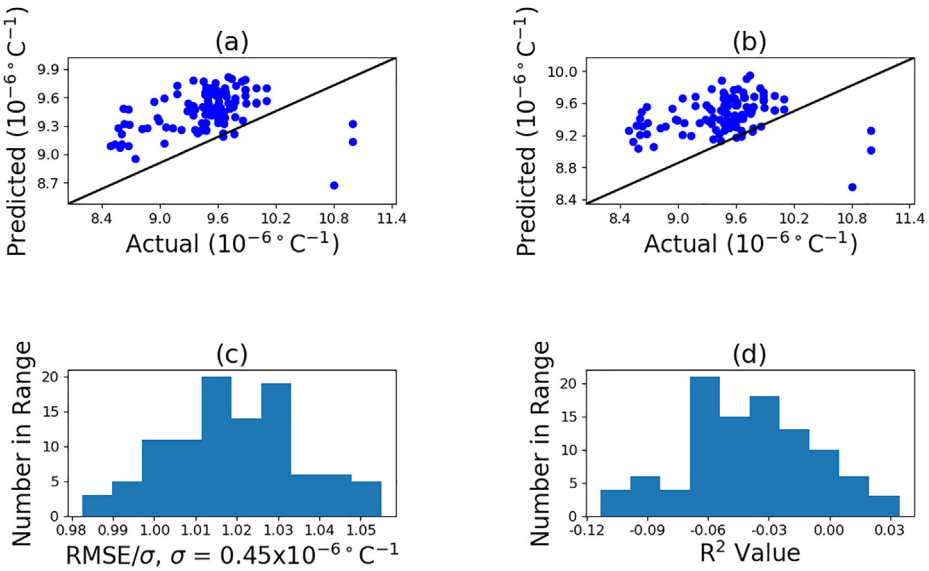

Fig. 6. Results for 100 rounds of 10-fold cross validation using multivariate linear regression. Parity plots for (a) best and (b) worst fits as determined by RMSE (45° lines represent perfect fits), and histograms of (c) RMSE and (d) R^2 values.

tions are shown in Fig. 7 and Fig. 8, respectively. The overall predictive ability of the random forest model is significantly better than the MVLR model, as shown by the statistics in Table 2. In particular, the 2- (10-) fold RMSE/σ values of 0.78 (0.39) are both significantly less than one, and the 2- (10-) fold R^2 values of 0.48 (0.76) are positive and not too far from the ideal of 1.0 for the 10-fold, both strongly suggesting the model has the ability to predict trends in the CTE data. This conclusion can be seen more intuitively by examining the parity plots in Figs. 7 and 8, all of which show a clear correlation and similarity to the perfect agreement 45° line. Furthermore, the three outliers seen in the MVLR are no longer always outliers, although some of them still stand out in some of the worst fit cases. The statistics are much better for 10- vs. 2-fold cross validation, suggesting that using just half the data to build the model may be inadequate for a robust model. Focusing on just the 10-fold cross validation, Fig. 8 shows that the spread in R^2 is 0.68–0.85 and in RMSE/σ is 0.60–0.90, which shows that essentially every fold studied yielded significant predictive ability for the model. This result suggests that the model can be used to predict new data at least for similar systems. In addition to predictions of CTE, random forests can also be used to assess feature importance. This study found that the coarse aggregate type and the fine aggregate type are the most important features in predicting concrete CTE, a result also found by Yang and Kim [10].

As summarized in Table 2, the random forest model dramatically outperforms the more simplistic predictions of CTE, namely level-1, level-2, and MVLR, and is the only model where the analysis suggests any ability to predict trends within our data set. This result suggests that non-linear regression approaches, widely used in machine learning, are an essential tool for accurate modeling of CTE.

5. Using machine learning for reducing the necessary tests for CTE

One potential use of the ML model is to reduce the number of experimental tests needed to obtain a target predictive ability for

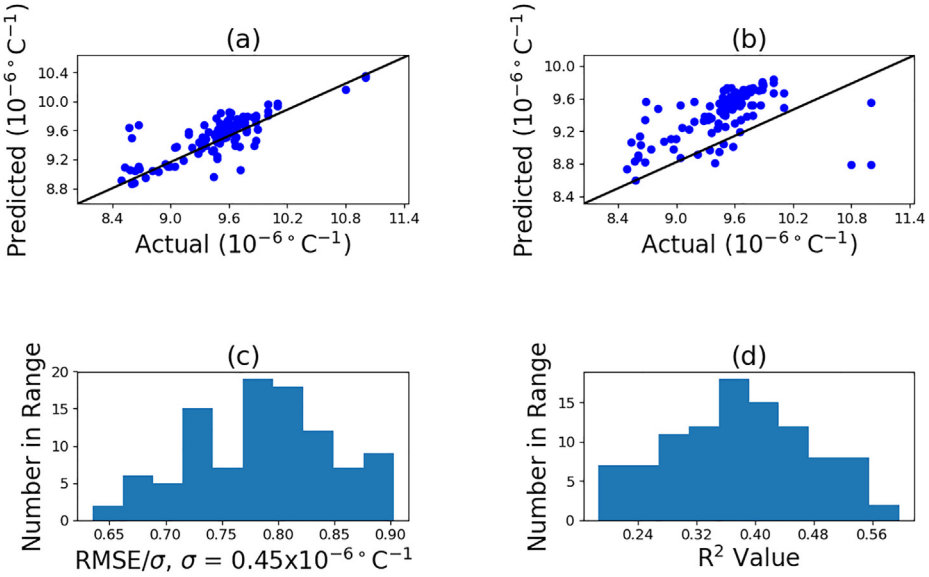

Fig. 7. Parity plots for the a) best and b) worst fits of the random forest model, as well as c) histograms of RMSE and d) R^2 values for 100 rounds of 2-fold cross validation using the random forest model.

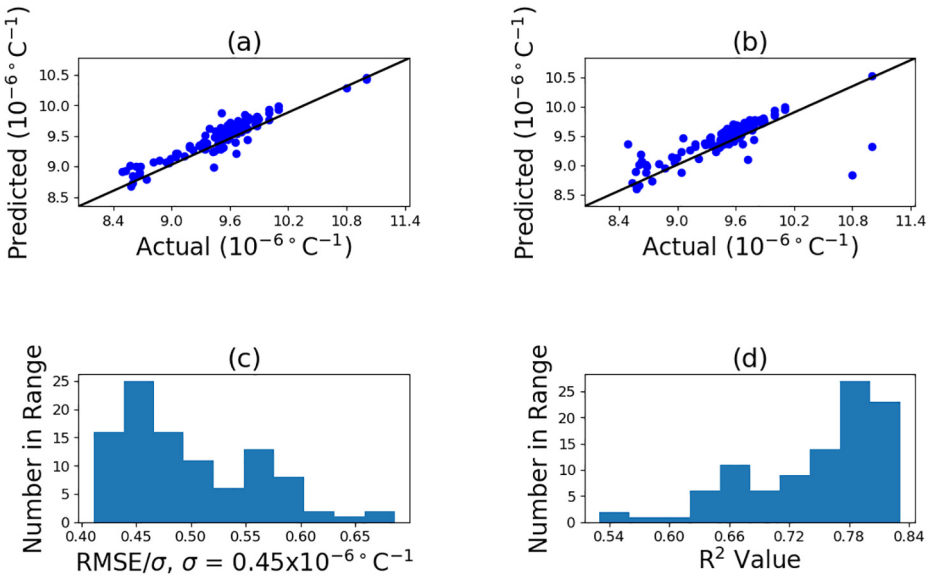

Fig. 8. Parity plots for the best and worst fits of the random forest model, as well as histograms of RMSE and R^2 values for 100 rounds of 10-fold cross validation using the random forest model.

a new mix. In this study, the predictive model was used to assess how much data might be needed to obtain a target RMSE/σ by analyzing the predictive capabilities of the random forest model as a function of the number of data points used in training the model. A random forest was trained using an increasing fraction of the data set and tested using the remaining data, similar to the methods of cross validation used previously. However, the data was not sectioned into equally-sized folds for training and testing as for k-fold cross validation. Instead, the data was randomly separated into only 2 subsets, one each for training and testing, and the size of the training set was varied. Any data points not used for training were then used for validation. RMSE/σ values for a given size training set are averaged over 100 randomly selected training sets of that size. A plot showing the trend of the average RMSE/σ of the testing set assessed for the random forest as a function of the number of data points used in training is shown in Fig. 9. The error bars represent standard deviations of the RMSE/σ of the 100 trainings, where σ is kept constant as the standard deviation of the total data set, as used above.

In this analysis, the lowest RMSE/σ was achieved by training the model with the entire data set. However, using 55 of the 110 data points (50% of the data set) to train the model produced a noticeable decrease in the RMSE/σ. The uncertainty of each prediction was also lowered when predicting using a larger portion of the data set for training up until training using all but one data point. The model proves to be effective in predicting the CTE of concrete in the dataset with only half of the mixes in the total design matrix being measured. Using a similar machine learning model in the future would therefore likely be able to reduce the number of tests required to determine mechanical properties for a similar test matrix.

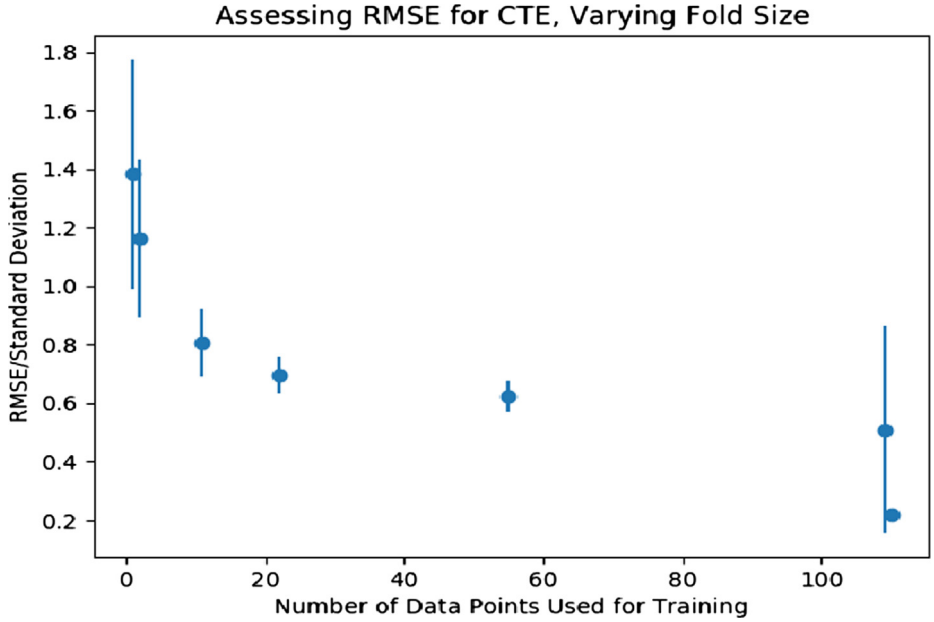

Fig. 9. RMSE/σ for the testing set as a function of the number of data points from this data set used for training. A decreasing trend can be noticed, with the greatest decreases in RMSE/σ seen by adding up to 20 data points for training, and a leveling off around 50 data points for training. The final point on the right represents using all the data in the fit, which gives RMSE/standard deviation of 0.21.

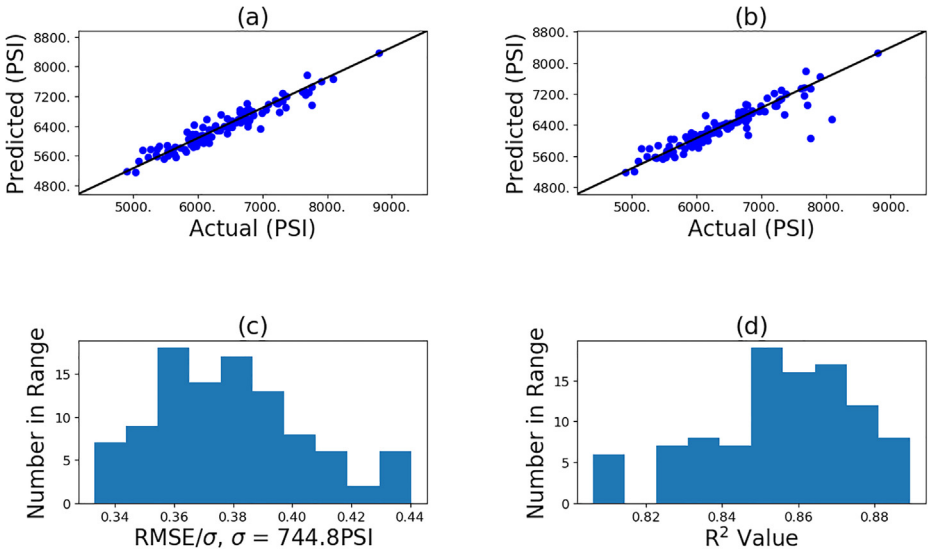

Fig. 10. Results of predictions for 90 day CS predicted using the random forest model and 10 fold CV.

6. Machine learning models for other concrete properties

While CTE is the property discussed in most detail in this analysis, other mechanical properties of concrete were also predicted using the random forest model and assessed using k-fold cross validation. All mechanical properties measured in the data set (CS, modulus of elasticity, modulus of rupture, splitting tensile strength, CTE, and Poisson's ratio) were predicted using random forests [15]. Each of these additional mechanical properties were predicted at ages of 7, 14, 28, and 90 days, and results of these predictions can be found in the Supplementary Information.

Because the CS of concrete has been analyzed and predicted using other machine learning methods [1–5,11–13], the results of the methods used in this work for CS are discussed for comparison. The results of the predictions for CS at an aging time of 90 days are summarized in Fig. 10.

Our methods yield an RMSE of 1.9 MPa for data with a range of 27 MPa, yielding an error to range ratio of 0.07. A representative previous study achieved RMSE of 3.8 MPa given data with a range of 82 MPa [2], yielding an error to range ratio of 0.05. This comparison suggests that our methods yield accuracy consistent with those of previous studies.

7. Conclusions

The machine learning model used in this study was effective in the prediction of concrete CTE for the mixes tested. Analysis of the RMSE of each prediction method revealed that the random forest model's predictions were the most accurate of the models reviewed during this process. In particular, the random forest's predictions were significantly more accurate than estimations using the level-2 and level-3 inputs [7]. Furthermore, the predictions of the random forest model were more accurate than those obtained through multivariate linear regression, demonstrating the need for a non-linear model. Of all the models considered, only the random forest showed any ability to predict trends within the data, although the other methods can estimate the overall typical CTE value. Therefore, such a modeling approach can be used in conjunction with databases similar to the one used for this experiment to predict CTE for new concrete mixes.

This study suggests that for randomly sampled values of the 10 independent variables considered here, reductions in the number of measurements required to accurately predict the CTE of concrete can be achieved with machine learning methods, as shown in Fig. 9. Using 50% of the available data set, the CTE of a given mix in the data set can be predicted with better accuracy than could be achieved using level-2 or level-3 prediction methods. When compared to results of training the machine learning model using just one experimental CTE value, errors were reduced by about 30% when 20 data points were used for training and about 40% when 50 data points were used. Further improvements in machine learning models and expansion of available CTE data for concrete could lead to predictions of concrete CTE approaching the accuracy of laboratory measurements without requiring an experiment for every desired value.

The RMSE/σ values obtained by predicting the mechanical properties of concrete used in this study suggest that machine learning can be effectively applied for a range of concrete properties. Detailed results are given for CTE, but results are also presented for the CS, modulus of elasticity, modulus of rupture, splitting tensile strength, CTE, and Poisson's ratio. Previous machine learning models of concrete properties have focused on CS, and our models show comparable accuracy to the study by Yeh, et al. [2]. Overall, the results show significant opportunity for machine learning to obtain more accurate predictions of mechanical properties of concrete or reduce the number of necessary measurements in developing databases, or both.

Acknowledgements

Support for Dane Morgan was provided by the National Science Foundation (SI2) award No. 1148011 and for Vanessa Nilsen by the Materials Research Science and Engineering Ccenter (MRSEC) program, which is funded by the National Science Foundation Grant No. DMR-1720415. We thank the Wisconsin Department of Transportation (WisDOT) and Wisconsin Highway Research Program (WHRP) for allowing us to use the data set.

Declaration of Competing Interest

There are no competing interests in relation to the work described.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.conbuildmat.2019.05.006.

References

[1] Z.H. Duan, S.C. Kou, C.S. Poon, Prediction of compressive strength of recycled aggregate concrete using artificial neural networks, Constr. Build. Mater. 40 (2013) 1200–1206, https://doi.org/10.1016/j.conbuildmat.2012.04.063.

[2] I.-C. Yeh, Modeling Concrete Strength with Augment-Neuron Networks, J. Mater. Civ. Eng. 10 (1998) 263–268, https://doi.org/10.1061/(ASCE)0899-1561(1998) 10:4(263).

[3] I.-C. Yeh, Analysis of Strength of Concrete Using Design of Experiments and Neural Networks, J. Mater. Civ. Eng. 18 (2006) 597–604, https://doi.org/10.1061/(ASCE)0899-1561(2006) 18:4(597).

[4] M. Uysal, H. Tanyildizi, Estimation of compressive strength of self compacting concrete containing polypropylene fiber and mineral additives exposed to high temperature using artificial neural network, Constr. Build. Mater. 27 (2012) 404–414, https://doi.org/10.1016/j.conbuildmat.2011.07.028.

[5] M. Uysal, H. Tanyildizi, Predicting the core compressive strength of self-compacting concrete (SCC) mixtures with mineral additives using artificial neural network, Constr. Build. Mater. 25 (2011) 4105–4111, https://doi.org/10.1016/j.conbuildmat.2010.11.108.

[6] American Association, of State Highway and Transportation Officials, Mechanistic-Empirical Pavement Design Guide: A Manual of Practice, 2nd ed., American Association of State Highway and Transportation Officials, Washington D.C., 2015. https://www-worldcat-org.ezproxy.library.wisc.edu/title/mechanistic-empirical-pavement-design-guide-a-manual-of-practice/oclc/974387504&referer=brief_results (accessed December 5, 2017).

[7] K. Hall, S. Tayabji, Coefficient of thermal expansion in concrete pavement design, ACPT Techbrief No. FHWA-HIF-09-015. (2011) 1–6.

[8] C. Zhou, X. Shu, B. Huang, Predicting concrete coefficient of thermal expansion with an improved micromechanical model, Constr. Build. Mater. 68 (2014) 10–16, https://doi.org/10.1016/j.conbuildmat.2014.06.039.

[9] S.-H. Kim, Determination of Coefficient of Thermal Expansion for Portland Cement Concrete Pavements for MEPDG Implementation, 2012. http://www.dot.ga.gov/BuildSmart/research/Documents/10-04.pdf (accessed December 5, 2017).

[10] J. Yang, S.H. Kim, Factorial effects of mix design variables on the coefficient of thermal expansion of concrete mixtures, Road Mater. Pavement Des. 15 (2014) 942–952, https://doi.org/10.1080/14680629.2014.920272.

[11] I.-C. Yeh, Modeling of strength of high-performance concrete using artificial neural networks, J. Civ. Eng. 28 (1998) 1797–1808, https://doi.org/10.1017/CBO9781107415324.004.

[12] H.-G. Ni, J.-Z. Wang, Prediction of compressive strength of concrete by neural networks, Cem. Concr. Res. 30 (2000) 1245–1250, https://doi.org/10.1016/S0008-8846(00)00345-8.

[13] H. Mashhadban, S.S. Kutanaei, M.A. Sayarinejad, Prediction and modeling of mechanical properties in fiber reinforced self-compacting concrete using particle swarm optimization algorithm and artificial neural network, Constr. Build. Mater. 119 (2016) 277–287, https://doi.org/10.1016/j.conbuildmat.2016.05.034.

[14] Y. Reich, Machine learning techniques for civil engineering problems, Comput. Civ. Infrastruct. Eng. 12 (1997) 295–310, https://doi.org/10.1111/0885-9507.00065.

[15] J.B. Effinger, R. Li, J. Silva, S.M. Cramer, Laboratory Study of Concrete Properties to Support Implementation of the New AASHTO Mechanistic Empirical Pavement Design Guide, Report No. 0092-10-11, Wisconsin Department of Transportation, 2012.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[17] L. Wang, X. Zang, Implementation of a scalable decision forest model based on information theory, Expert Syst. Appl. 38 (2011) 5981–5985, https://doi.org/10.1016/j.eswa.2010.11.023.

[18] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, 1984. doi: 10.1371/journal.pone.0015807.

[19] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106, https://doi.org/10.1023/A:1022643204877.

[20] F. Pedregosa, O. Grisel, R. Weiss, A. Passos, M. Brucher, Scikit-learn: machine learning in python, J. Mach. Learn. Res. (2011).

[21] K. Kim, A hybrid classification algorithm by subspace partitioning through semi-supervised decision tree, Pattern Recognit. 60 (2016) 157–163, https://doi.org/10.1016/j.patcog.2016.04.016.

[22] G. Louppe, Understanding random forests: from theory to practice, (2014). doi: 10.13140/2.1.1570.5928.

[23] M. Stone, Regression models and life-tables: D. R. Cox Source: Journal of the Royal Statistical Society. Series B (Methodological), 34 (2) Published by: Wiley for the Royal Statistical Society Stable, URL: http://www.jstor.org/stable/, J. R. Stat. Soc. 36 (1974) 111–147.

[24] C.F.J. Fabozzi, S.M. Focardi, S.T. Rachev, B.G. Arshanapalli, Financial Econometric Basics: Concepts and Real-World Applications, John Wiley & Sons, Incorporated, 2014.

[25] ARA–ERES, Guide for Mechanistic–Empirical Design of New and Rehabilitated Pavement Structures, NCHRP Project 1-37a, Final Report, National Cooperative Highway Research Program, Transportation Research Board, Washington, DC, 2004.