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

Narayanan Neithalath

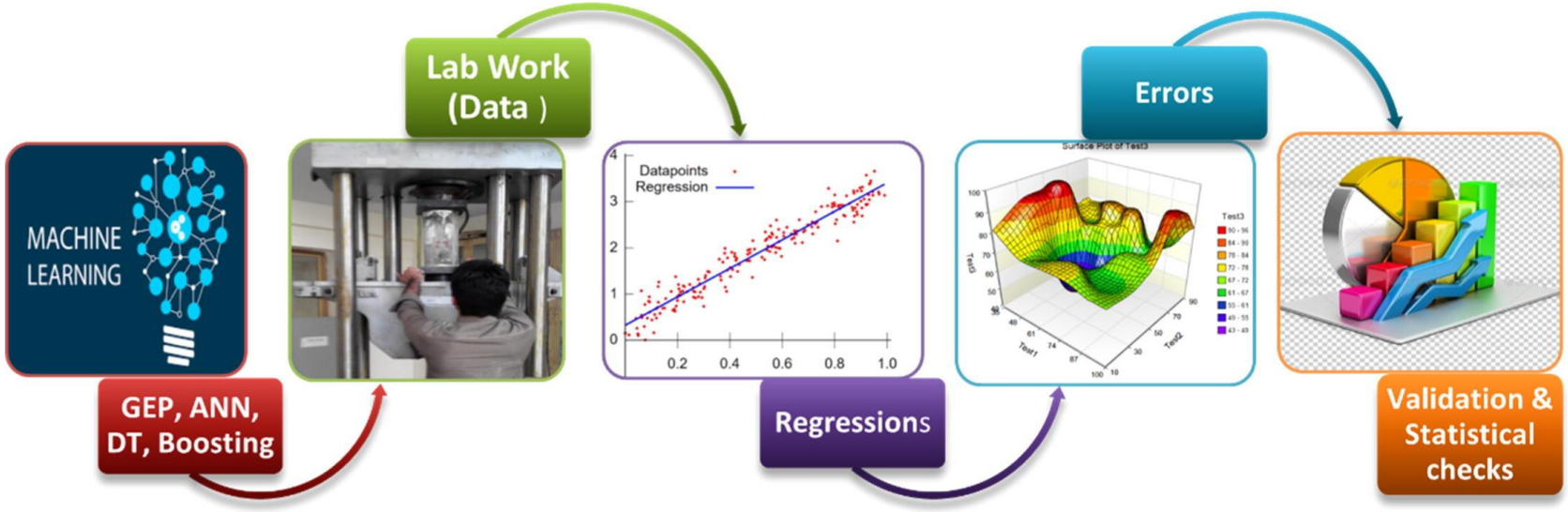
School of Sustainable Engineering and the Built Environment
Arizona State University
Narayanan.Neithalath@asu.edu
http://neithalath.engineering.asu.edu

Emily Ford; Arizona State University
Kailasnath Maneparambil; Intel Corporation
Gaurav Sant; University of California Los Angeles
Aditya Kumar; Missouri University of Science and Technology

Presented at The Spring ACI convention, San Francisco, April 2023

ML in concrete applications

- Utilize the semi-empirical rules that inform the relationship between constitutive materials/phase chemistry and concrete properties
- Perform predictions on previously untrained datasets
- A large body of publications to predict strength and other properties from mixture proportions; optimization of concrete mixtures for strength, durability, cost, CO_2 impacts….
- ML is dependent on large, high-quality datasets



Song et al. 2021

ML in concrete applications

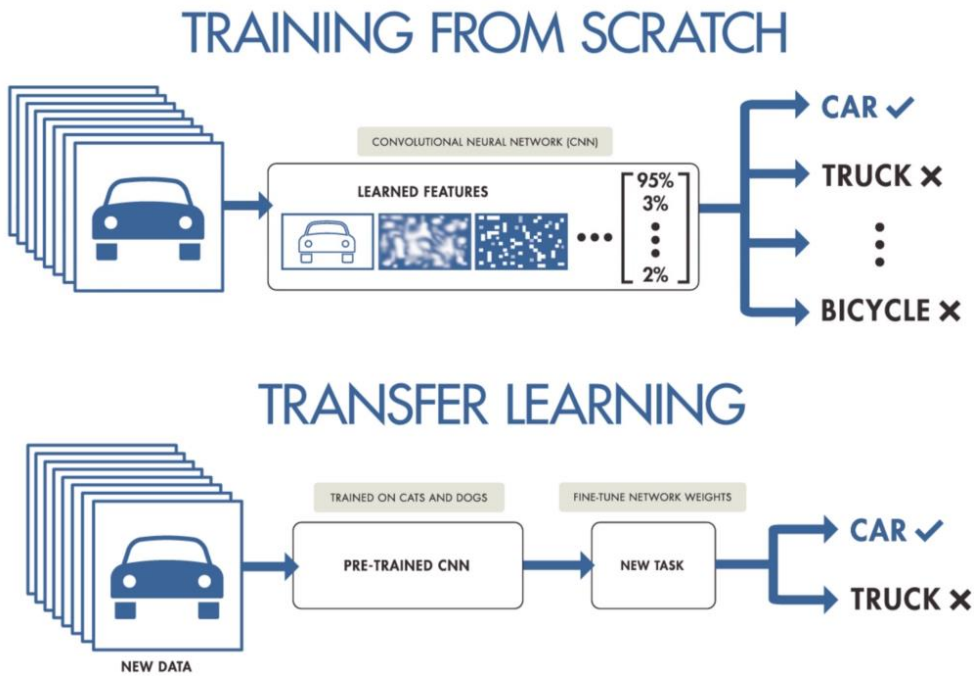
- After training, traditional ML models are only able to predict on untrained datasets with identical input ranges and relationships between variables
- When the proportions and types of ingredients (e.g., cement and supplementary cementing material chemistry, admixtures, aggregate type, etc.) change, previously trained ML models would not be able to adapt to the changes.

ML with missing data

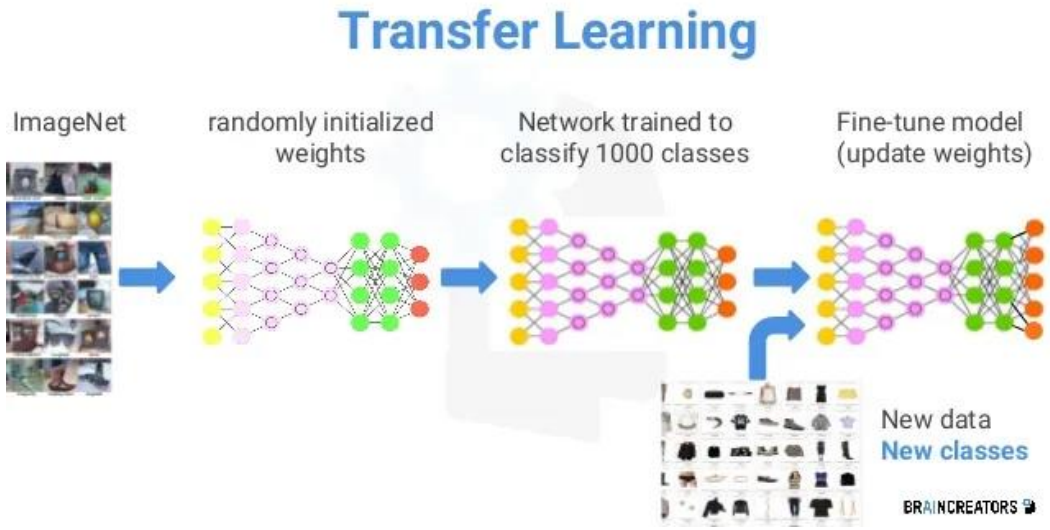
- Missing data is poorly handled in ML models – and *data does go missing*……
- Most common approach for handling missing data - *deletion*
- Data augmentation to fill the missing data gaps
  - Decision trees with surrogate splits, use of pattern sub models, or incorporation of auto encoders)
- Statistics-based approaches – mean substitution, interpolation, regression, stochastic regression, K nearest neighbor
  - Biased estimates
- Adversarial nets to create 1000s of additional data points – convergence to the true model, a risk
- Empirically established relationship between the variables *("crude estimation")*

Transfer learning (TL)

- Lost in the wilderness of neural network's hyperparameters with no idea where to start and what to configure???
- No need to train from scratch - Leverage past powerful deep learning models and transfer their knowledge to the specific domain



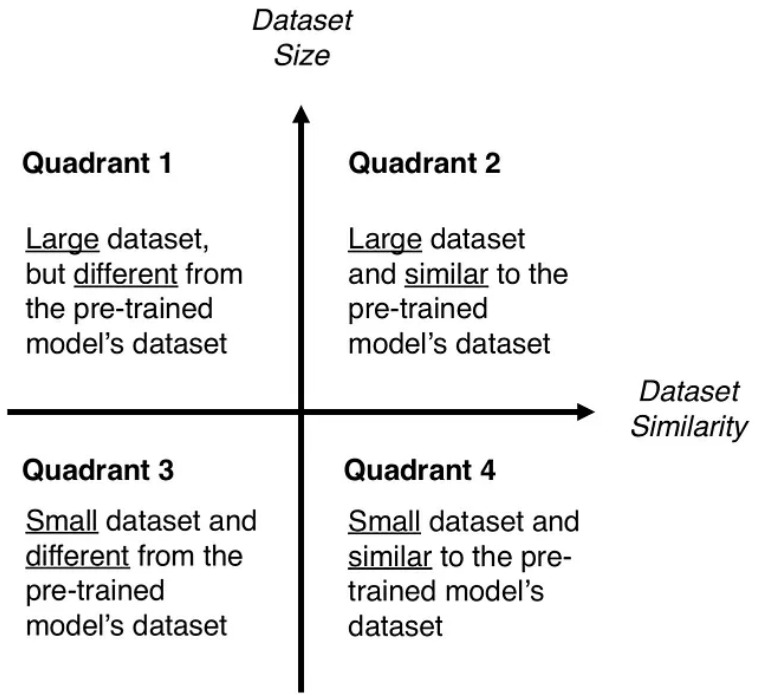
https://medium.datadriveninvestor.com/introducing-transfer-learning-as-your-next-engine-to-drive-future-innovations-5e81a15bb567



https://medium.datadriveninvestor.com/what-you-must-know-about-transfer-learning-4a6e4cb9fbad

Transfer learning (TL)

- 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
- 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
- Domain expansion
  - Domain of the target lies somewhat outside that of the source
- Domain adaptation
  - Target output is different, but can be related to the source output



TL for concrete properties prediction

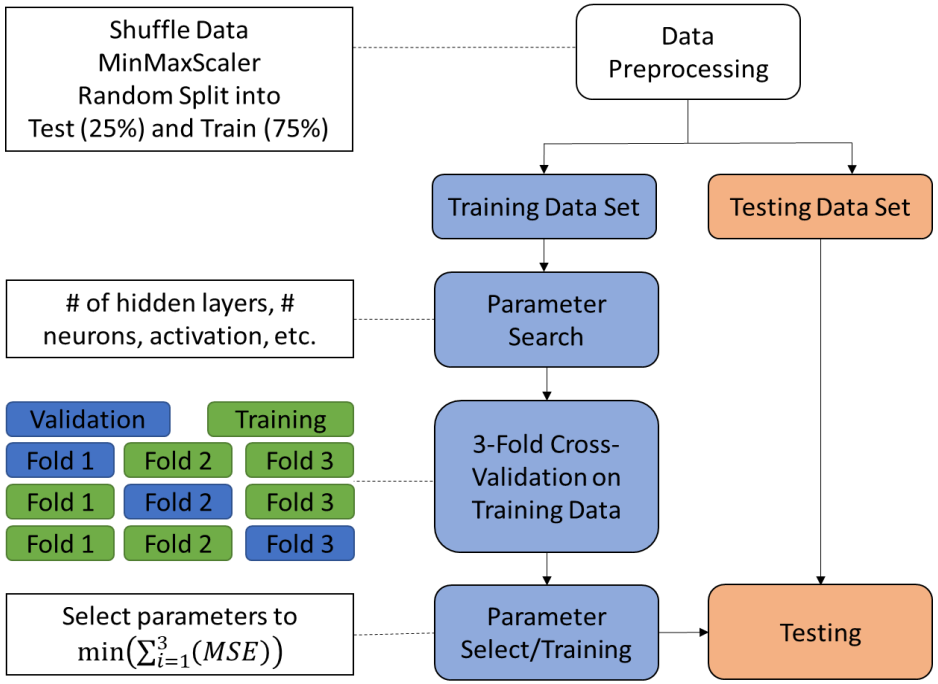
- For concrete, more often than not, only a limited range of tests are carried out (e.g., compressive strength), and other parameters (e.g., elastic modulus, durability) are indirectly inferred from the strength results
- Expanding the applicable range of mixture proportions through ML; enhancing the predictability of concrete properties with changes in constituents through ML models reduces expensive and time-consuming testing.
- Combines ML with data augmentation and transfer learning
- Allows prediction of 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.

Data sets

Dataset	No. of Data Records	Output	Variables	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	8	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	13	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	6	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	6	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	

- Even though D1 and D3 contain similar inputs and outputs, the range of some of the parameter values of D3 lie outside the range of D1

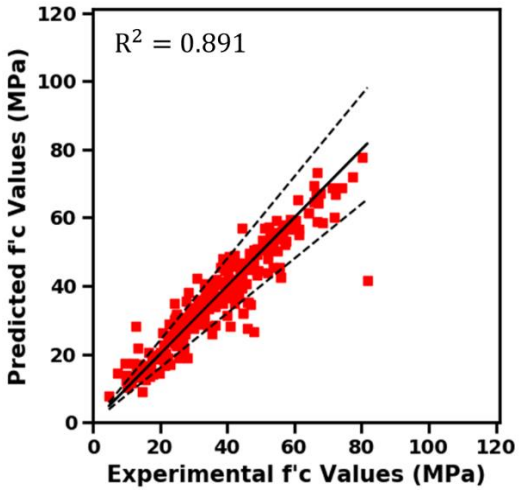
ML details



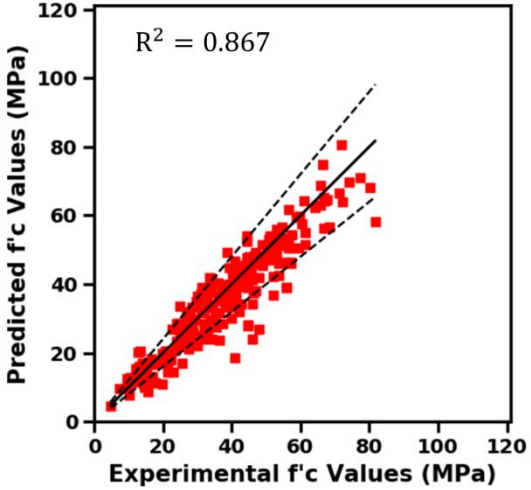
- Keras neural network framework written in Python to build and train the ANNs
- ANN and Ensemble methods (Random forest RF, Extra trees ET, Gradient boosted tree GBT)
- ReLu activation function with a learning rate of 0.001 and an RMSprop optimization scheme with backpropagation
- Training of ANN neuron weights using backpropagation
- Coarse optimization of the hyperparameters - random search pattern

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]

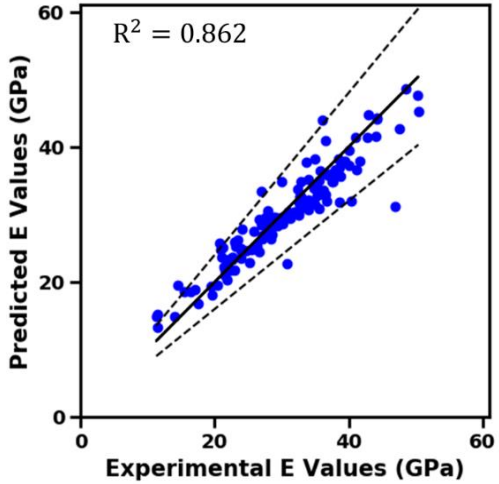
ML (*not TL*) predictions



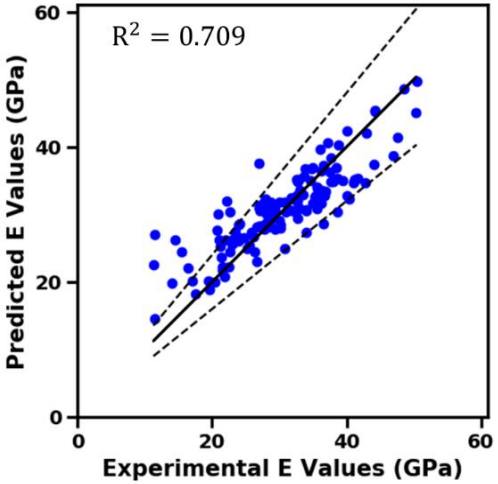
D-1 with original 8 inputs



D-1 with truncated 7 inputs

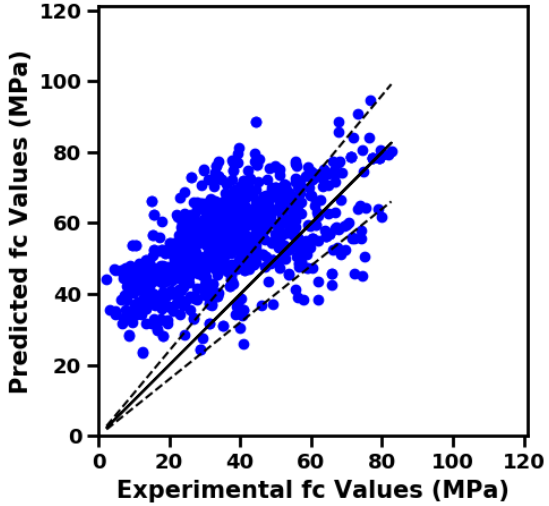


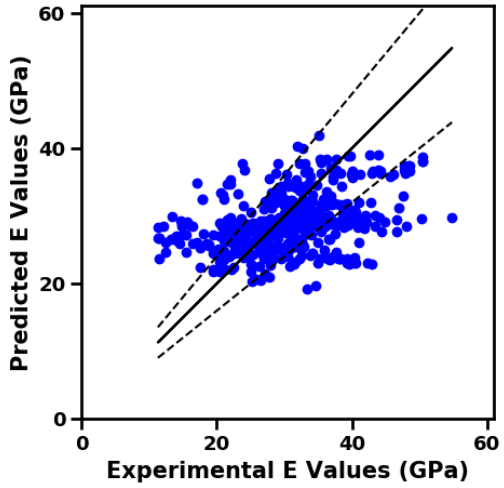
D-2 with original 13 inputs



D-2 with truncated 7 inputs

TL via retraining the best-fit ML model

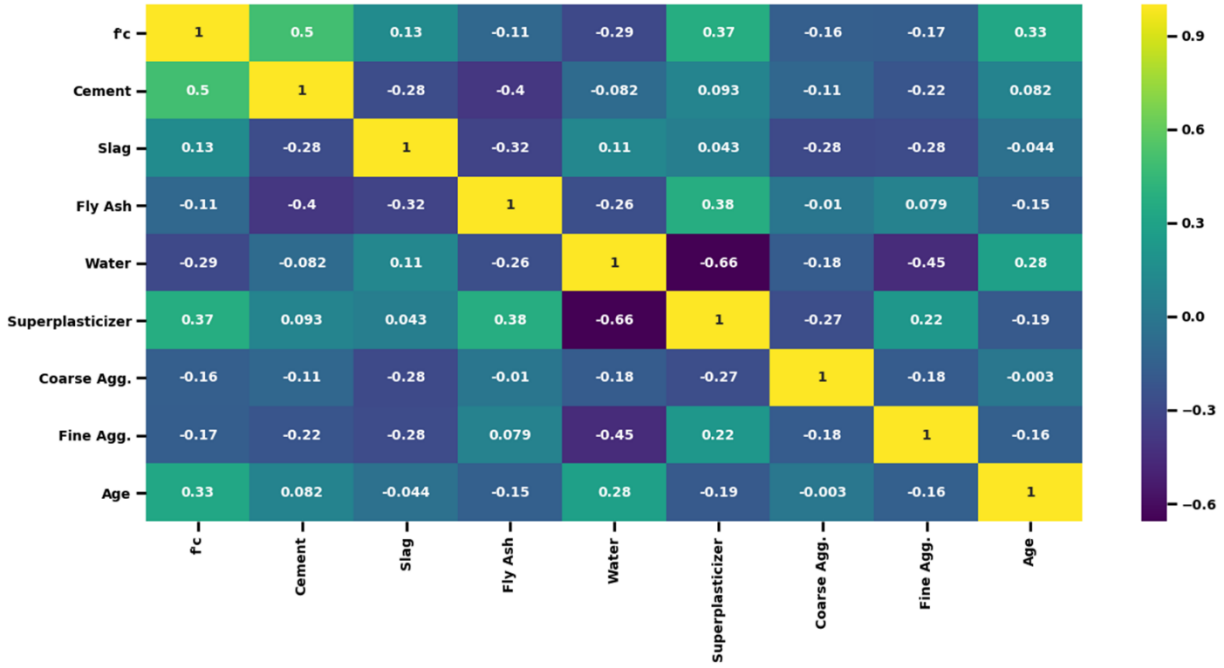




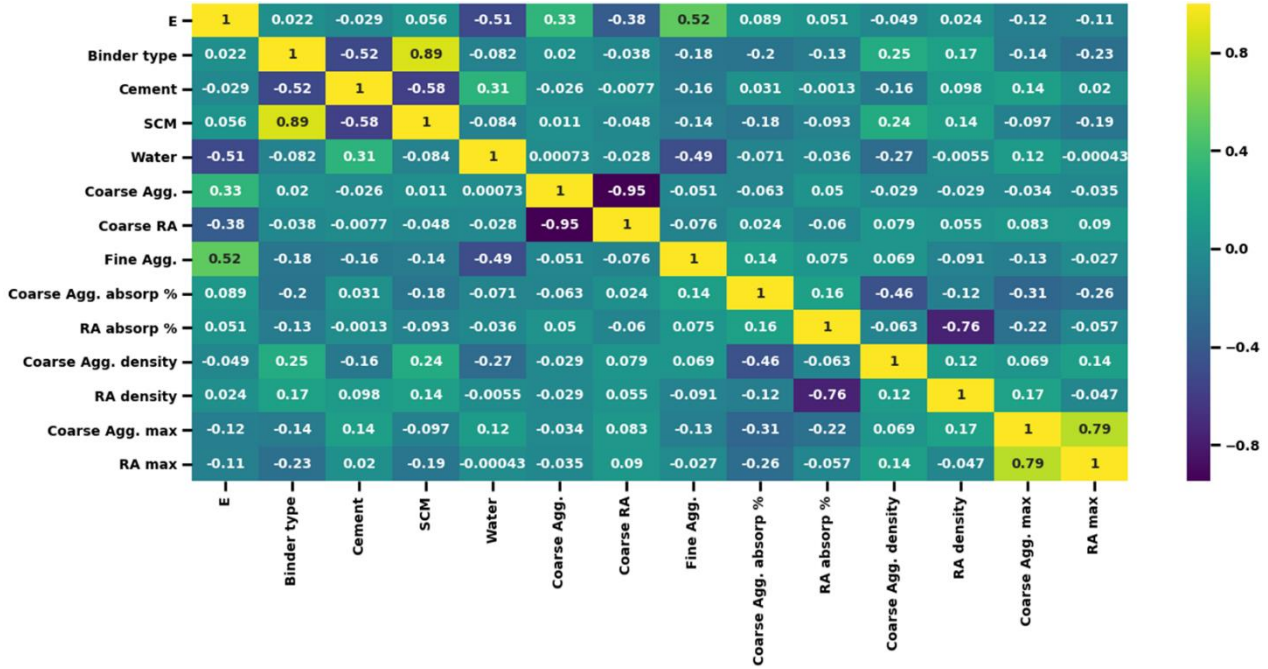
Presented at The Spring ACI convention, San Francisco, April 2023

Pearson coefficient for dataset organization

D1



D2

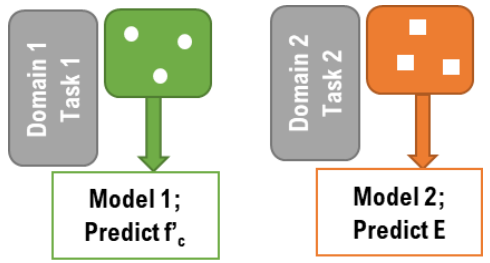


- Pearson correlation coefficients determined for datasets D-1 and D-2 (no input truncation was needed for D-3 and D-4) with the original 8 and 13 inputs, and the truncated 7 inputs.
- Absolute values of the correlation coefficients with respect to E in dataset D-2 for most of the ignored terms were less than 0.10.

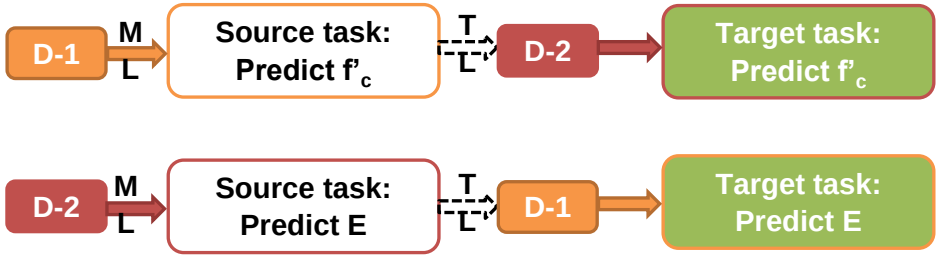
Presented at The Spring ACI convention, San Francisco, April 2023

TL details

- In Transfer learning prior knowledge from one domain and task can be applied to another domain and task
- $\mathcal{D} = \{\mathcal{X}, p(\mathcal{X})\}$, where $\mathcal{X}$ represents the feature space
  - feature space includes the input parameters, such as the contents of different ingredients, age
- 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
- 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



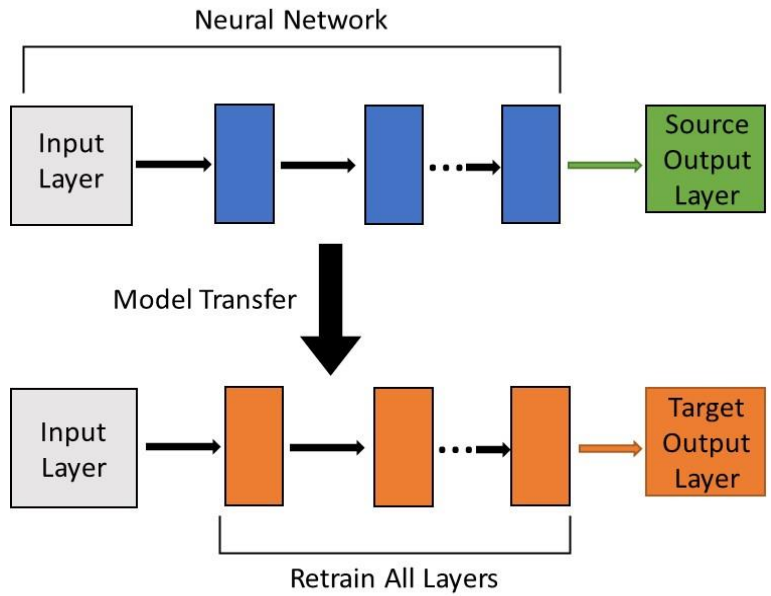
Traditional ML – separate isolated models



TL – knowledge gained in one task utilized for the other

TL implementation

- Converting a *transductive TL* ($\mathcal{D}_S \neq \mathcal{D}_T$; and labeled data in source and target domains not the same) into *an inductive TL* (labeled data available in both source and target domains)
- *Inductive TL* 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
- Empirically derived strength-E relationships to provide missing target data for training
- Parameter transfer architecture for ANN/CNN
  - Model weights act as initial values to retrain the target model

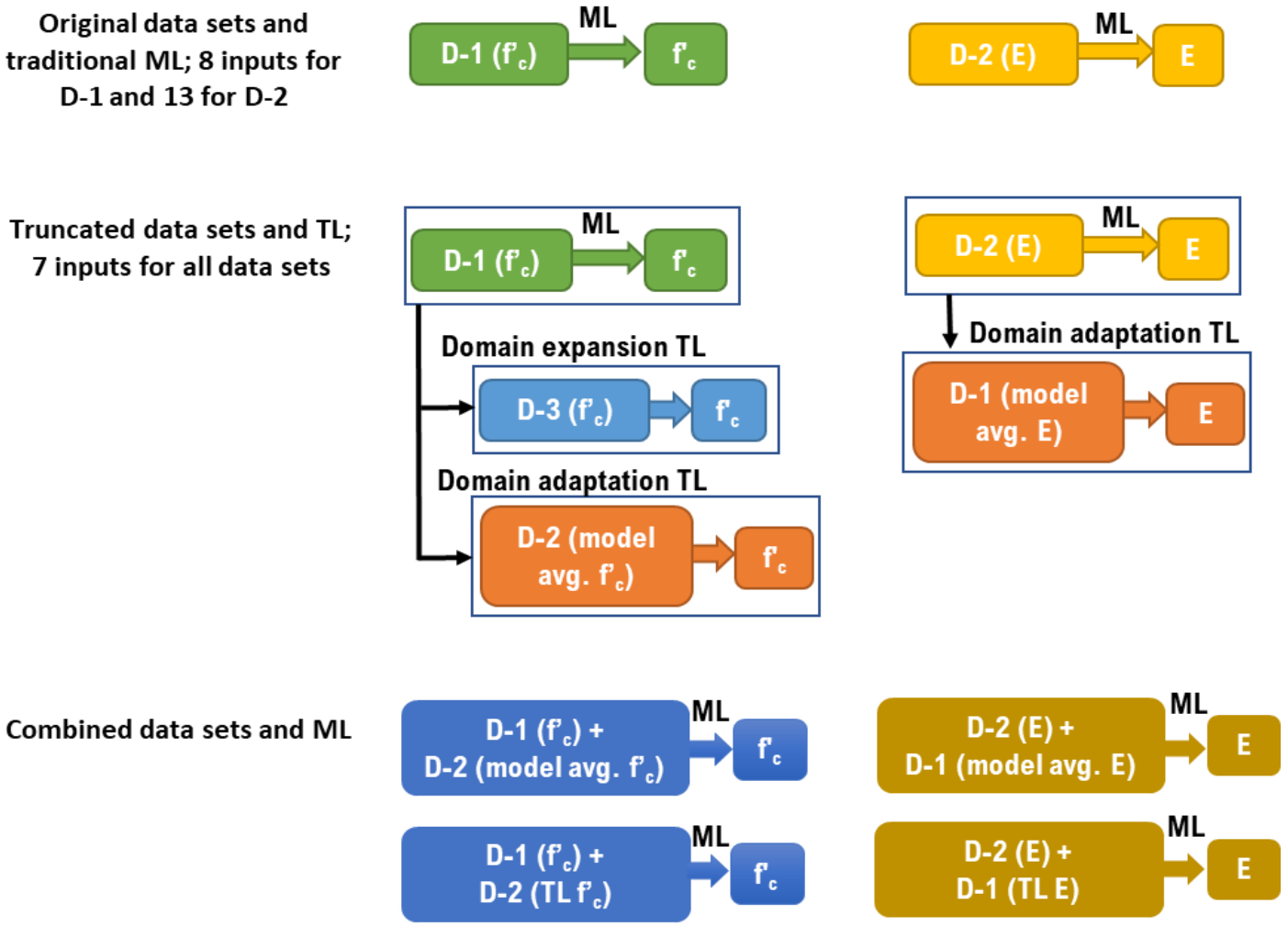


Data augmentation

Data Set	Formula	Source
D-1	$E_c = 4.7\sqrt{f'c}$	ACI 318 19.2.2.1.b (ACI 318-14 Building Code Requirements for Structural Concrete, 2014)
D-1	$E_c = 4.5\sqrt{f'c}$	CSA A23 8.6.2.3 (CSA A23.3-04 Design of Concrete Structures, 2004)
D-1	$E_c = 5.0\sqrt{f'c}$	IS 456 6.2.3.1 (IS 456 Plain and Reinforced Concrete Code of Practice, 2007)
D-2 and D-4 Model Avg. 1	$f'c = \left(\frac{E_c}{4.7}\right)^2$	ACI 318 19.2.2.1.b (ACI 318-14 Building Code Requirements for Structural Concrete, 2014)
D-2 and D-4 Model Avg. 2	$f'c = \left(\frac{E_c}{4.63}\right)^2$	Ravindrarajah and Tam (Sadati, da Silva, Wunsch II, & Khayat, 2019)
D-2 and D-4 Model Avg. 2	$f'c = \frac{E_c - 8.242}{0.378}$	Mellmann (Sadati, da Silva, Wunsch II, & Khayat, 2019)
D-4 Model Avg. 2	$f'c = \left(\frac{E_c * 1000}{w_c^{1.5} * 0.043}\right)^2$	ACI 318 19.2.2.1.a (ACI 318-14 Building Code Requirements for Structural Concrete, 2014) with $w_c = 1922.22\frac{kg}{m^3}$ $(120\frac{lb}{ft^3})$
D-4 Model Avg. 1	$f'c = 10 * \left(\frac{E_c}{22}\right)^3$	Eurocode 2 Table 3.1 (1992-1-1, 2004)
D-4 Model Avg. 1	$f'c = \left(\frac{E_c * 1000}{63351 * \frac{4700}{57000}}\right)^2$	Georgia granite aggregate (Haranki, 2009)
D-4 Model Avg. 1	$f'c = \left(\frac{E_c * 1000}{55824 * \frac{4700}{57000}}\right)^2$	Miami Oolite limestone (Haranki, 2009)

- Code estimations and empirical equations to provide labeled target data for inductive transfer
- Data augmentation helps to change the target from unlabeled to labeled – enabling better predictions
- Serve as a guiding basis for the TL models to learn the relationship between the inputs and both desired output mechanical properties
- Changing the weights of the initial models for E or f'_c through backpropagation
- Source model acts as initialized weights based on the source data rather than the usual randomized weights when first generating an ANN model
- Objective function gradients and backpropagation are now performed with respect to the target dataset

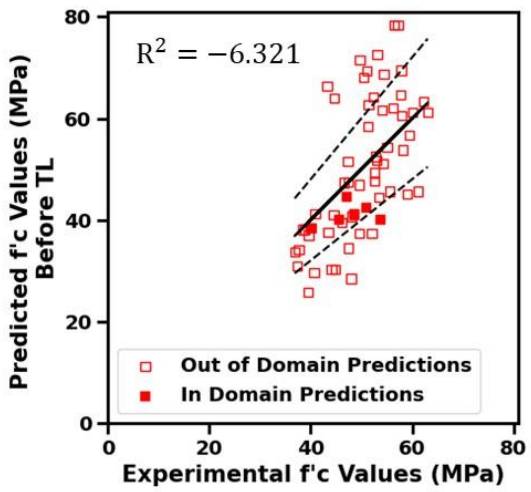
Domain expansion and domain augmentation TL



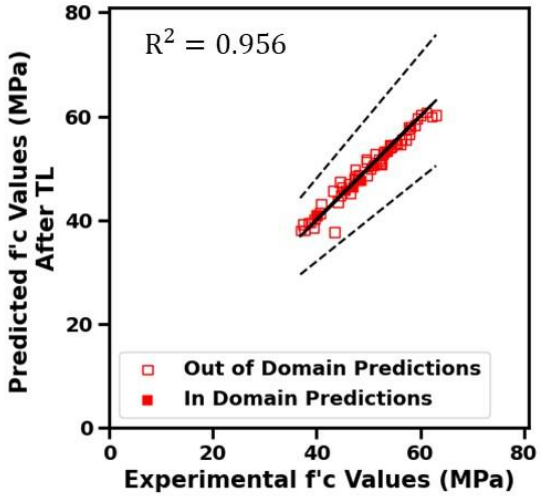
Domain expansion TL

- Domain expansion case, from dataset D-1 to D-3
- A more generalized model is created from a restricted domain and retrained to account for the expanded domain
- ANN based model developed using dataset D-1 re-trained to predict f'_c of the dataset D-3
  - a more generalized model is created from a restricted domain and retrained to account for the expanded domain

No TL



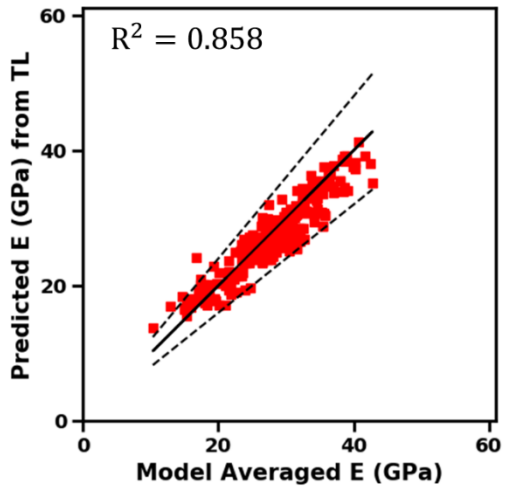
TL



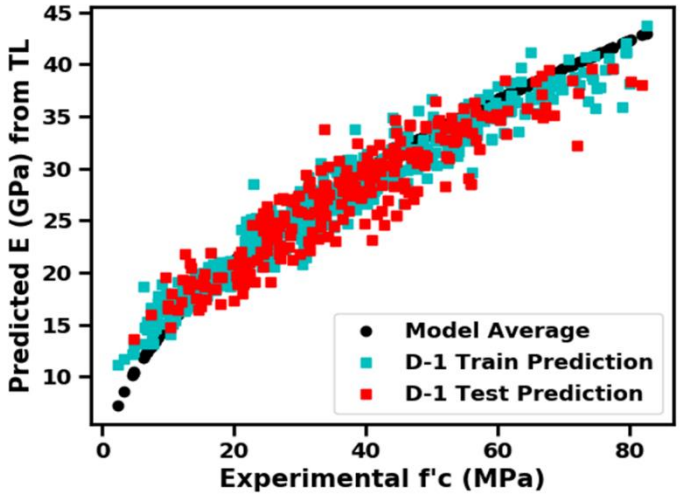
Ford et al. Machine Learning with Appl. 2022

Data augmentation assisted TL

- 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 (use of different cement replacement materials, recycled aggregates)
- Domain adaptation strategy



E prediction of D-1 using model-averaged E to retrain the model developed for D-2



Relationship between experimental f'_c and predicted E for D-1

- Parameter-TL from D-2 to D-1 (to predict E;) generated an ANN model as accurate as the initial E prediction model that incorporated all 13 of the original D-2 dataset inputs
- Source model typically with greater diversity and coverage than the target model
- Performing TL from an expanded to a more restricted domain is better

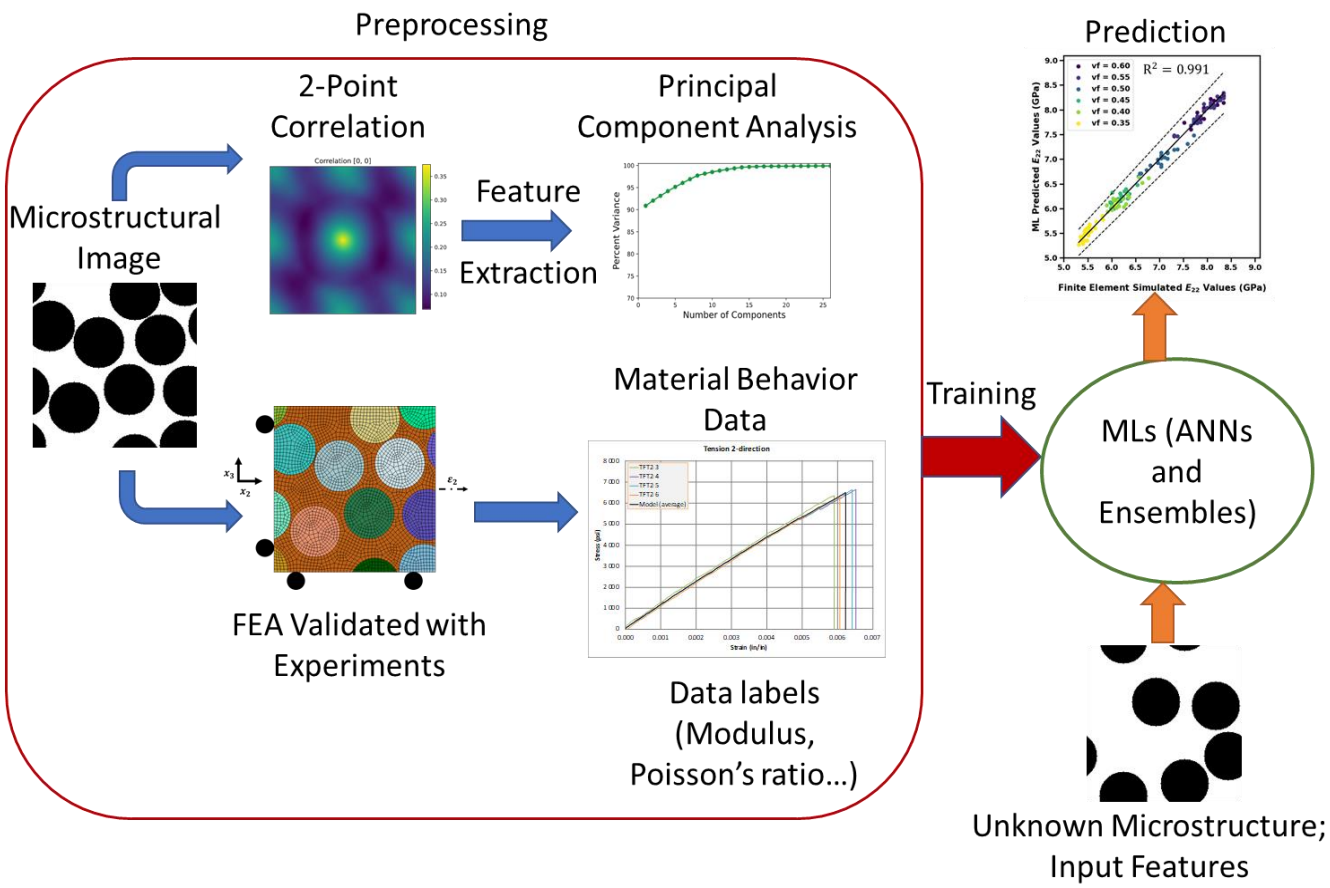
Ford et al. Machine Learning with Appl. 2022

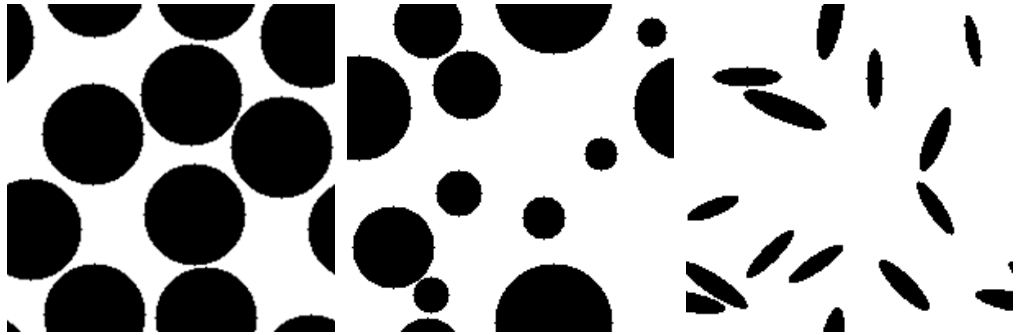
Combining datasets

- No longer assume that the source and target domains are different, by combining data
- Considered that the model-averaged data or transfer-learned data are reasonable estimates of true target labels
- Missing labels were provided using: (a) empirically-derived model-averaged data, and (b) transfer-learned E and f'c from the domain adaptation models
- Larger amount of data to train, and an expanded domain

Output	Model Type	RMSE	MAE	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

Image-based application to composites

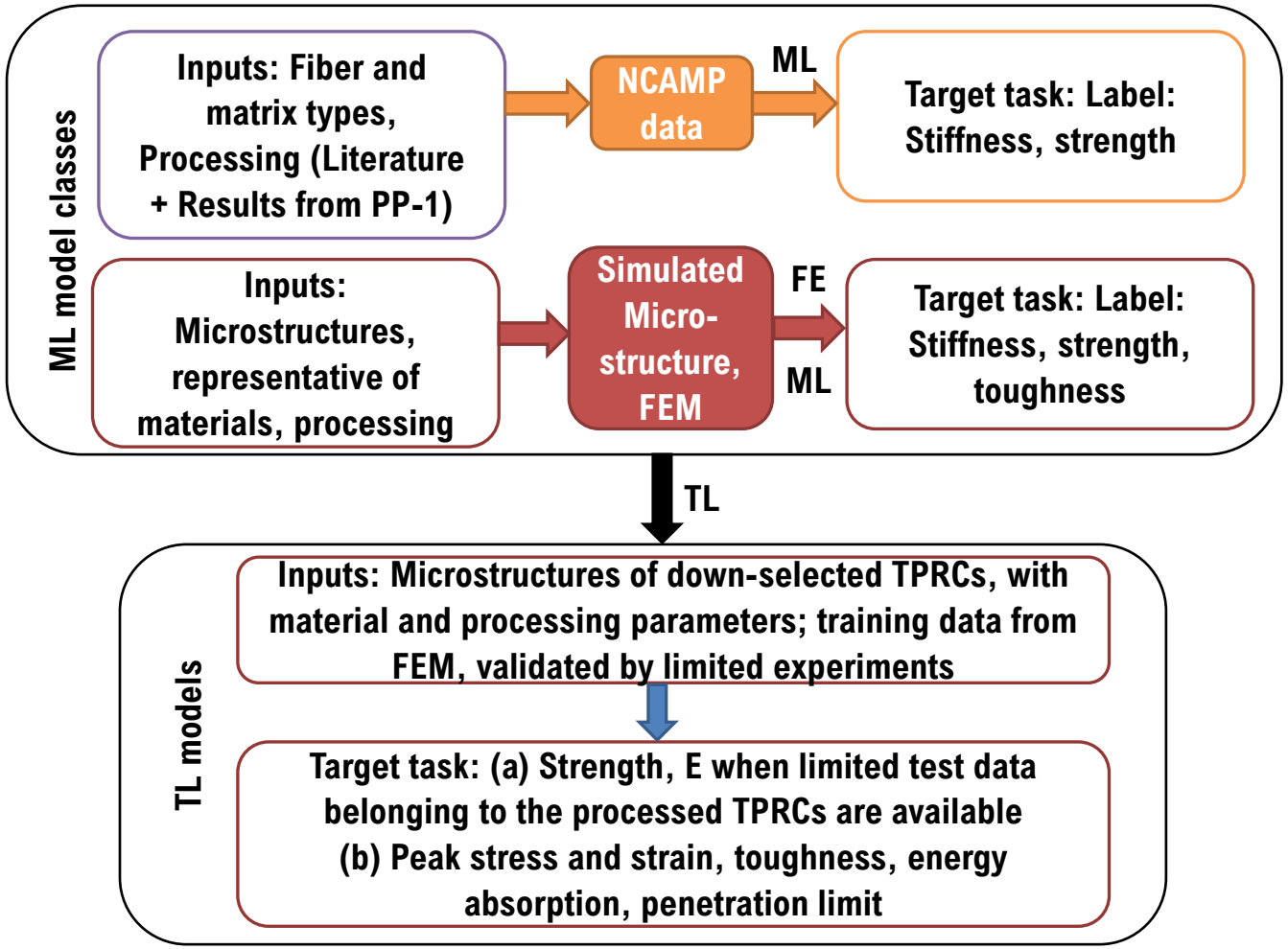


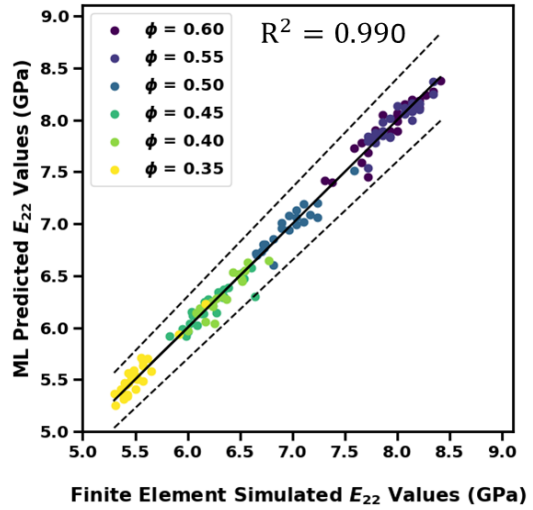


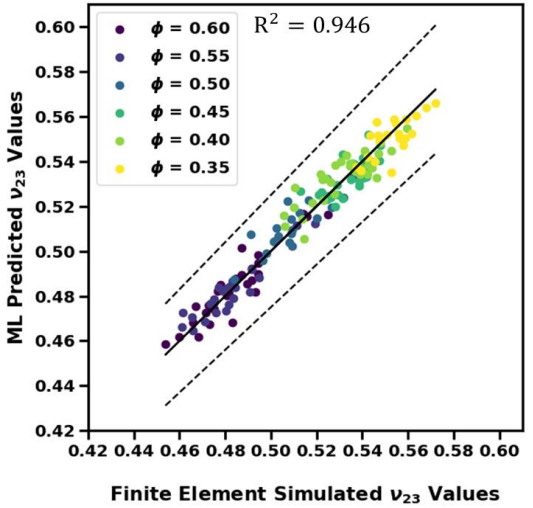
- Microstructural analysis
- Finite element analysis (linear/non-linear) for properties as ground truths
- Training and prediction
- TL to unknown microstructures

Ford et al. Comp. Mat. Sci 2021

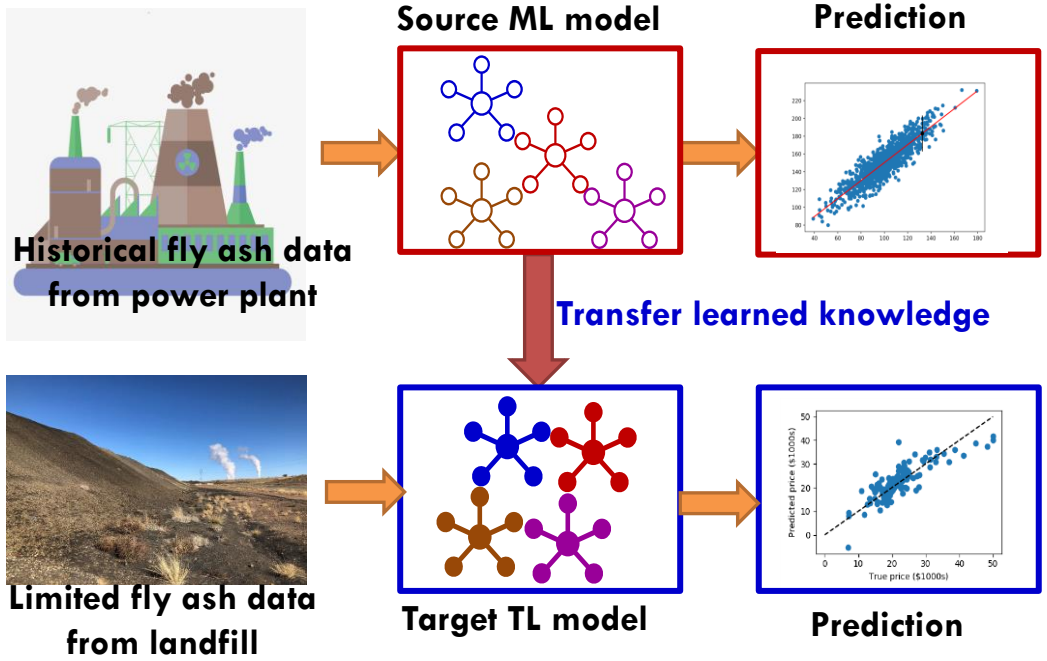
Image-based application to composites

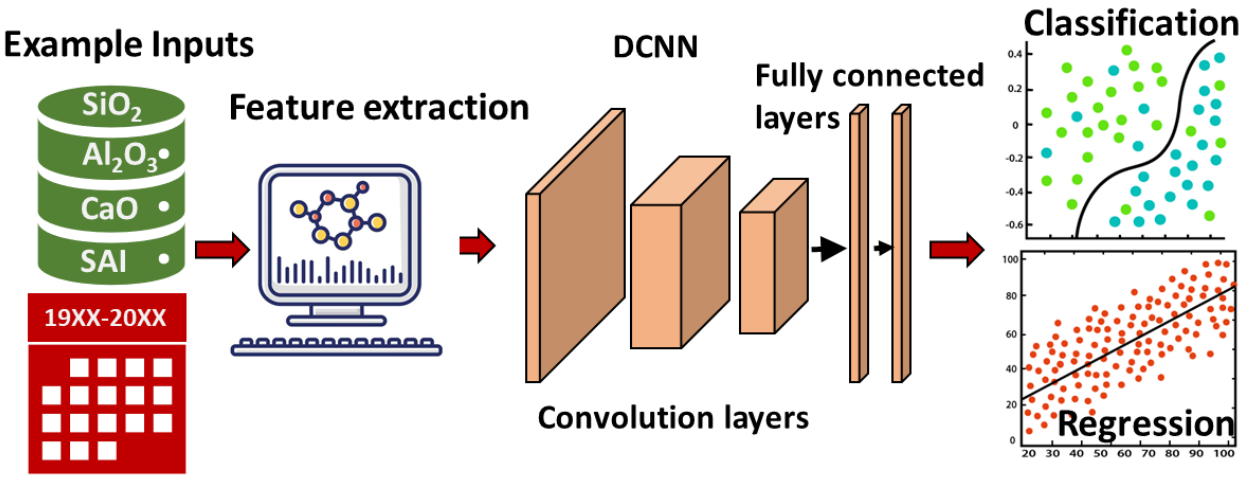






TL to large datasets





- TL from one power plant data to corresponding landfill data (Domain expansion)
- TL from one power plant data to another power plant data (Domain adaptation)
- TL from one landfill data to another landfill data (Domain adaptation)

Conclusions

- Demonstrates the use of TL 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
- Inductive parameter-transfer learning approach using data augmentation through empirical models
- Development of models to aid
  - 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
  - the prediction of f'_c of a dataset that had parameter ranges outside that of the trained model (domain expansion).
- 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.

Presented at The Spring ACI convention, San Francisco, April 2023