Transfer (machine) learning approaches coupled with target data augmentation to predict the mechanical properties of concrete

Emily Ford[*], Kailasnath Maneparambil[†,‡], Aditya Kumar[§], Gaurav Sant[**], Narayanan Neithalath[††]

[*] Graduate student, School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe AZ 85287, United States of America; elford1@asu.edu

[†] Intel Corporation, Chandler AZ 85224, United States of America

[‡] Adjunct Faculty, Computer Science and Engineering, Arizona State University, Tempe AZ 85287, United States of America; Kailasnath.Maneparambil@asu.edu

[§] Assistant Professor, Department of Materials Science and Engineering, Missouri University of Science and Technology, Rolla, MO 65409, United States of America; kumarad@mst.edu

[**] Professor and Henri Samueli Fellow, Department of Civil and Environmental Engineering, University of California Los Angeles, Los Angeles CA 90095, United States of America; gsant@ucla.edu

[††] Professor, School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe AZ 85287, United States of America. [*]Corresponding author; Narayanan.Neithalath@asu.edu; Phone: +1-480-965-6023; Fax: +1-480-965-055

Abstract

Transfer learning, a machine learning technique which employs prior knowledge from solving a source problem to solve a related target problem, is utilized in this work to predict the compressive strength and modulus of elasticity of different concrete mixtures. The use of data augmentation through empirical models to estimate missing data outputs allows for the use of inductive parameter transfer-learning artificial neural network (ANN) models for fast convergence. The paper considers two distinct cases: one where the domain of the target lies somewhat outside that of the source – termed domain expansion, and another where the target output (e.g., elastic modulus) is different, but related to the source output (e.g., compressive strength) – termed domain adaptation. Transfer learning is found to be most accurate when the source dataset is more complex than the target dataset, since more features could be learned. Data augmentation and the coupling of traditional machine learning with transfer learning are demonstrated to greatly enhance the predictive capability for important concrete properties, from mixture proportions. Limited experimental data can be used to transfer-learn the properties (output) of a new data set from a reliable source model for a related system.

Keywords: Machine Learning; Transfer Learning; Data Augmentation; Elastic Modulus; Compressive Strength; High Performance Concrete

1.0 Introduction

Mixture design and testing of concrete to ensure that the design specifications are met, is a time-consuming and expensive process that must take into consideration the properties of constitutive materials and their interactions (Young et al., 2019). To better model such complex, multi-dimensional relationships that cannot easily be fitted using traditional statistical and regression methods, machine learning (ML) methods provide improved options (Bock et al., 2019). ML models, when trained on large datasets: (a) utilize the semi-empirical rules that inform the relationship between constitutive materials and concrete properties, and (b) perform predictions on previously untrained datasets. A significant number of previous publications have dealt with the prediction of compressive strength and elastic modulus of conventional, high-performance, or geopolymer concrete from mixture proportions using a variety of ML models (Young et al., 2019; DeRousseau et al., 2019; Chou et al., 2011; Dao et al., 2020; Nguyen et al., 2020; Han et al., 2020). Additionally, complex non-linear data sets such as solar radiation or wind speed predictions have demonstrated the need to move beyond simple single ML models and evolve into hybrid models incorporating preprocessing, multi-model implementation, and meta-heuristic optimization (Karasu & Altan, 2019; Altan et al., 2021).

ML is dependent on large, high-quality datasets that may not be available due to time-and-cost limitations. When data is missing or unavailable, data augmentation techniques can be performed to fill the gaps, thereby allowing the ML training to proceed (Altan et al., 2021; Farhan, 2015; Ohno, 2020). Statistics-based approaches to data augmentation include mean substitution, interpolation, regression, and stochastic regression, which are known to produce biased estimates that reduce the variance of the dataset (Farhan, 2015). Modern strategies focus on generating data via ML such as adversarial nets to create thousands of additional data points, but at the risk of being unable to converge to the true model (Ohno, 2020). Another method is to use empirically established relationships between variables. The incorporation of “crude estimation properties” from empirical models and limited experimental measurements have been shown to significantly improve the accuracy of ML models to predict material behavior based on small datasets (Zhang & Ling, 2018).

After training, traditional ML models are only able to predict on untrained datasets with identical input ranges and relationships between variables. Thus, when the proportions and types of ingredients (e.g., cement and supplementary cementing material chemistry, admixtures, aggregate type, etc.) change, previously trained ML models would not be able to adapt to the changes. A solution to further the applications of ML to aid in the design and optimization of a wide variety of concrete mixtures is the use

of transfer learning (TL), which applies prior knowledge from an established source model to a new dataset. Transfer learning operates on the assumption that the source and target datasets lie within a similar domain space and that their input and output variables have similar relationships (Yamada et al., 2019; Gardner et al., 2020). The strength of TL is the ability to attain significant improvement in predictive capabilities using a pre-trained model as a basis, compared to only using the limited data available for training and testing. Recent research has used TL to predict building energy consumption (Chen et al., 2020; Fan et al., 2020), structural health monitoring (Gardner et al., 2020), product defect detection (Li et al., 2020; Liu et al., 2019), and material properties (Yamada et al., 2019; Jha et al., 2019; Childs & Washburn, 2019). The complexity of the TL process depends on the output in the target dataset and the similarity of the domain (feature space) between the source and target datasets (Gardner et al., 2020). For example, a study predicting the specific heat capacity of 58 amorphous homopolymers found a significant reduction in error after first training their models on 15000−30000 instances of a larger database, then performing TL for the original 58 points (Yamada et al., 2019). Working with a successfully trained ML from a large database as the source model, TL allows the ML model to be adapted to the specific target concrete property prediction and optimization problem, even when only a small amount of data is available, as shown in the example above. For data that is especially expensive or cumbersome to obtain, TL offers a solution to successfully and meaningfully incorporate ML into the data processing effort. This aspect is utilized in our TL approach in this paper.

This study leverages the beneficial attributes of TL where a source model trained on an abundant dataset is used to predict the outcomes of a related target dataset. Domain expansion and domain adaptation transfer learning techniques are implemented on source ML models to predict unknown attributes of concrete mixtures in the target dataset. This work is motivated by the fact that, for concrete, more often than not, only a limited range of tests are carried out (e.g., compressive strength), and other parameters (e.g., elastic modulus, durability) are indirectly inferred from the strength results. Expanding the applicable range of mixture proportions through ML or enhancing the predictability of concrete properties with changes in constituents through ML models reduces the amount of expensive and time-consuming testing. Empirically informed models are utilized to augment the missing outputs in the data, and combined with TL, are shown to provide accurate predictions of f'_c and E. Here, ML modeling of concrete mixture proportions to predict desired mechanical properties is enhanced by filling missing data points with well-established empirical relationships and transfer-learned domain expansion and adaptation results. This study combines ML with data augmentation and transfer learning to develop ML models uniquely able to predict material properties across a diverse range of inputs and types of concretes, thus

providing a methodology to tackle a wide variety of cross-property prediction problems related to concrete. Data augmentation informed by well-established empirical relationships and the coupling of traditional ML with TL are demonstrated to greatly enhance the predictive capability for important concrete properties from mixture proportions.

2.0 Datasets and Organization

Four different datasets, hereinafter termed as D-1, D-2, D-3, and D-4 are used in this study. The dataset D-1, consisting of 1030 individual data records, is obtained from (Yeh, 1998), which consists of 8 mixture proportioning features (e.g., contents of cement, slag, fly ash, water, coarse and fine aggregates, admixture, and age) of conventional concrete as inputs, and the compressive strength (f'_c) as the output. The dataset D-2 with 526 data records is based on (Han et al., 2020), where 13 mixture proportioning inputs of recycled aggregate concretes (RAC) (binder type, contents of cement, supplementary cementitious materials, water, coarse and fine aggregates, coarse recycled aggregate (RA), and maximum particle size, absorption and density of coarse aggregates and RA) are provided along with the elastic modulus (E) after 28 days of curing. The dataset D-3 with 228 data records is obtained from (Chopra et al., 2016), where 6 mixture proportioning inputs (contents of cement, fly ash, coarse and fine aggregates, water, and age) are provided along with f'_c after different ages of curing. Even though datasets D-1 and D-3 contain similar inputs and outputs, the range of some of the parameter values of D-3 lie outside the range of D-1. Finally, dataset D-4 with 104 data records is obtained from (Haranki, 2009) with 6 mixture proportioning inputs similar to D-1, along with the outputs f'_c and E.

The objective of this work is to use ML models: (i) trained on dataset D-1, to predict the compressive strength of RAC mixtures in dataset D-2, for which no strength data is available, (ii) trained on dataset D-2, to predict the elastic modulus of mixtures in dataset D-1, for which no modulus data is available, and (iii) trained on dataset D-1 to predict the compressive strength of mixtures in dataset D-3, many of which lie outside the range of D-1. The need for TL, rather than traditional ML to solve such problems is explained in the forthcoming section. Dataset D-4 is used as an independent validation set for the model of f'_c based on dataset D-1, to bring out the advantages and limitations of TL.

One of the important requirements in dealing with parameter-transfer learning approaches (detailed in section 3.2) is that the input parameters over different datasets be consistent. Careful observation of all the datasets showed that seven common inputs can be selected by carrying out some simple operations. The slag and fly ash contents in dataset D-1 are combined to form the supplementary cementitious materials (SCM) content, and the admixture content is ignored. Since RA is present in dataset D-2,

corresponding columns with values of 0 are added to D-1, D-3, and D-4, making the total inputs 7 for these datasets. For the dataset D-2, the aggregate sizes, densities, and absorption are ignored, to bring the total inputs to 7 as well. Table 1 shows the details of the 7 inputs and their statistics for all the datasets used in this paper. To ensure that any significant input parameters are not ignored, Pearson correlation coefficients (Chou et al., 2011; Konstantopoulos et al., 2020) were determined for datasets D-1 and D-2 (no input truncation was needed for D-3 and D-4) with the original 8 and 13 inputs, and the truncated 7 inputs. Pearson coefficient heat maps are shown in the Supplementary Information (Figures A1 and A2). The absolute values of the correlation coefficients with respect to E in dataset D-2 for most of the ignored terms were less than 0.10. Combining slag and fly ash into SCM for dataset D-1 resulted in a less than 0.03 change in the Pearson coefficient, indicating that all the major parameters were accounted for in the truncated datasets to be used for transfer learning. The predictive efficiency of the traditional ML models (i.e., predicting f'_c of D-1, or E of D-2 directly from the data) are slightly compromised by the input truncation, as will be shown later.

Table 1. Details of datasets used as inputs to the ML models. SCM is secondary cementitious material replacement with fly ash and/or slag, and RA is coarse recycled aggregate.

Data set	No. of Data Records	Output	Statistic	OPC ($\frac{kg}{m^3}$)	SCM ($\frac{kg}{m^3}$)	Water ($\frac{kg}{m^3}$)	Coarse Agg. ($\frac{kg}{m^3}$)	Fine Agg. ($\frac{kg}{m^3}$)	RA ($\frac{kg}{m^3}$)	Age (day)	Source
D-1	1030	f'_c	Max	540.0	382.0	247.0	1145.0	992.6	0.0*	365.0	(Yeh, 1998)
			Mean	281.2	128.1	181.6	972.9	773.6	0.0*	45.7	
			Min	102.0	0.0	121.8	801.0	594.0	0.0*	1.0	
D-2	526	E	Max	597.0	225.1	234.0	1950.0	1301.1	1800.0	28.0	(Han et al., 2020)
			Mean	338.7	32.3	170.7	563.1	730.7	495.4	28.0	
			Min	150.0	0.0	108.3	0.0	465.0	0.0	28.0	
D-3	228	f'_c	Max	475.0	71.3	263.9	1253.8	641.8	0.0*	91.0	(Chopra et al., 2016)
			Mean	433.9	24.0	213.7	1050.9	524.3	0.0*	58.3	
			Min	350.0	0.0	178.5	798.0	176.0	0.0*	28.0	
D-4	104	f'_c and E	Max	474.6	273.5	164.9	1231.0	715.5	0.0*	91.0	(Haranki, 2009)
			Mean	280.7	162.0	155.3	1071.1	615.2	0.0*	32.6	
			Min	116.9	73.0	139.6	735.1	506.1	0.0*	3.0	

* RA not part of mixture proportioning for D-1, D-3, and D-4 datasets; so a value of 0 is used.

3.0 A Brief Overview of Machine Learning (ML) and Transfer Learning (TL)

The traditional ML techniques used in this work are artificial neural networks (ANN) and the ensemble methods, which have been used in the prediction of concrete strength and elastic modulus (Young et al., 2019; DeRousseau et al., 2019; Chou et al., 2011; Dao et al., 2020; Nguyen et al., 2020). Since TL is rather

new in the field of general materials science, and to the authors' knowledge has never been used for concrete property prediction, the essential concepts and background are introduced.

3.1 Artificial Neural Networks (ANN) and Ensemble Methods

An artificial neural network (ANN) is organized into an input layer, hidden layer(s), and an output layer. Each layer contains neurons with values of information that typically range between 0 and 1 (Li et al., 2019). The ANN can be shallow with only a few hidden layers, or deep with many layers. Generally, networks with > 10 hidden layers are considered deep networks (Schmidhuber, 2015). The ANNs presented in this study utilize 1-3 hidden layers, which are appropriate for the number of unique data records used. The value within each neuron of the hidden layer(s) and output layer depends on the previous neurons, the weights, and the chosen activation function (Li et al., 2019). Sigmoidal function or rectified linear unit (ReLu) is commonly used as the activation function (Young et al., 2019; Li et al., 2019; Son et al., 2012). Based on a previous work by the authors (Ford et al., 2020), this study uses the ReLu. Optimization is performed using RMSprop, which features an adaptive learning rate formula (Tieleman & Hinton, 2012). More details on common ANNs can be found in (Schmidhuber, 2015; Carpenter & Hoffman, 1997). Neural networks feature many fitting parameters that allow them to predict nonlinear interactions. A disadvantage of neural networks it the potential for over-fitting the data, or training the weights to precisely match the training dataset and render the algorithm unable to accurately predict results of the test dataset (Young et al., 2019). To minimize over-fitting, a dropout rate was incorporated into the ANN (Srivastava et al., 2014). While testing, the entire neural network is used, but the connection weights are multiplied by the dropout rate to combine the effect of the thinned-out training networks. In this study, the Keras neural network framework written in Python is utilized to build and train the ANNs (Chollet, 2015). TensorFlow is used as the back-end engine. Additionally, a wrapper was implemented to use the scikit learn GridSearchCV and RandomizedSearchCV functions (Pedregosa et al., 2011) in order to determine and compare hyperparameter settings, as detailed in Section 4.2.

Machine learning forest ensemble methods are based on the structure of a decision tree (Young et al., 2019; Oey et al., 2020). A basic form of forest ensemble is the Random Forest (RF), in which the best split of the data into branches and nodes is determined by considering all of the input features and checking a criterion, such as mean-squared error, to select the most discriminative threshold (Pedregosa et al., 2011; Oey et al., 2020). Each individual decision tree in the RF ensemble does not use the entire set of training data, but a bootstrap sample made from subsets of the training data with replacement (Oey et al., 2020). Another forest ensemble is the Extra Trees (ET) regressor in which the splits are drawn at random for each

feature and the best split, as measured by the chosen criteria, is selected as the splitting rule (Pedregosa et al., 2011; Oey et al., 2020). In the ET regression model, the entire dataset is incorporated into each individual tree (Pedregosa et al., 2011). The prediction results of the individual trees are averaged to produce the output prediction in the RF and ET regressions. In a Gradient Boosted Tree (GBT) ensemble, an initial tree is trained with the entire dataset. All subsequent trees in the forest are trained to minimize the residual (least squares error) between the predicted and actual values of the previous tree via steepest gradient descent (Young et al., 2019; Pedregosa et al., 2011). The final prediction is calculated as the weighted sum of the predictions of each tree, where for each tree beyond the first, the prediction is multiplied by the learning rate (Young et al., 2019).

3.2 Transfer Learning

Traditional ML deals with training separate, isolated models on datasets, as shown in Figure 1(a). No knowledge is retained which can be transferred from one model to another. In TL, knowledge (features, weights, etc.) from previously trained models can be leveraged to train newer models. Thus, TL offers significant improvement in predictive capabilities for small datasets using pre-trained models as a basis, compared to only using the limited data available for testing and training in traditional ML (Yamada et al., 2019; Gardner et al., 2020). Transfer learning is premised on the fact that prior knowledge from one domain and task can be applied to another domain and task, as shown in Figure 1(b). The domain $\mathcal{D} = \{\mathcal{X}, p(\mathcal{X})\}$, where $\mathcal{X}$ represents the feature space. Here feature space includes the input parameters, such as the contents of different ingredients, age, etc. as shown in Table 1, and $\mathrm{p}(\mathcal{X})$ represents the marginal probability distribution. A task in a domain, $\mathcal{T} = \{\mathcal{Y}, \mathrm{f}(.)\}$, where $\mathcal{Y}$ is the label space (the output f'$_c$ or E vectors), and $f(.)$ is the predictive function that relates the features and the labels (Gardner et al., 2020; Yun et al., 2019; Weiss et al., 2016). Given a source domain $\mathcal{D}_S$ with a corresponding task $\mathcal{T}_S$, and a target domain $\mathcal{D}_T$ with a task $\mathcal{T}_T$, the goal of TL is to improve the target predictive function $f_T(.)$ by using knowledge learned from $\mathcal{D}_S$ and $\mathcal{T}_S$. Here, $\mathcal{D}_S \neq \mathcal{D}_T$ and/or $\mathcal{T}_S \neq \mathcal{T}_T$. In fact, if $\mathcal{D}_S = \mathcal{D}_T$ and $\mathcal{T}_S = \mathcal{T}_T$, this becomes the case of traditional ML (Weiss et al., 2016).

In the scope of this paper, let us consider the case of predicting f'$_c$ of dataset D-2, which is not available. However, a predictive function, $f_S(.)$, can be made available using traditional ML (e.g., ANN) that relates the f'$_c$ of a different domain and its features (dataset D-1). Thus $\mathcal{D}_S$ corresponds to dataset D-1 and $\mathcal{D}_T$ to dataset D-2. While the feature space, $\mathcal{X}$ (input parameters) is the same in the source and target, their marginal probability distributions, $p(\mathcal{X})$, are different, owing to the values of the inputs and the relationships between the inputs being different in both the datasets (e.g., the use of RA in D-2

fundamentally alters the relationships between various inputs in D-2, as compared to D-1). Thus $\mathcal{D}_S \neq \mathcal{D}_T$. Since the same $f_S(.)$ from dataset D-1 cannot be used to predict the f'$_c$ of dataset D-2 (as will be shown later), $\mathcal{T}_S \neq \mathcal{T}_T$. TL synthesizes different domains as a whole, enabling the model to acquire knowledge from the source domain (e.g., predicting f'$_c$ of dataset D-1) and improve its performance to predict the target task (predicting f'$_c$ of dataset D-2), so that a more efficient scheme can be generated to predict the labels for an expanded domain, which is more useful in a practical sense. A similar approach can be used to predict the E of dataset D-1 based on information from dataset D-2. Figure 1(b) shows the application of TL to the problem of interest.

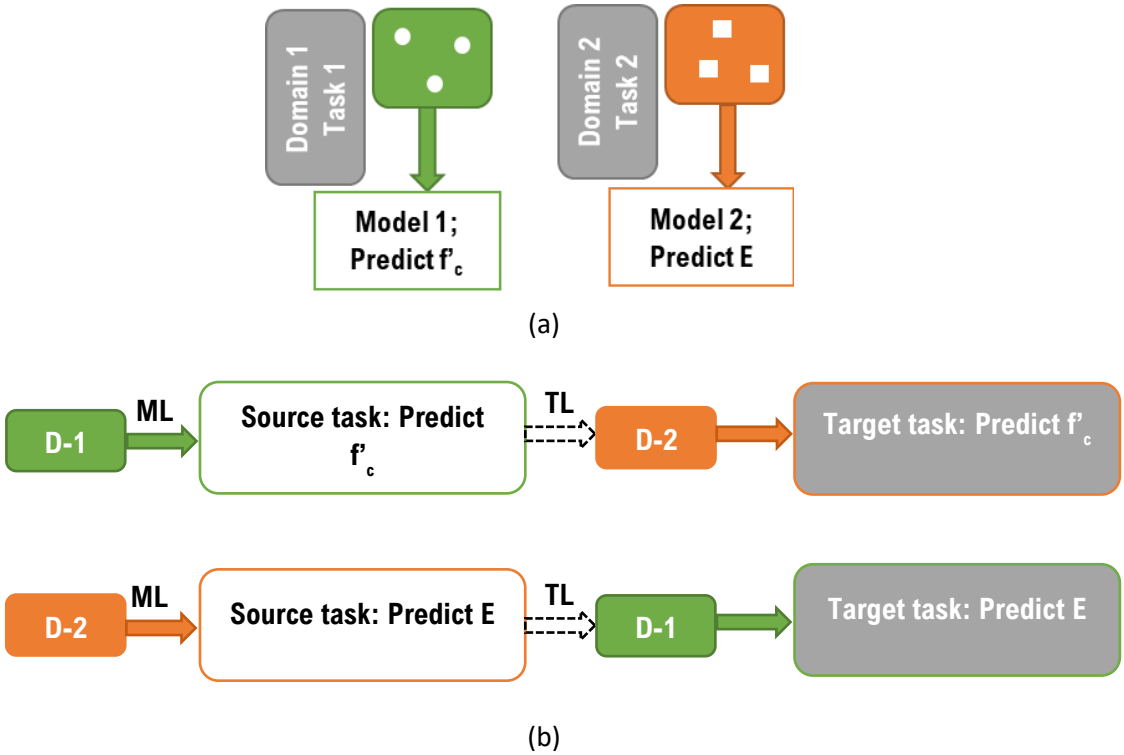


Figure 1. (a) Traditional ML that employs separate, isolated models, and (b) transfer learning where the knowledge gained from one task is utilized for another

Since the feature spaces are similar in the source and target domains, one can consider this to be a case of homogeneous transfer learning (Weiss et al., 2016). The presence of labeled data in the source domain (for e.g., f'_c of dataset D-1, and E of dataset D-2 are available), its absence in the target domain (f'_c of dataset D-2, and E of dataset D-1 are not available), and $\mathcal{D}_S \neq \mathcal{D}_T$ makes this a case of transductive transfer learning (Bahadori et al., 2014), though there are still inconsistencies in the terminologies used in literature (Gardner et al., 2020; Fan et al., 2020; Li et al., 2020; Weiss et al., 2016; Pan & Yang, 2010). Transductive TL is a highly challenging problem to solve (Arnold et al., 2007; Ohno, 2019) and is computationally expensive. Hence, this study converts this into a more efficient inductive TL problem, where labeled data is available in both the source and target domain (Li et al., 2020; Yun et al., 2019). In this case, labeled data that is missing from the target tasks is supplemented by empirically-derived f'_c-E relationships. Inductive learning allows for the use of a more intuitive parameter transfer approach that focuses on shared parameters between the source and target domains or prior distribution of hyperparameters (Li et al., 2020). Parameter transfer is applicable when the source model is used as a starting point that has a similar input-output relationship as the target data (Yamada et al., 2019; Gardner et al., 2020). Parameter transfer-learning architecture for an ANN is presented in

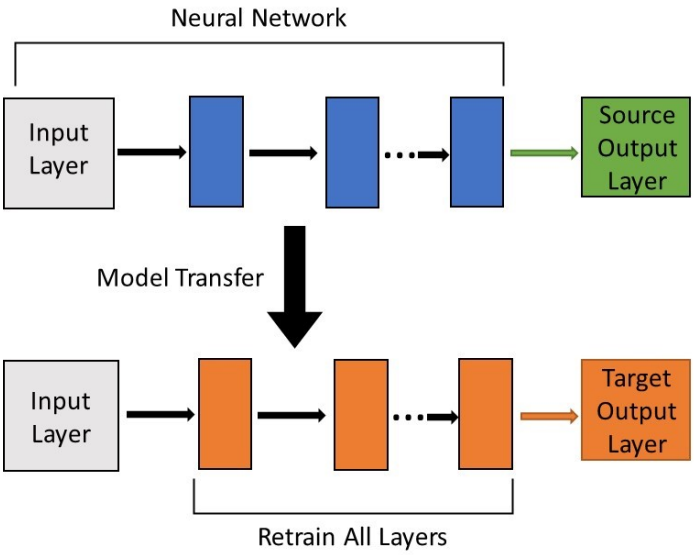


Figure 2, in which the source and target populations have the same number of inputs and outputs as well as the same type of inputs.

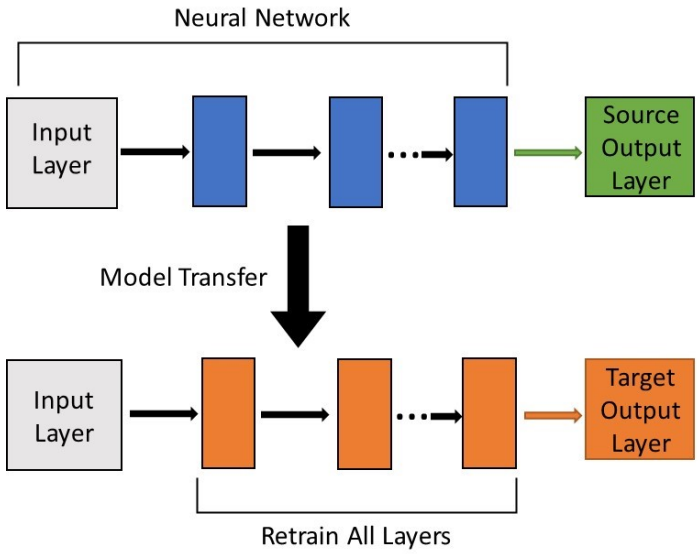


Figure 2. Schematic of parameter-transfer learning where the source model weights act as initial values for retraining to the target model.

4.0 Data Processing

Several data preprocessing and hyperparameter optimization methods are needed as part of implementing ML/TL techniques for concrete property prediction. They are discussed here.

4.1 Preprocessing and evaluation

First, each dataset shown in Table 1 was shuffled to provide a greater chance of an equal distribution of the various mixtures within the testing and training datasets. The entire dataset was split such that 25% of the points were assigned to the testing set and 75% of the points were assigned to the training set. An artifact of the weight assignments in ML is that larger values will inherently be given a larger weight, which can skew the prediction significantly (Oey et al., 2020). To address this, the input and output data points were pre-processed before separation into the testing and training sets. Equation 1, based on MinMaxScaler in scikit-learn (Pedregosa et al., 2011), was used on the data to ensure that all of the inputs and outputs lie in the range [0, 1].

$$z_{new} = \frac{z - z_{min}}{z_{max} - z_{min}} \quad (1)$$

where z_{new} is the value of the variable after transformation, z is the current value of the variable, z_{min} is the minimum value of that variable, and z_{max} is the maximum value of that variable. After training and

predicting, the normalized test data is converted back to the original scale using the "inversetransform" function.

Training was performed by fitting the ML algorithm to the training dataset, allowing the algorithm to adjust its internal features to minimize the error via backpropagation. Model performance was evaluated using the testing dataset, which the ML algorithm has not seen yet, and determining the resulting errors. Each instance that the entire set of data (testing and training) is processed through the ML algorithm is called an epoch. Selection of different hyperparameters (see next section for more details) was accomplished by utilizing the same number of epochs to train MLs and comparing their accuracy. In training, the goal of the ML models is to minimize the objective function, which was the mean squared error (MSE), given as:

$$\mathrm{MSE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{P_i} - \mathrm{A_i})^2 \quad (2)$$

where n is the total number of data points, $\mathrm{A_i}$ is the actual value, and $\mathrm{P_i}$ is the predicted value. Other metrics tracked, but not used to train the models, were the mean absolute error (MAE) and the coefficient of determination (R^2), given as:

$$\mathrm{MAE} = \frac{1}{\mathrm{n}}\left(\sum_{\mathrm{i}=1}^{\mathrm{n}}\left|\frac{\mathrm{A_i}-\mathrm{P_i}}{\mathrm{A_i}}\right|\right) \quad (3)$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{P_i}-\mathrm{A_i})^2}{\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{A_i}-\bar{\mathrm{A}})^2} = \left(\frac{\mathrm{n}(\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{A_i}*\mathrm{P_i})-(\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{A_i})*(\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P_i})}{\sqrt{\mathrm{n}(\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{A_i^2})-(\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{A_i})^2}*\sqrt{\mathrm{n}(\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P_i^2})-(\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P_i})^2}}\right)^2 \quad (4)$$

Where $\bar{\mathrm{A}}$ is the mean of the actual values.

4.2 Hyperparameter optimization

The three parameters to optimize in the ANN models were the number of hidden layers, the number of neurons in each hidden layer, and the dropout rate. ReLu activation function with a learning rate of 0.001 and an RMSprop optimization scheme with backpropagation was used. For the RF, ET, and GBT models, the number of trees in the forest were tuned. Coarse optimization of the hyperparameters followed a random search pattern, found to be the most efficient method to optimize parameters (Bergstra & Bengio, 2012), by randomly generating 20 different combinations of hyperparameters. The hyperparameters for random testing were chosen from the uniform distributions shown in Table 2.

Table 2. Hyperparameters tuned based on a uniform distribution range of potential values.

Model	Hyperparameter	Uniform Distribution Range
ANN	No. hidden layers	[1, 3]
	No. starting neurons	[10, 55]
	Drop rate	[0, 0.10]
Random Forest (RF), Extra Trees (ET), and Gradient Boosted Trees (GBT) forests	No. of trees	[50, 400]

To test the accuracy of the predictions under each set of test parameters, an n-fold cross-validation technique was employed (Young et al., 2019; Chou et al., 2011; Oey et al., 2020). A 3-fold cross-validation, deemed sufficient for the size of the datasets, was performed using the following steps: (i) randomizing the dataset and splitting into 3 folds, (ii) training the model with selected parameters using 2 of the folds, (iii) testing the model using the remaining fold, (iv) repeating steps (ii) and (iii) until each fold has been used for testing once, acquiring 3 independent performance measures, and (v) averaging the individual accuracy measures to obtain the cross-validation errors. The parameters which minimized the cross-validation error was used as a basis for the final models with some additional fine-tuning searches. A concise summary of the parameter selection, training, and testing procedures are shown in Figure A3 in the Supplementary Information (SI). The final hyperparameters chosen for each ML technique for each of the datasets shown is shown in Tables A3 and A4 of the Supplementary Information.

4.3 Empirical data augmentation and weight adjustment through backpropagation in TL

ML training and testing depends on large datasets with complete records of input and output information, that are expensive and often difficult to maintain (Farhan, 2015). Approaches such as maximum entropy methods have been used for unlabeled target data (transductive learning), which is computationally intensive (Arnold et al., 2007). On the other hand, parameter-transfer learning is a type of inductive transfer learning that uses trained weights from a labeled source model as the initial weights for the training of a labeled target dataset. Performing inductive transfer learning on unlabeled target datasets requires data augmentation to provide inexpensive and generalizable information that allows the models to accurately converge. Traditional approaches that include mean substitution, interpolation, regression, and stochastic regression are known to produced biased estimates that reduce the variance of the dataset, while modern strategies such as adversarial nets sometimes are unable to converge to the true model (Ohno, 2020). Thus, to provide labeled target data for inductive transfer, we make use of the

empirically established relationships between compressive strength and elastic modulus for different types of concretes, reported in several past studies and code specifications. Such information from empirical models and non-expensive experimental measurements has been shown to significantly improve the accuracy of ML models to predict material behavior based on small datasets (Zhang & Ling, 2018). This data augmentation to change the target from unlabeled to labeled enables TL to arrive at better predictions, more easily.

For dataset D-1, the estimation of E as a function of f'_c was obtained from several international codes, including American Concrete Institute's ACI 318-19 (American Concrete Institute [ACI], 2014), Canadian Standards Association's CSA A23 (Canadian Standards Association [CSA], 2004), the Indian Standard IS 456 (Indian Standard [IS], 2007), and the Eurocode 2 (ENV 1992-1-1, 2004). For dataset D-2 on RAC, the average of the E-f'_c relationships given in ACI 318-19 (ACI, 2014), along with two other published models for RACs (Sadati et al., 2019) were used. These empirical estimations serve as a guideline for the ML models to learn the relationship between the inputs and the desired output mechanical properties. For example, the ML model developed to predict E of dataset D-2 is retrained via TL using the feature space of D-1 and E from empirical models relating E to the known f'_c of D-1. The presence of augmented data leads to faster convergence (error minimization) by changing the weights of the initial models for E or f'_c through backpropagation. However, one needs to be cognizant of the fact that the quality of the augmented data and the strength of the relationships that are used in creating augmented data influences the predictions, as will be shown later. Supplementary Information (SI) contains more details on the E and f'_c relationships utilized.

As discussed previously, training of ANN neuron weights is performed in this study using backpropagation on a RMSprop optimization scheme. Backpropagation fundamentally relies on the gradient of the objective function with respect to the current weight, and thus is a function of both the input and output values (Tieleman & Hinton, 2012). When parameter-transfer learning is performed, a source model with ANN structure and weights already generated is re-trained to the target model (Yamada et al., 2019; Gardner et al., 2020). This means that the target inputs and outputs are run through the model and the objective function gradients and backpropagation are now performed with respect to the target dataset. In this way, the source model acts as initialized weights based on the source data rather than the usual randomized weights when first generating an ANN model.

4.4 Domain expansion and domain augmentation TL for strength and elastic modulus prediction

Figure 3 depicts the different ML and TL models developed, along with the datasets used in traditional ML and TL. First, traditional ML ANN models were tested and trained on the original datasets using all inputs (8 for dataset D-1 and 13 for dataset D-2), similar to those reported in (Han et al., 2020; Yeh, 1998). Next, new ANN models were developed to predict f'_c and E of datasets D-1 and D-2 based on the truncated parameter space consisting of the designated seven inputs. These ANN models were then used as a basis for parameter-transfer learning.

The first transfer model developed was from dataset D-1 to D-3, which both utilize experimental f'_c outputs; however, D-3 featured only ~20% the amount data of D-1, as well as an expanded range of input variables outside the scope of D-1. Hence this approach is termed as "domain expansion" TL. Next, a TL model was developed for f'_c of mixtures in dataset D-2 based on retraining the weights of the source ML model for f'_c developed from dataset D-1. Data augmentation, using the average f'_c obtained from three empirical models described in the previous subsection relating the experimental E of dataset D-2 to f'_c (termed hereinafter as model-averaged data), was used to facilitate retraining. This resulted in ANN models that could predict the missing output parameter (f'_c) of the dataset D-2. Since the model developed on D-1 was used to predict the strength of D-2, this approach is termed "domain adaptation". A similar approach was used to predict the E of dataset D-1 using a source model trained on dataset D-2.

After TL was performed on the individual datasets D-1 and D-2, they were combined (1556 data records, 7 inputs), and E and f'_c of these mixtures were predicted. Note that, in the combined datasets, E is not available for D-1, and f'_c for D-2. Hence, two approaches were used for data augmentation in this case. The missing labels were provided using: (a) empirically-derived model-averaged data, and (b) transfer-learned E and f'_c from the domain adaptation models. In the combined dataset, traditional ML techniques can be used to carry out the predictions; we have used ANN and ensemble methods in this paper. In the event that a few experimentally obtained E (of dataset D-1) and f'_c (of dataset D-2) are available, the TL approaches described in this paper also could be used efficiently for new predictions, since TL is known to work very well with small datasets (Zhang & Ling, 2018; Yamada et al., 2019; Gardner et al., 2020) as will be shown in a later section.

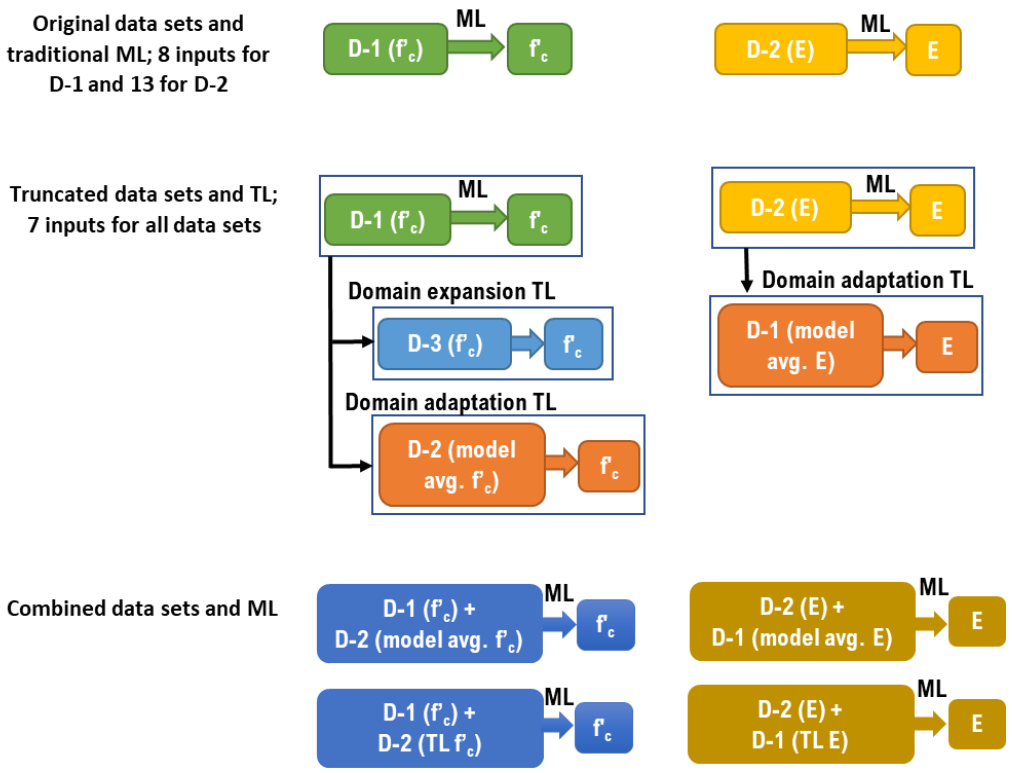


Figure 3: Illustration of the use of three datasets to carry out ML and TL to predict the strength and elastic modulus of concrete mixtures. The larger cells show the source domain used for training, along with the output in parenthesis; D-1 (f'_c) means the source domain is D-1 with f'_c data. Similarly, D-1 (TL E) means the data came from D-1 with the transfer learned E from the domain adaptation model. The smaller cells show the target output.

5.0 ML and TL Predictions and Evaluation of Predictive Efficiencies

5.1 ML ANN models on complete datasets

For the original datasets corresponding to D-1 and D-2, ANN models were used to predict the compressive strength and elastic modulus respectively, utilizing all the input parameters as well as the truncated set of parameters. Figure 4(a) shows that the developed ANN model is capable of accurately predicting the f'c of the entire dataset D-1 (8 parameters). Similar predictive efficiencies have been reported for this dataset using other ML methods also (Young et al., 2019; Yeh, 1998). When a truncated dataset is used (7 parameters) as shown in Table 3 and Figure 4(b), the predictive efficiency of the ANN model for the dataset D-1 is not compromised. Similar analysis is shown for the dataset D-2 in Figure 4(c) and (d). Using all the 13 input parameters as reported in (Han et al., 2020), the ANN model is capable of accurately

predicting E, whereas truncating the parameter space to 7 inputs results in a reduction in the R^2 value (and increase in RMSE and MAE; see Table 3) as shown in Figure 4(d). This reduction is not surprising since six of the parameters from the original model were not considered in the truncated model, due to the need to keep the parameter space the same for parameter-transfer learning. A look at the Pearson correlation coefficients (Figures A1 and A2 in SI) shows that ignoring the maximum coarse aggregate and RA size information is likely the reason for this efficiency loss. Table 3 lists the other metrics including the root mean square error (RMSE) and the mean absolute error (MAE) for all the models.

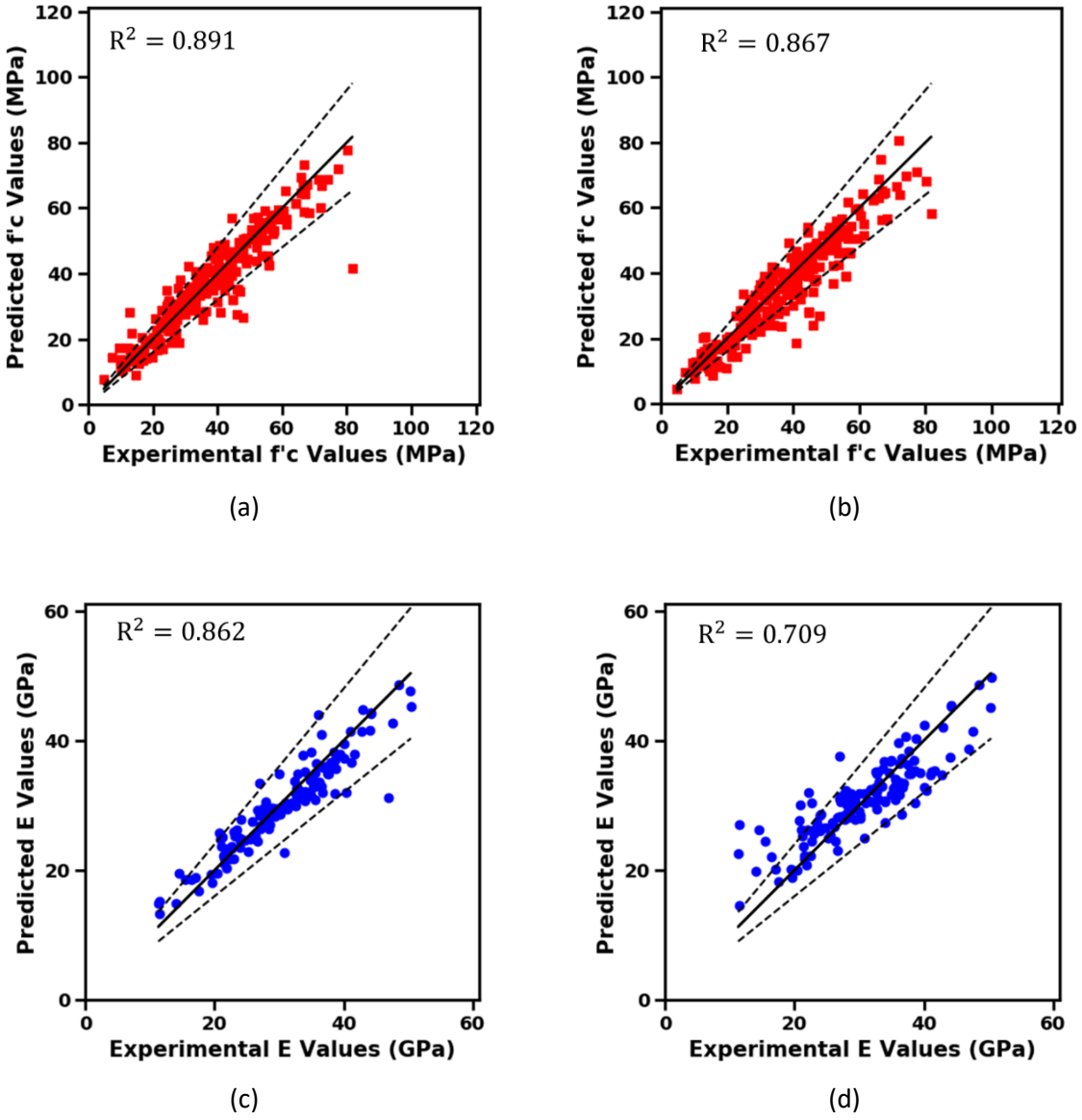
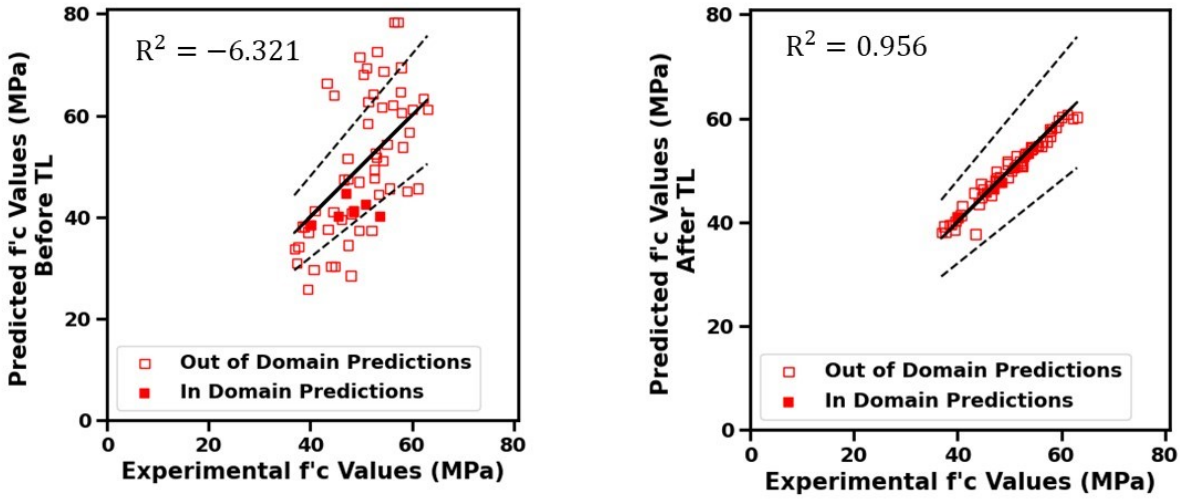


Figure 4. Prediction of mechanical properties using ANN: (a) dataset D-1 with original 8 inputs, (b) truncated dataset D-1 with 7 inputs, (c) dataset D-2 with original 13 inputs, and (d) truncated dataset D-2 with 7 inputs. The solid line represents the line of ideality, and the dashed lines represent a ± 20% bound. Testing data that represents 25% of the datasets is shown. R^2 for test data is reported.

The ANN model based on dataset D-1 is developed for f'_c of conventional concretes whereas the one based on D-2 is developed for E of RAC. It should therefore be obvious that the model for f'_c cannot be directly used to predict the f'_c of dataset D-2, since the marginal probability distributions of the domain from which the source model was generated and that of the target domain are different. The same is the case with the model for E. This necessitates the use of transfer learning, the results of which are discussed in the forthcoming sections.

5.2 Transfer learning for domain expansion

To first demonstrate the capabilities of parameter-transfer learning, we consider the domain expansion case, from dataset D-1 to D-3. The ANN based model developed using dataset D-1 was re-trained to predict f'_c of the dataset D-3. As shown in Table 1, the some of the parameter values of dataset D-3 (e.g., water, coarse aggregate, and fine aggregate contents) lie outside the range of the dataset D-1. This form of transfer learning represents a domain expansion, where a more generalized model is created from a restricted domain and retrained to account for the expanded domain. Figure 5(a) shows the strength prediction using the ANN model without TL (i.e., the model for D-1 is directly used), where it is seen that a large number of data points in dataset D-3 that do not belong to the domain of D-1 are poorly predicted. The ANN model had sporadic predictions for points lying outside of the training input domain, under- and over-predicting significantly. When the model weights were adjusted through TL, the predictive quality improved tremendously as shown in Figure 5(b). The metrics used to evaluate the performance of the models are shown in Table 3. This shows the success of taking a source model and expanding its domain via transfer learning. Even with a target dataset featuring significantly less number of data points, the fundamental relationship between the input and output variables was similar enough that TL was able to produce a very accurate model for the dataset D-3. This is one of the main advantages of TL, where new models can be trained using smaller datasets.



(a) (b)

Figure 5. ANN estimation of f'_c of dataset D-3 using the model developed from dataset D-1 as the source: (a) prior to transfer learning, and (b) after domain expansion transfer learning. The solid line represents the line of ideality, and the dashed lines represent a ± 20% bound. Testing data that represents 25% of the D-3 dataset is shown. R^2 for test data is reported.

5.3 Data augmentation-assisted transfer learning for datasets D-1 and D-2

This section addresses TL between a source model and a data-augmented target model, where the fundamental nature of concrete is different between the datasets D-1 and D-2, primarily through the use of RA in D-2 and the differences in cement-replacement materials. Parameter-transfer learning was performed using the 7-input ANN models as the source models and then retraining on the model-averaged data, as explained in Section 4.4. Figure 6(a) shows the model-averaged f'_c of dataset D-2 plotted against the predicted f'_c through domain adaptation TL. Note that the source model developed based on dataset D-1 has been retrained to predict the strength of dataset D-2. The predicted f'_c values are plotted against the known experimental E values of dataset D-2 in Figure 6(b). The model-averaged fit is also shown in this figure. Using model-averaged E of dataset D-1 as training labels, the predicted E values of dataset D-1 are shown in Figure 6(c). Figure 6(d) plots the relationship between experimental f'_c of dataset D-1 and its predicted E. Table 3 summarizes the metrics for the original and transfer models for the f'_c and E outputs.

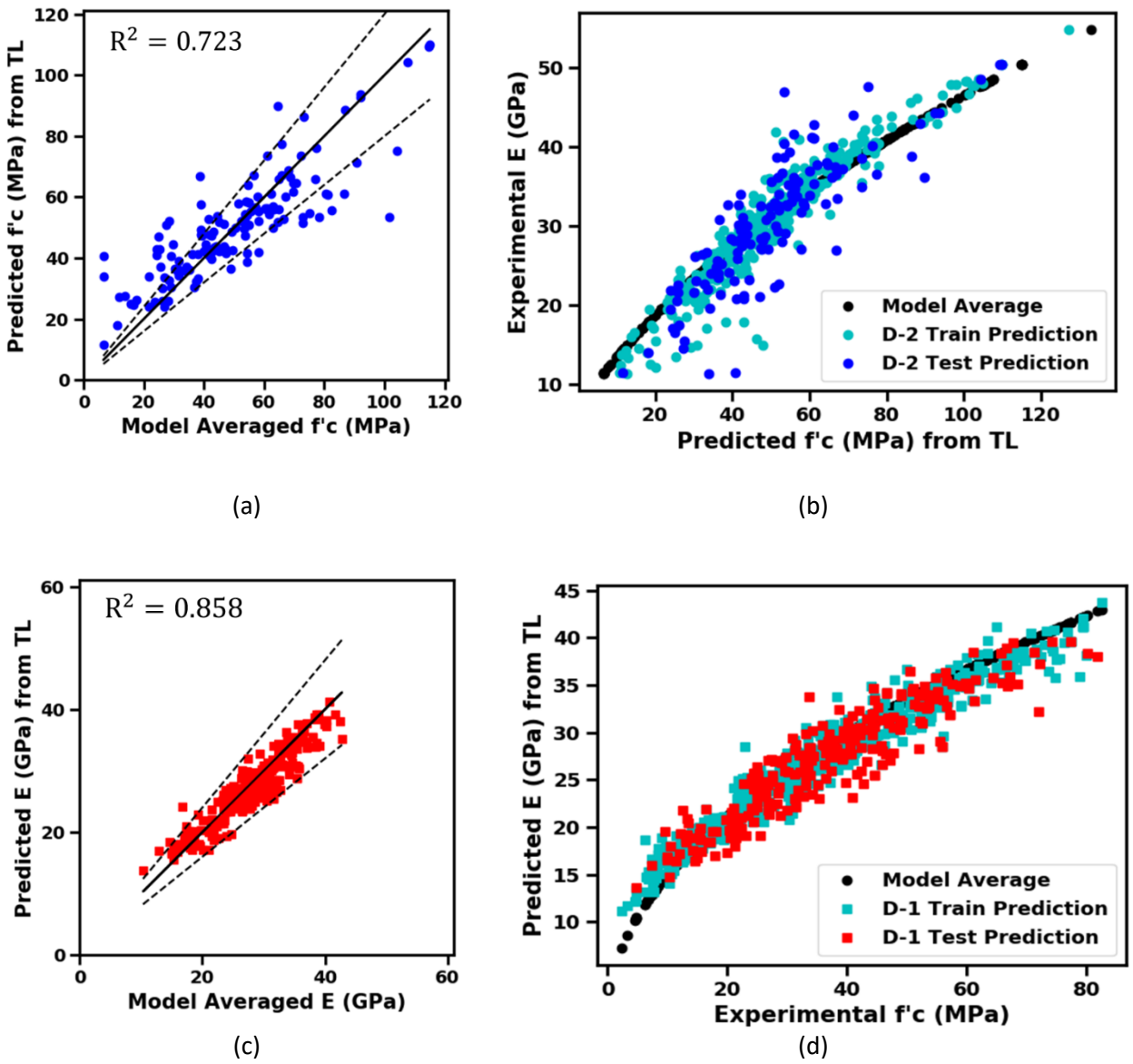


Figure 6. (a) f'_c prediction of D-2 using model-averaged f'_c to retrain the model developed for D-1, (b) relationship between predicted f'_c and experimental E for D-2, (c) E prediction of D-1 using model-averaged E to retrain the model developed for D-2, and (d) relationship between experimental f'_c and predicted E for D-1. In (a) and (c) the solid line represents the line of ideality, and the dashed lines represent a ± 20% bound. Testing data that represents 25% of the datasets is shown in (a) and (c). R^2 for test data is reported.

Table 3. Average and standard deviation values over three consecutive runs of ANN models trained with the original inputs, the common 7 inputs, and the transfer datasets.

Output	Dataset	No. of Data Points	No. of Inputs	RMSE	MAE	R^2
f'c (MPa)	D-1	1030	8	5.16 ± 1.91	3.74 ± 0.10	0.891 ± 0.015
	D-1	1030	7	5.70 ± 1.71	4.21 ± 0.19	0.867 ± 0.012
	Domain expansion TL on D-3	228	7	1.47 ± 0.29	1.11 ± 0.04	0.956 ± 0.002
	Domain adaptation TL on D-2	526	7	11.9 ± 2.81	8.92 ± 0.17	0.723 ± 0.015
E (GPa)	D-2	526	13	3.04 ± 0.90	2.19 ± 0.08	0.862 ± 0.012
	D-2	526	7	4.41 ± 1.19	3.26 ± 0.12	0.709 ± 0.021
	Domain adaptation TL on D-1	1030	7	2.36 ± 0.71	1.82 ± 0.09	0.858 ± 0.013

It is instructive to notice the differences in predictive abilities of the two domain adaptation TL models shown in Figure 6, based on the differences in the domains of these datasets (see Table 1). The cement, SCM, and water contents are similar between the two datasets, however, the D-2 dataset has a much greater range of coarse, fine, and RA contents, while the dataset D-1 has an expanded age range. When transferring the ANN model developed on dataset D-2 (which predicts E) to predict the E of dataset D-1, the absence of RA in dataset D-1 makes the model slightly over-parameterized. The fact that the model developed based on dataset D-2 has some features relating to D-1 also (e.g., the mixes with no RA content in D-2 are in theory, similar to those in dataset D-1) makes retraining more efficient. Moreover, dataset D-1 had about twice as many data points as dataset D-2, allowing for more data to be used in re-training the model. Parameter-transfer learning from D-2 to D-1 (to predict E; Figure 6(c)) ultimately generated an ANN model as accurate as the initial E prediction model that incorporated all 13 of the original D-2 dataset inputs (Figure 4(c)), as can be noticed in Table 3. On the other hand, the strength prediction of dataset D-2 based on transferring the model developed for D-1 is found to be less efficient than the case mentioned above, as can be seen in Table 3. In general, for efficient TL, the source model should typically be based on problems with greater diversity and coverage than the target model (Chen et al., 2020). In this case, the fundamental relationship between the input and output parameters captured by the model for dataset D-1 cannot account for the complexity present in the D-2 model, especially in terms of the RA content, which is well known to influence concrete properties (Han et al., 2020; Sadati et al., 2019). Overall, performing TL from an expanded to a more restricted domain, along with more data available for training, leads to improved modeling accuracy.

5.4 Combining datasets using empirical and transfer-learned labels for predicting f'_c and E using ML

Another approach to solving the problem of predicting mechanical properties of concrete mixtures using data augmentation of the type reported here is to no longer assume that the source and target domains are different, thus reducing the problem to one of traditional ML. This can be implemented once it is considered that the model-averaged data or transfer-learned data (shown in Section 5.3) are reasonable estimates of true target labels. This is not an unfair assumption, especially for the labels from the transfer-learned task as seen from Figure 6. The D-1 and D-2 datasets were thus combined, and the data entries randomly switched to form a larger database, with enhanced diversity of data. This combined database benefited from a larger amount of data to train, and an expanded domain, taking full advantage of the information on different aggregate types and different ages available with both the datasets. In the first full database, four ML techniques (one ANN and three ensembles) were used to predict f'_c and E from mixture proportions of datasets D-1 and D-2. The missing labeled data are replaced by the model-averaged f'_c or E in this case.

Figure 7 shows the predictive efficiencies of f'_c of the four ML models utilizing the model-averaged combined dataset, and Table 4 shows the metrics adopted to ascertain predictive efficiencies. Compared to the domain adaptation transfer learning models shown in Table 3, the use of combined dataset produces comparable accuracy for E prediction, and better accuracy for f'_c prediction. This can be partly attributed to the greater diversity and range in the input and output parameters compared to training (or re-training) on a single dataset as was carried out for domain adaptation. The use of a single database also allows for the use of ensemble methods. For predicting both E and f'_c the ensemble forest methods were found to be more accurate than the ANN models trained on the same database. A previous work on concrete mixture optimization found that forest models outperformed ANN in the case of unbalanced and discrete data (Zhang et al., 2020). This improvement was credited to the improved generalization in the forest methods, which reduces the instability of individual trees through random sampling of data and random selection of input features (DeRousseau et al., 2019; Su et al., 2015). Forest ensembles also offer insight into the relative importance of each of the inputs and an interpretable path – good for tracing dichotomies in data as seen with the incorporation (or lack) of RA in the mixture. This means forest ensembles are well-suited to regression problems where there are definitive splits in the mixtures utilized as inputs.

Table 4. ML results for the combined datasets with experimental and model-averaged data labels for f'_c and E prediction, with the average and standard deviation values over three consecutive runs shown. The most accurate ML model for each output is shown in **bold**.

Output	Model Type	RMSE	MAE	$\mathbf{R^2}$
f'c (MPa)	ANN	9.40 ± 1.83	6.53 ± 0.13	0.778 ± 0.008
	RF	7.64 ± 0.51	5.14 ± 0.05	0.853 ± 0.001
	ET	**7.25 ± 0.77**	**4.67 ± 0.04**	**0.868 ± 0.001**
	GBF	7.70 ± 0.62	5.15 ± 0.01	0.851 ± 0.001
E (GPa)	ANN	3.62 ± 0.24	2.53 ± 0.02	0.751 ± 0.001
	RF	3.20 ± 0.12	2.17 ± 0.00	0.806 ± 0.000
	ET	2.95 ± 0.10	1.90 ± 0.01	0.836 ± 0.000
	GBF	**2.93 ± 0.14**	**1.92 ± 0.00**	**0.838 ± 0.000**

While Figure 7 and Table 4 demonstrate the use of a combined dataset where empirical model-averaged target labels were used, a second combined dataset was developed where the missing target labels were filled in using the TL predictions for f'_c (for dataset D-2) and E (for dataset D-1). The same four ML methods as described earlier were used for prediction of f'_c and E, and the model efficiency metrics are listed in Table 5. It is seen that the model with TL predictions used as target labels results in improvements of 13-30% in forecast accuracy in terms of RMSE and MAE over those based on the empirical data augmentation. The successful use of the outputs of the ANN TL models as inputs into new ML models demonstrates the synergistic capabilities of combining traditional ML and TL modeling for the prediction of important concrete properties.

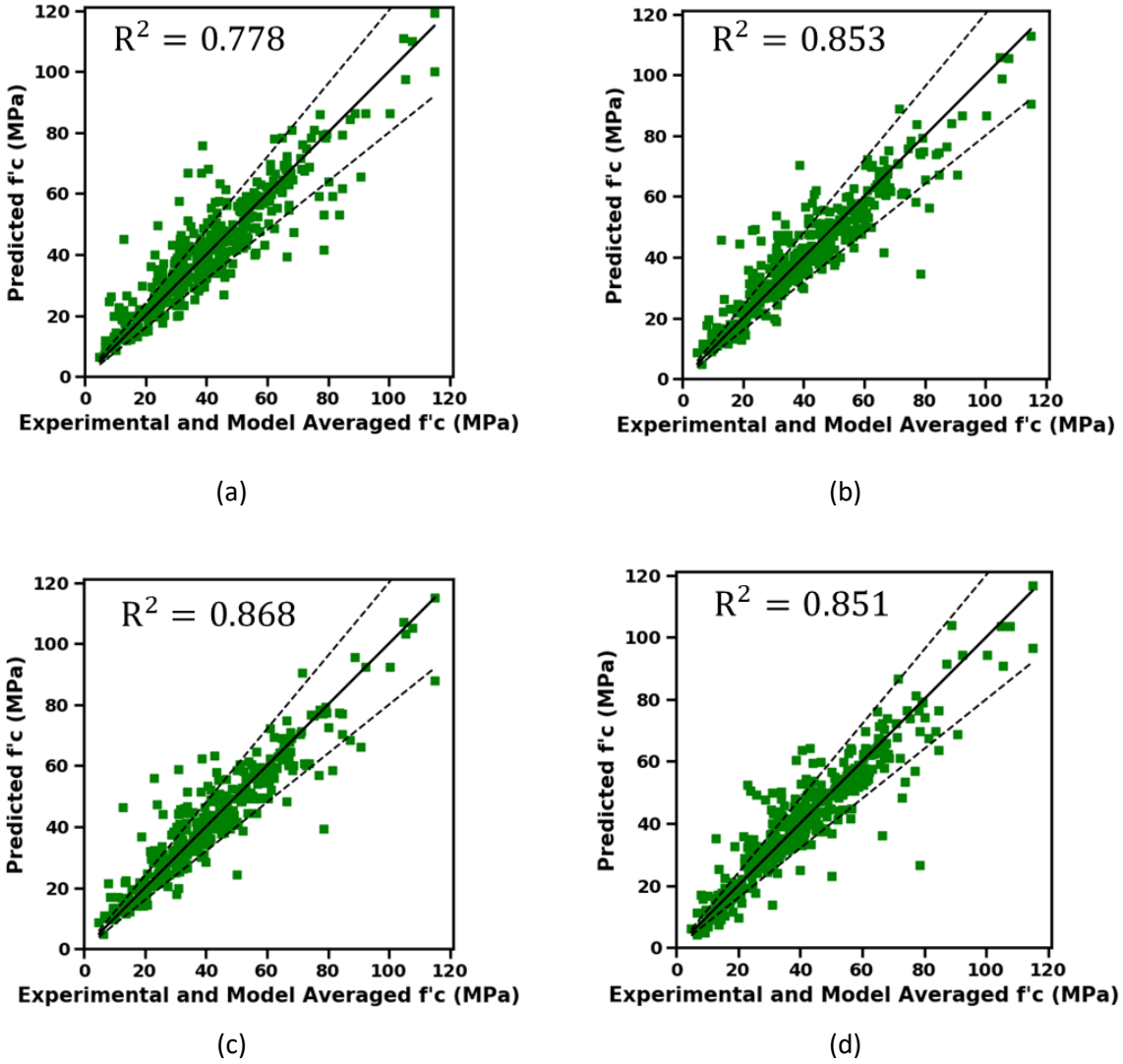


Figure 7. Estimation of f'_c using the combined database where missing target labels are augmented by the model-averaged data, using: (a) ANN, (b) RF, (c) ET, and (d) GBF models. The solid line represents the line of ideality, and the dashed lines represent a ± 20% bound. Testing data that represents 25% of the datasets are shown. R^2 for test data is reported.

Table 5. ML results for the combined datasets with experimental and transfer-learned data labels for f'_c and E prediction, with the average and standard deviation values over three consecutive runs shown. The most accurate ML model for each output is shown in **bold**.

Output	Model Type	RMSE	MAE	$\mathbf{R^2}$
f'c (MPa)	ANN	6.53 ± 2.05	4.51 ± 0.14	0.886 ± 0.011
	RF	6.30 ± 0.55	4.10 ± 0.01	0.895 ± 0.001
	ET	**5.83 ± 0.70**	**3.56 ± 0.02**	**0.909 ± 0.001**
	GBF	6.42 ± 0.58	4.38 ± 0.01	0.890 ± 0.001
E (GPa)	ANN	2.91 ± 0.81	1.83 ± 0.14	0.819 ± 0.014
	RF	2.83 ± 0.38	1.72 ± 0.02	0.829 ± 0.003
	ET	**2.46 ± 0.35**	**1.39 ± 0.01**	**0.871 ± 0.003**
	GBF	2.51 ± 0.31	1.67 ± 0.00	0.865 ± 0.002

5.5 Some considerations while using data-augmented learning

An important caveat of data-augmentation in ML using empirical equations is that the efficiency of the model trained on a dataset depends on how well the empirical model applies to that dataset. To demonstrate this, Figure 8 shows the predicted f'_c values of dataset D-4 (belonging to conventional concrete), where data augmentation from the available E values is carried out either using relevant empirical and code estimations, or the empirical models for RAC described earlier. Note that the output label comes directly from the empirical models, and traditional ML is used here. The need for an appropriate model to train is evident from this figure. To effectively utilize data augmentation though empirical relationships, it must be emphasized that the ML models can only be as accurate and representative of the true output properties as the training data which shapes their architecture and weights.

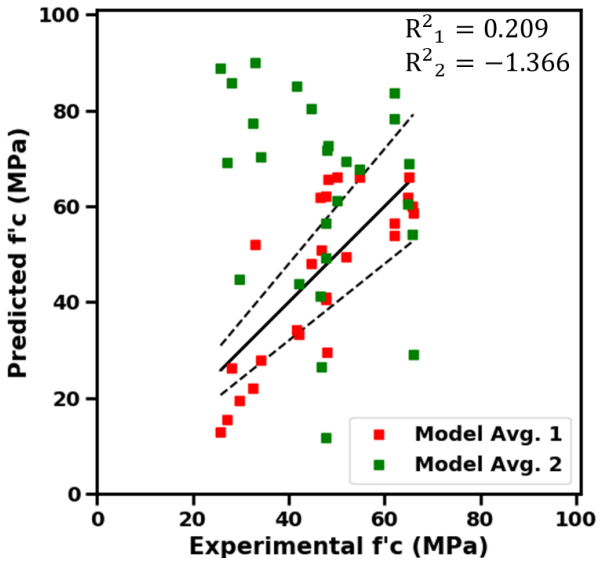
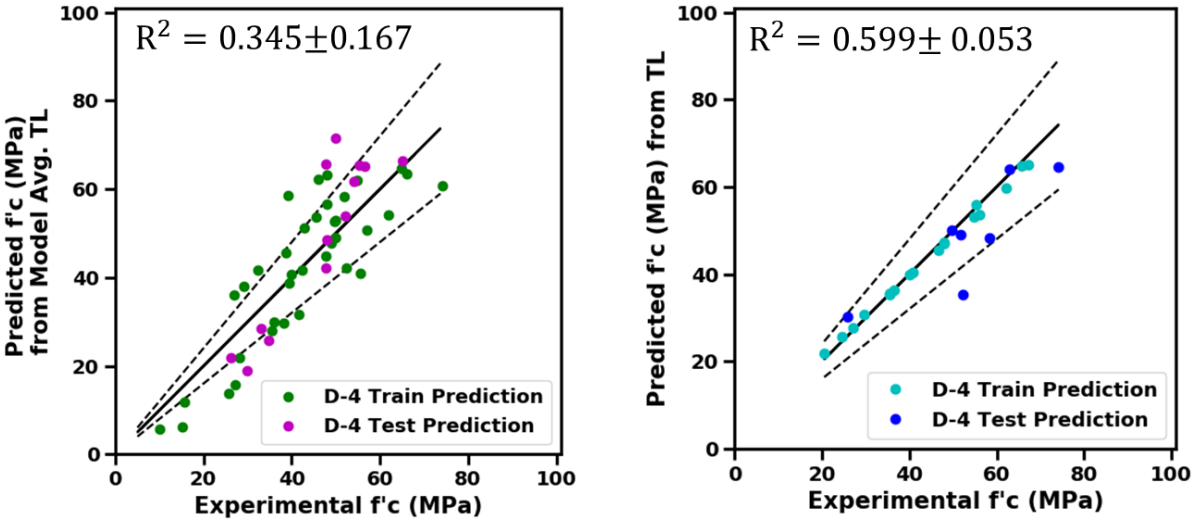


Figure 8. ANN ML predictions of f'$_c$ for dataset D-4 trained on model-averaged estimations: 1. Using international code and empirical models related to the concrete of data D-4, and 2. Using empirical models related to lightweight concrete and concrete containing recycled aggregate. Testing data that represents 25% of the amount of the D-4 dataset is shown. R^2 for test data is reported.

Next, the model for f'$_c$ trained on dataset D-1 is used as the source model for TL to predict the f'$_c$ of dataset D-4. Here, TL is implemented by using 50 randomly drawn input data records from D-4, and the target label used is the model-averaged f'$_c$ shown in Figure 8 as "Model Avg. 1" (details provided in Table A1 of the SI). The process was repeated 25 times and averaged since randomly drawing a few data records from the entire set produces different results each time. Figure 9(a) shows the prediction efficiency in this case, which is higher than when traditional ML was used with the model-averaged output label as shown in Figure 8. The final TL model's accuracy with respect to the experimental values is highly dependent on data size and how many epochs the source model was re-trained. In this case 250 epochs of retraining were performed to try to prevent overfitting, which especially tends to occur in small datasets (Fan et al., 2020).

Since TL is known to work well with smaller real datasets, we use the model for f'$_c$ trained on dataset D-1 as the source model to predict f'$_c$ of D-4, by using a small subset of 25 randomly drawn data records from D-4. These 25 points are drawn from the actual f'$_c$ output of D-4, as opposed to using the model-averaged outputs shown in Figure 9(a). This process was also repeated 25 times and the average reported. Figure 9(b) shows the predictions of transfer-learned model from D-1 on this small subset of D-4. These predictions were superior to that from the model-averaged data, which is along expected lines. If it is possible to obtain even a small amount of actual output data corresponding to the inputs through minimal experiments (e.g., 6 mixture compositions and 4 ages provide ~25 data records), one can transfer-learn

better from a source model, saving time and resources. This is extremely advantageous when the need is to predict the outputs for large datasets. Thus, using the source model for f'$_c$ developed using dataset D-1, and a limited set of new experimental data, it is possible to adequately predict the outcome for a large set of concrete mixtures that have the same feature space as D-1, but proportioned using different cement and replacement materials, w/c, aggregates etc. The same applies to RAC or other types of special concretes like ultra-high performance concrete, where limited actual experimental data can feed into a reliable source model for a similar system to enable fast predictions of properties based on mixture compositions. This ultimately aids in improved mixture proportioning for desirable properties using minimal trial-and-error experimentation, which is common in concrete-related studies.



6.0 Figure 9. ANN estimation of f'$_c$ of dataset D-4 using the model developed from dataset D-1 as the source. TL was performed using (a) 50 randomly drawn points from D-4 Model Avg. 1, and (b) 25 randomly drawn points from the actual D-4 dataset. The solid line represents the line of ideality, and the dashed lines represent a ± 20% bound. R^2 values correspond to the test data. **Summary and Conclusions**

This study demonstrates for the first time, the use of transfer learning techniques to predict the compressive strength or modulus of elasticity from concrete mixture proportions when such data is not directly available, but complementary information is available. This paper lays out the methodology of inductive parameter-transfer learning approach using data augmentation through empirical models to accomplish this objective. This leveraging of empirical relationships between the known and desired

output parameters allowed the TL algorithm to predict an output that the dataset originally did not possess.

The major contributions of the work are development of models to aid: (i) the prediction of E of a conventional concrete dataset from mixture proportions and f'_c based on a model for E developed from a dataset of concrete with a different domain and features (domain adaptation), and (ii) the prediction of f'_c of a dataset that had parameter ranges outside that of the trained model (domain expansion). It has been demonstrated that for datasets with similar input-output relationships, TL increases generalizability. The domain adaptation TL was shown to be more efficient when the source model for TL was based on a dataset with greater diversity and coverage than the target model. Overall, performing TL from an expanded domain to a more restricted domain, along with more data available for training, has been shown to lead to improved modeling accuracy. It was also shown that TL can be used with an established traditional ML model and trained with a smaller dataset corresponding to the target dataset to arrive at improved predictions.

This paper also evaluated combined datasets with data augmentation that replaced missing target labels to develop traditional ML models with enhanced predictive capability. The missing labels were derived either from empirical model-averaging, or from the predictions of domain adaptation TL models (i.e., coupling traditional ML and TL, for the latter case). The traditional ML model coupled with TL predictions as target labels were found to improve the RMSE and MAE accuracy by 13-30% over the model where target labels were augmented by model-averaged data. The successful use of the outputs of the ANN TL models as inputs into new ML models demonstrate the synergistic capabilities of combining traditional ML with TL modeling for the prediction of important concrete properties.

7.0 Acknowledgements

The first author acknowledges a Dean's Fellowship from Arizona State University.

8.0 References

ACI 318-14 Building Code Requirements for Structural Concrete. (2014). Farmington Hills: American Concrete Institute.

Altan, A., Karasu, S., & Zio, E. (2021). A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. *Applied Soft Computing Journal*, 106996.

Arnold, A., Nallapati, R., & Cohen, W. W. (2007). A Comparative Study of Methods for Transductive Transfer Learning. *IEEE International Conference on Data Mining (ICDM) Workshop on Mining and Management of Biological Data.* Omaha, NE.

Bahadori, M. T., Liu, Y., & Zhang, D. (2014). A general framework for scalable transductive transfer learning. *Knowlege Information Systems, 38*, 61-83.

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research, 13*, 281-305.

Bock, F. E., Aydin, R. C., Cyron, C. J., Huber, N., Kalidindi, S. R., & Klusemann, B. (2019). A Review of the Application of Machine Learning and Data Mining Approaches in Continuum Materials Mechanics. *Frontiers in Materials, 6*.

Carpenter, W. C., & Hoffman, M. E. (1997). Selecting the architecture of a class of back-propagation neural networks used as approximators. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 11*, 33-44.

Chen, Y., Tong, Z., Zheng, Y., Samuelson, H., & Norford, L. (2020). Transfer learning with deep neural networks for model predictive control of HVAC and natural ventilation in smart buildings. *Journal of Cleaner Production, 254*, 119866.

Childs, C. M., & Washburn, N. R. (2019). Embedding domain knowledge for machine learning of complex material systems. *MRS Communications, 9*, 806-820.

Chollet, F. (2015). *Keras*. (GitHub repository) Retrieved from https://keras.io/api/

Chopra, P., Sharma, R. K., & Kumar, M. (2016). Prediction of Compressive Strength of Concrete Using Artificial Neural Network and Genetic Programming. *Advances in Materials Science and Engineering*, 7648467.

Chou, J.-S., Chiu, C.-K., Farfoura, M., & Al-Taharwa, I. (2011). Optimizing the Prediction Accuracy of Concrete Compressive Strength Based on a Comparison of Data-Mining Techniques. *Journal of Computing in Civil Engineering, 25*(3), 242-253.

CSA A23.3-04 Design of Concrete Structures. (2004). Mississauga: Canadian Standards Association.

Dao, D. V., Adeli, H., Ly, H.-B., Le, L. M., Le, V. M., Le, T.-T., & Pham, B. T. (2020). A Sensitivity and Robustness Analysis of GPR and ANN for High-Performance Concrete Compressive Strength Prediction Using a Monte Carlo Simulation. *Sustainability, 12*(830), 1-22.

DeRousseau, M. A., Laftchiev, E., Kasprzyk, J. R., Rajagopalan, B., & Srubar III, W. V. (2019). A comparison of machine learning methods for predicting the compressive strength of field-placed concrete. *Construction and Building Materials, 228*, 116661.

ENV 1992-1-1. (2004). Eurocode 2: Design of concrete structures - Part 1-1: General rules and rules for buildings. *European Committee for Standardization*.

Fan, C., Sun, Y., Xiao, F., Ma, J., Lee, D., Wang, J., & Tseng, Y. C. (2020). Statistical investigations of transfer learning-based methodology for short-term building energy predictions. *Applied Energy, 262*, 114499.

Farhan, J. (2015). Overview of missing physical commodity trade data and its imputation using data augmentation. *Transportation Research Part C, 54*, 1-14.

Ford, E. L., Kailas, S., Maneparambil, K., & Neithalath, N. (2020). Machine learning approaches to predict the micromechanical properties of cementitious hydration phases from microstructural chemical maps. *[Under Review] Construction and Building Materials*.

Gardner, P., Liu, X., & Worden, K. (2020). On the application of domain adaptation in structural health monitoring. *Mechanical Systems and Signal Processing, 138*, 106550.

Han, T., Siddique, A., Khayat, K., Huang, J., & Kumar, A. (2020). An ensemble machine learning approach for prediction and optimization of modulus of elasticity of recycled aggregate concrete. *Construction and Building Materials, 244*, 118271.

Haranki, B. (2009). Strength, Modulus of Elasticity, Creep, and Shrinkage of Concrete used in Florida. *Dissertation from the University of Florida*.

IS 456 Plain and Reinforced Concrete Code of Practice. (2007). New Delhi: Bureau of Indian Standards.

Jha, D., Choudhary, K., Tavazza, F., Liao, W.-k., Choudhary, A., Campbell, C., & Agrawal, A. (2019). Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. *Nature Communications, 10*, 5316.

Karasu, S., & Altan, A. (2019). Recognition Model for Solar Radiation Time Series based on Random Forest with Feature Selection Approach. *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)* (pp. 8-11). IEEE.

Konstantopoulos, G., Koumoulos, E. P., & Charitidis, C. A. (2020). Testing Novel Portland Cement Formulations with Carbon Nanotubes and Intrinsic Properties Revelation: Nanoindentation Analysis with Machine Learning on Microstructure Identification. *nanomaterials, 10*, 645.

Li, X., Hu, Y., Li, M., & Zheng, J. (2020). Fault diagnostics between different type of components: A transfer learning approach. *Applied Soft Computing Journal, 86*, 105950.

Li, X., Liu, Z., Cui, S., Luo, C., Li, C., & Zhuang, Z. (2019). Predicting the effective mechanical property of heterogeneous materials by image based modeling and deep learning. *Computer Methods Applied Mechanics and Engineering Methods, 347*, 735-753.

Liu, Y., Yang, C., Liu, K., Chen, B., & Yao, Y. (2019). Domain adaptation transfer learning soft sensor for product quality prediction. *Chemometrics and Intelligent Laboratory Systems, 192*, 103813.

Nguyen, K. T., Nguyen, Q. D., Le, T. A., Shin, J., & Lee, K. (2020). Analyzing the compressive strength of green fly ash based geopolymer concrete using experiment and machine learning approaches. *Construction and Building Materials, 247*, 118581.

Oey, T., Jones, S., & Sant, G. (2020). Machine learning can predict setting behavior and strength evolution of hydrating cement systems. *Journal of the American Ceramic Society, 103*(1), 480-490.

Ohno, H. (2019). Neural network-based transductive regression model. *Applied Soft Computing Journal, 84*, 105682.

Ohno, H. (2020). Training data augmentation: An empirical study using generative adversarial net-based approach with normalizing flow models for materials informatics. *Applied Soft Computing Journal, 86*, 105932.

Oviedo, F., Ren, Z., Sun, S., Settens, C., Liu, Z., Putri Hartono, N. T., . . . Buonassisi, T. (2019). Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks. *npj Computational Materials, 5*, 60.

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345-1359.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Sadati, S. (2017). High-volume recycled materials for sustainable transportation infrastructure. *Missouri University of Science and Technology Scholars' Mine Doctoral Dissertations*, 2583.

Sadati, S., da Silva, L. E., Wunsch II, D. C., & Khayat, K. H. (2019). Artificial Intelligence to Investigate Modulus of Elasticity of Recycled Aggregate Concrete. *ACI Materials Journal, 116*(1), 51-62.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks, 61*, 85-117.

Son, H., Kim, C., & Kim, C. (2012). Automated Color Model–Based Concrete Detection in Construction-Site Images by Using Machine Learning Algorithms. *Journal of Computing in Civil Engineering, 26*(3), 421-433.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research, 15*, 1929-1958.

Su, C., Ju, S., Liu, Y., & Yu, Z. (2015). Improving Random Forest and Rotation Forest for highly imbalanced datasets. *Intelligent Data Analysis, 19*, 1409-1432.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5 RmsProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data, 3*(9).

Yamada, H., Liu, C., Wu, S., Koyama, Y., Ju, S., Shiomi, J., . . . Yoshida, R. (2019). Predicting Materials Properties with Little Data Using Shotgun Transfer Learning. *American Chemical Society Central Science, 5*(10), 1717-1730.

Yeh, I.-C. (1998). Modeling of Strength of High-Performance Concrete Using Artifical Neural Networks. *Cement and Concrete Research, 28*(12), 1797-1808.

Young, B. A., Hall, A., Pilon, L., Gupta, P., & Sant, G. (2019). Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: New insights from statistical analysis and machine learning methods. *Cement and Concrete Research, 115*, 379-388.

Yun, H., Zhang, C., Hou, C., & Liu , Z. (2019). An Adaptive Approach for Ice Detection in Wind Turbine With Inductive Transfer Learning. *IEEE Access, 7*, 122205-122213.

Zhang, J., Huang, Y., Wang, Y., & Ma, G. (2020). Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms. *Construction and Building Materials, 253*, 119208.

Zhang, Y., & Ling, C. (2018). A strategy to apply machine learning to small datasets in materials science. *npj Computational Materials, 4*, 25.