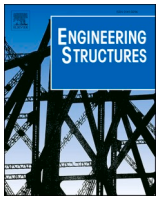
Engineering Structures 266 (2022) 114579

Contents lists available at ScienceDirect

Engineering Structures

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



Evaluation of transfer learning models for predicting the lateral strength of reinforced concrete columns

Hongrak Pak [a,*], Stephanie German Paal [a]

[a] Zachry Department of Civil & Environmental Engineering, Texas A&M University, College Station, TX 77843, United States

ARTICLE INFO

Keywords:
Transfer learning
Machine learning
Reinforced concrete
Lateral strength
Seismic capacity

ABSTRACT

Transfer learning aims to extract knowledge from one or more source tasks and apply the knowledge to a different task for more accurate predictions. The main purposes of this study are to investigate different knowledge transfer techniques, apply them to accurately predict the lateral strength of reinforced concrete columns with only a small amount of training data, and compare the transferability of each method. According to the various source and target domains, three different experiments are designed to directly compare the performance across section type, shear reinforcement area, and concrete compressive strength. In all cases in this study, knowledge transfer techniques show better prediction performance than the models trained without any knowledge transfer techniques. Therefore, we can conclude that transferring pre-trained knowledge from the source domain enables a model to better explain the response variable in the target domain. The performance improvement is particularly emphasized when the available data for the target domain is small. Thus, transfer learning can be one way to address the data scarcity problem in structural engineering. Furthermore, transferring the pre-trained knowledge is more associated with the underlying physical relationship between the source and the target domains and less associated with the discrepancy between the source and the target domain distributions.

1. Introduction

1.1. Background

Artificial Intelligence (AI) and Machine Learning (ML) are developing rapidly and have shown tremendous success in various application domains [1,2]. One of the more powerful and essential characteristics of AI and ML is the capability to carry out tasks autonomously by understanding and analyzing a given dataset. This characteristic has led to a variety of research domains and industrial fields adopting AI and ML including object detection [3–5], natural language processing [6–8], autonomous vehicles [9–11], and applied science and engineering fields [12–15] among many others. This success has also translated to the realm of structural engineering. In recent years, substantial advancements have been made in an effort to apply ML to the structural engineering fields [16–19]. The previous studies have demonstrated the effectiveness of AI algorithms over traditional procedures pertaining to evaluation, decision-making, prediction and optimization in structural engineering applications.

One of the basic assumptions in a traditional ML algorithm is that the training and testing data must share not only the same feature space and distribution, but also the task. In other words, an individual learner can only acquire knowledge of a specific task from the identical feature space and distribution. Suppose that the trained model may need to predict a different task where the feature space or data distribution is not identical to the previously trained task but somewhat similar. In that case, the trained model will very likely yield poor performance and should be reconstructed from scratch based on new data. However, making a new ML model from scratch while maintaining good performance requires another hyperparameter tuning process and may be impractical if the new data samples are insufficient. Furthermore, in reality, obtaining more data samples involves additional costs and time, and often is not feasible.

One alternative way to address this problem is via Transfer Learning (TL). TL is a sub-field of ML and also known as knowledge transfer. Fig. 1 shows the schematic learning procedure of ML and TL. Generally, for training a typical ML model, an individual learner requires an individual dataset, as can be shown in Fig. 1(a). However, in Fig. 1(b), TL aims to

* Corresponding author.
E-mail addresses: hongrak822@tamu.edu (H. Pak), sgpaal@tamu.edu (S.G. Paal).

https://doi.org/10.1016/j.engstruct.2022.114579

Available online 2 July 2022

extract knowledge from one or more source tasks and apply the knowledge to a different target task [2]. The main concept of TL is to efficiently learn a target task with help of not only the target domain, but also from a single or multiple source domains or source tasks.

Several approaches have been proposed for transferring knowledge, e.g., instance-based transfer [20–24], feature-representation-transfer [25–27], parameter-transfer [28–31], and information transfer from unlabeled data [32,33]. While various theories and applications have been developed, a few TL studies have been conducted in the structural engineering domain [34,35] and a limited number of studies have been proposed to estimate structural capacity [36] compared to the other fields where TL is more actively used. Furthermore, the majority of TL research in the structural engineering domain is associated with image-based models [37–40]. As the relevance between the source and target domains and tasks is critical in the success of such approaches, more investigation into the application of transfer learning within the realm of structural engineering is necessary.

For structural engineering practices, in-depth understanding and prediction of the performance of existing or new structural components are essential for the effective design and maintenance of structures during their life cycle. In recent years, substantial advancements have been made in an effort to apply AI to the applied science and engineering fields, especially in civil engineering. Kakatand et al. [41] have proposed data-driven models to predict the shear strength of RC columns. 145 rectangular and 91 circular columns were used to train the models, and the maximum R^2 values from 10^6, 10^7, or 10^8 iterations of Monte Carlo simulation were reported. However, developing a robust and accurate AI model for any purpose, including to estimate the performance of a physical structure, should generally be accompanied by large amounts of data. Even in the field of structural engineering, to properly validate new structural materials, configurations, designs, and modeling techniques, a comprehensive experimental test setup is required. Therefore, by adopting knowledge transfer techniques into the structural engineering domain, researchers and practitioners will be able to efficiently use an ML model to quantify the structural capacity without excessive efforts to augment data samples.

1.2. Lateral strength of RC columns

Columns are the prime source of energy dissipation in a structural system and often the most critical components resisting seismic hazard [42]. Column failures are commonly classified as one of the following modes: flexure, shear, or flexure-shear failure. Flexure failure normally occurs after yielding of the longitudinal reinforcement and shear failure occurs before yielding of the longitudinal reinforcement. Many post-earthquake reconnaissance and researches have indicated that light and inadequately detailed transverse reinforcement are vulnerable to shear failure during seismic events [43–45]. Shear failure would drastically reduce the structural seismic performance and sometimes lead to structural collapse. Thus, special care should be needed to ensure enough amounts of transverse reinforcement to avoid shear failure prior to a flexural failure [46].

High-strength concrete (HSC) has been increasingly used in buildings and infrastructures because of its advantages, such as high strength, good durability, and reduction of member size. Due to these advantages of HSC, it displays more brittle behavior under the same reinforcement details, compared to the normal strength concrete (NSC) [47–49]. The lower ductility and undesirable brittleness of HSC restricts the use of HSC in the seismic regions.

With these perspectives of views on ductility and HSC, this study mainly deals with developing a knowledge transfer data-driven model that can precisely calculate the shear strength of the RC column even with the small number of non-ductile columns or HSC samples.

2. Transfer learning algorithms for regression problems

The main purpose of this study is to investigate different knowledge transfer techniques and compare the transferability of each method. Three knowledge transfer methods are considered: Instance Weighting Kernel Ridge Regression (IW-KRR) [22], Two-stage TrAdaBoost.R2 [50], and Double-Weighted Support Vector Transfer Regression (DW-SVTR) [36]. Based on the categories summarized by Pan and Yang [2], the problems dealt with in this study are all classified as inductive transfer learning problems, where the response variable in the source and target domains are available. This study focuses on an instance transfer approach belonging to inductive transfer learning. This is attributed to the fact that the source and target domains related to the input dataset used in this study already have an identical feature space after using dimensional reduction techniques. Furthermore, this study has used different types of base learners to compare various aspects of ML and TL. The base learner of the selected methods uses a different ML algorithm: Kernel Ridge regression (KRR), random forest, and a support vector machine. Each method is briefly introduced in the following sections.

2.1. Definition of transfer learning

Let $\mathscr{D}^S$ and $\mathscr{D}^T$ denote the source and target domain data, respectively. Given $\mathscr{D}^S$ and the source learning task $\mathscr{T}^S$, $\mathscr{D}^T$ and the target learning task $\mathscr{T}^T$, TL aims to help improve the learning of the target predictive function $f^T(\cdot)$ in $\mathscr{D}^T$ using the knowledge in $\mathscr{D}^S$ and $\mathscr{T}^S$,

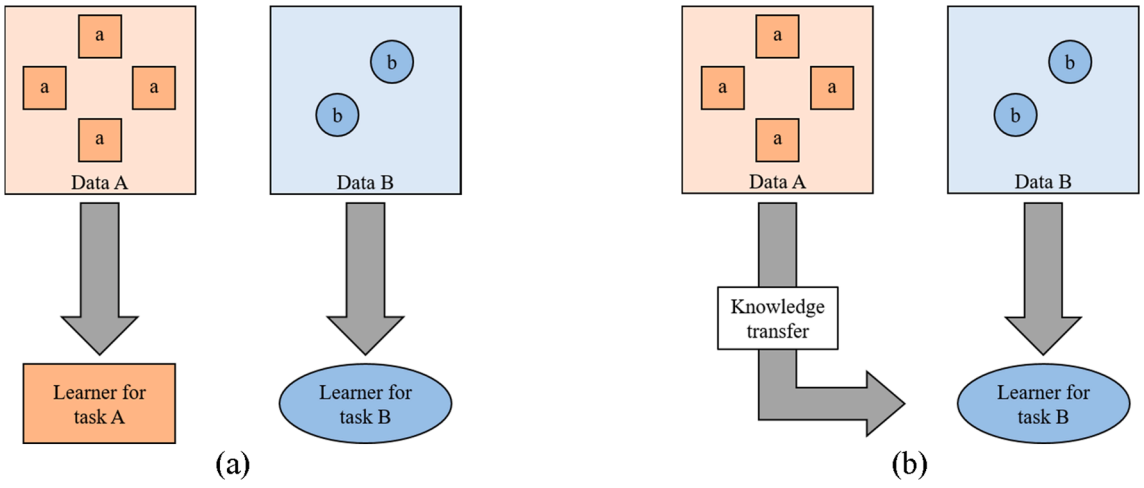


Fig. 1. Schematic learning procedure; (a) Machine Learning (ML) and (b) Transfer Learning (TL).

where $\mathscr{D}^S \neq \mathscr{D}^T$, or $\mathscr{T}^S \neq \mathscr{T}^T$. Even though $\mathscr{D}^S \neq \mathscr{D}^T$ or $\mathscr{T}^S \neq \mathscr{T}^T$, $\mathscr{D}^S$ can be a useful resource to improve the prediction for $\mathscr{D}^T$. This is the main idea behind TL, inspired by the human learning process. Based on this definition, some general notations used in the remainder of this discussion are defined as follows. $\mathscr{D}^S$ can be represented as $\left\{\left(\mathbf{x}_i^S, y_i^S\right)_{i=1}^n\right\}$ where n is the number of samples in the source domain. Similarly, $\mathscr{D}^T$ can be represented as $\left\{\left(\mathbf{x}_i^T, y_i^T\right)_{i=1}^m\right\}$, where m is the number of samples in the target domain. Here $\mathbf{x}_i \in \mathbb{R}^d$ is the i th explanatory variable vector, where d indicates the vector dimension, and y_i denotes the response variable, which could be a discrete variable for a classification problem or a continuous variable for a regression problem. In most cases, $m \ll n$, and probabilistic distributions of the source and target domains are not equal but somewhat related.

$$P^S\left(\mathbf{x}^S, y^S\right) \approx P^T\left(\mathbf{x}^T, y^T\right) \text{ for some } (\mathbf{x}, y) \tag{1}$$

where, P^S and P^T denote the probabilistic distribution of the source and target domain, respectively.

2.2. Kernel Ridge regression (KRR) for transfer learning

The most important part of reweighting instances in KRR is to determine samples from $\mathscr{D}^S$ which positively or negatively influence training and testing on $\mathscr{D}^T$. While there are some existing solutions to address this challenge, one effective way to find appropriate samples is to measure the similarity between the source and target distributions. Defining the importance weight function as $w(\mathbf{x}, y) := P^T(\mathbf{x}, y)/P^S(\mathbf{x}, y)$, the prediction or estimated value can be written as:

$$\begin{aligned} \widehat{y}^T &= \underset{y}{\operatorname{argmax}}\left(P^T\left(y|\mathbf{x}^T\right)\cdot P^T\left(\mathbf{x}^T\right)\right) \\ &= \underset{y}{\operatorname{argmax}}\left(P^T\left(y|\mathbf{x}^T\right)\cdot P^T\left(\mathbf{x}^T\right)\cdot\frac{1}{P^S\left(\mathbf{x}^T, y\right)}\cdot P^S\left(y|\mathbf{x}^T\right)\cdot P^S\left(\mathbf{x}^T\right)\right) \\ &= \underset{y}{\operatorname{argmax}}\left(w\left(\mathbf{x}^T, y\right)\cdot P^S\left(y|\mathbf{x}^T\right)\cdot P^S\left(\mathbf{x}^T\right)\right) \end{aligned} \tag{2}$$

In this study, a Gaussian kernel function is used for an approximation of the weight function,

$$\widehat{w}_i^{\alpha}(\mathbf{x}^*, y) = \alpha_i \cdot \exp\left(-\frac{\|(\mathbf{x}^*, y) - \left(\mathbf{x}_i^S, y_i^S\right)\|^2}{2\eta^2}\right) \tag{3}$$

where α_i is the coefficient of linear combination, η is a hyperparameter denoting the length scale of the kernel, and $\left(\mathbf{x}_i^S, y_i^S\right)$ denotes the center points for $i = 1, \cdots, n$.

The main goal of the regression problem is to minimize the residual sum of square error (RSS). Substituting Eq. (2) into the RSS equation and omitting the $P^S\left(\mathbf{x}_i^T\right)$ term since the argmax does not depend on it for a given sample $\mathbf{x}^T$, the RSS function for this problem can be represented as:

$$\min_{\widehat{w}} \sum_{i=1}^m \left(y_i^T - \underset{y}{\operatorname{argmax}}\left(\widehat{w}\left(\mathbf{x}_i^T, y\right)\cdot P^S\left(y|\mathbf{x}_i^T\right)\right)\right)^2 \tag{4}$$

Given that suitable weights were obtained, the common RSS error function that we need to minimize can be slightly modified as:

$$J_w(\theta) = \frac{1}{2}\left(\sum_{i=1}^m \left(y_i^T - \theta^{\mathrm{T}}\cdot\varphi\left(x_i^T\right)\right)^2 + \sum_{j=1}^n \widehat{w}_j^{\alpha}\left(x_j^S, y_j^S\right)\left(y_j^S - \theta^{\mathrm{T}}\cdot\varphi\left(x_j^S\right)\right)^2\right) + \frac{\lambda}{2}\|\theta\|^2 \tag{5}$$

where θ is the vector of model parameters, $\varphi(\cdot)$ is a feature mapping function that maps the input x into the feature space, and λ is the regularization parameter.

By defining the diagonal matrix $\widehat{\mathbf{W}} \in \mathbb{R}^{(m+n)\times(m+n)}$,

$$\widehat{\mathbf{W}} := \begin{bmatrix} \mathbf{I}_m & [0] \\ [0] & diag\left(w\left(\mathbf{x}_i^S, y_i^S\right)_{i=1\cdots n}\right) \end{bmatrix} \tag{6}$$

where $\mathbf{I}_m$ is the identity matrix with m dimensions.

The final prediction function with weighted terms can be defined as:

$$\widehat{y}^* = \underset{y}{\operatorname{argmax}}(\widehat{w}(\mathbf{x}^*, y)P(y|\mathbf{x}^*)) \approx f_{\widehat{w}(\mathbf{x}^*, y)}(\mathbf{x}^*) = \mathbf{a}^{\mathrm{T}}\widehat{\mathbf{W}}(\mathbf{x}^*, y)k_{-}(\mathbf{x}^*) \tag{7}$$

where the discriminative model f now depends on the weight function $\widehat{\mathbf{W}}(\mathbf{x}^*, y)$ which depends on a, $k_{-}(\mathbf{x}^*) := (k(\mathbf{x}_1, \mathbf{x}^*), \cdots, k(\mathbf{x}_n, \mathbf{x}^*))^{\mathrm{T}}$, $k(\mathbf{x}_i, \mathbf{x}^*) := \varphi(\mathbf{x}_i)^{\mathrm{T}}\varphi(\mathbf{x}^*)$, $\mathbf{a}^{\mathrm{T}}$ is the vector of dual coefficients, and $\mathbf{x}^*$ is a new data point.

By inserting Eq. (7) into Eq. (4), proper weights in the prediction function can be estimated as follows:

$$\min_{\alpha\geq 0} \sum_{i=1}^m \left(y_i^T - \mathbf{a}^{\mathrm{T}}\widehat{\mathbf{W}}^{\alpha}\left(\mathbf{x}_i^T, y_i^T\right)k_{-}\left(\mathbf{x}_i^T\right)\right)^2 + \gamma\|\alpha\|^2 \tag{8}$$

The last term in Eq. (8) is added to avoid overfitting and to penalize large coefficients. Now, the weight function $\widehat{\mathbf{W}}$ can be calculated with the help of the estimated α.

2.3. Two-stage TrAdaBoost.R2

Dai et al. [51] proposed an instance-based TL algorithm, TrAdaBoost, which extends the boosting-based method [52]. It was combined with AdaBoost.R2 [53] to develop a model to solve regression problems. It basically reduces the weight of an instance with a high error rate and increases the weight of an instance with a low error rate. However, two issues with the TrAdaboost.R2 algorithm were reported by Pardoe and Stone [50]. First, the weights of the target data may be heavily skewed if the size of the source data is much larger than the target data. Especially, the entire weight vector is highly dependent on some target instances that are either outliers or most dissimilar to the source data. Second, the weights of some source instances closely associated with the target task tend to eventually be reduced to zero. Based on these issues, the authors proposed a new version of TrAdaBoost.R2 where the weights are adjusted in two stages.

The first step of the Two-stage TrAdaboost.R2 is to set the initial weight vector $\mathbf{w}^1$,

$$w_i^1 = \frac{1}{n+m} \tag{9}$$

for $i = 1 \leq i \leq n+m$, where n is the number of source samples and m is the number of target samples.

In the first stage, the weights of source instances are designed to be gradually reduced until a certain point determined by cross validation. The next stage is to train a model on the dataset combining the source and target samples. The model used in this second stage is identical to the typical AdaBoost.R2, except that the weights of the source data will not be changed. The adjusted error e_i^t for each instance is calculated, and the weight vector is updated based on the following rule:

$$w_i^{t+1} = \begin{cases} \dfrac{w_i^t \beta_t^{e_i^t}}{Z_t}, & (1 \le i \le n) \\ \dfrac{w_i^t}{Z_t}, & (n+1 \le i \le n+m) \end{cases} \tag{10}$$

where Z_t is a normalizing constant, and β_t is determined such that the resulting weight of the target instances is $\frac{m}{n+m} + \frac{t}{S-1}\left(1 - \frac{m}{n+m}\right)$.

With this updating rule, the first total weight of the target instances starts from $\frac{m}{n+m}$, and increases uniformly up to 1. The binary search algorithm approximately searches the value of β_t. The procedure in this algorithm will be terminated when the errors start to increase.

2.4. DW-SVTR

A novel regression-based TL approach was proposed by Luo and Paal [36], which is called Double-weighted support vector transfer regression (DW-SVTR). It is extended from Least squares support vector machines for regression (LS-SVMR) by coupling two methods to effectively estimate instance weights. Firstly, kernel mean matching (KMM) is used to reweight the source domain samples such that the mean values of the source and target domain in a reproduced kernel Hilbert space are close. With this technique, the source domain samples relevant to the target domain samples have a larger weight than irrelevant source domain samples. The second weight is a function of estimated residuals to reduce the negative interference of irrelevant source domain samples. Given the dataset combining the source and target domain, the objective function of DW-SVTR can be expressed as follows:

$$\begin{aligned} J(\theta, e_i) &= \frac{1}{2}\theta^{\mathrm{T}}\theta + \frac{1}{2}\gamma \sum\nolimits_{i=1}^{m+n} w(\mathbf{z}_i) v(\mathbf{x}_i) e_i^2 \\ \text{Subject to} &: y_i = \theta^{\mathrm{T}}\varphi(\mathbf{x}_i) + b + e_i, \end{aligned} \tag{11}$$

where e_i is the error term for $i = 1 \cdots (m+n)$, γ is a regularization parameter, $w(\mathbf{z}_i)$ is a weight to determine the importance of each data point, $v(\mathbf{x}_i)$ is a weight function of the residuals, $\varphi(\cdot)$ is a mapping function into a higher dimensional space, and θ is a model parameter vector.

By using the Lagrange multiplier method with Karush-Kuhn-Tucker (KKT) conditions and solving the quadratic programming problem, the DW-SVTR algorithm eventually finds the optimal weight vector such that the objective function $J(\mathbf{w}, e_i)$ is minimized.

3. Evaluation metrics

3.1. Discrepancy measure

There should be one or more source domain(s) and target domain in the TL problem. The source and target domains are somewhat related but not identical. One of the crucial factors closely associated with the success of TL is how far away those source and target domains are. Obviously, the probability of success of the TL technique would be lower if the distance between the source and target domains is larger. Thus, it is important to understand the relationship and distance between the source and target domains, which heavily influence the success of knowledge transfer. Some statistics can adequately estimate the distance metric between two different probabilistic distributions. When it comes to estimating the distance between two probability distributions, Kullback-Leibler divergence, widely used for a measure of how one distribution is different from another, does not satisfy the symmetric condition and the triangle inequality condition. Therefore, it cannot be used as a statistical measurement of discrepancy. In this study, three distance metrics, Maximum Mean Discrepancy (MMD), Earth Mover's Distance (EMD), and Hellinger distance, are adopted to quantify the distance on the space of probability measures.

MMD is a relevant criterion for comparing distributions based on the Reproducing Kernel Hilbert Space (RKHS). It can be well-estimated by the distance between the means of the two distributions mapped into the RKHS. Unlike Kullback-Leibler divergence, which requires an intermediate probability density estimation, MMD is a non-parametric distance estimate between those distributions. The empirical MMD is defined as follows:

$$MMD(X, Y) = \| \frac{1}{n_1} \sum_{i=1}^{n_1} \varphi(x_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} \varphi(y_i) \|_{\mathscr{H}} \tag{12}$$

where $X = \{x_1, x_2, \cdots, x_{n_1}\}$, $Y = \{y_1, y_2, \cdots, y_{n_2}\}$, $\mathscr{H}$ is a universal RKHS, and φ is a kernel function mapping samples from $\mathscr{X}$ to $\mathscr{H}$.

The Wasserstein metric, also known as EMD (Earth Mover's Distance), is a distance function defined between probability distributions. EMD was proposed to measure a discrepancy between two distributions by quantifying the optimal cost of rearranging one distribution into the other. With this analogy, it is frequently referred to as the earth mover's distance. The EMD between two distributions f and g is:

$$EMD(f, g) = \int_{-\infty}^{+\infty} |F(x) - G(x)| dx \tag{13}$$

where F and G are the CDFs of f and g, respectively.

The Hellinger distance (HD) is a measure of divergence of two distributions and provides another way to estimate the distance between those distributions independent of parameters. It should be noted that the HD satisfies the triangle inequality, and the $\sqrt{2}$ coefficient in Eq. (14) is for ensuring that $HD(p, q) \le 1$. For the probability density functions p and q, the Hellinger distance between them can be expressed as:

$$HD(p, q) = \frac{1}{\sqrt{2}} \sqrt{\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx} \tag{14}$$

3.2. Prediction performance measure

In order to measure the model performance and estimate the error, the root mean square error (RMSE) is monitored during the training process. Root mean square error (RMSE) and the coefficient of determination (R^2) are employed to directly compare the generalization ability of typical ML models and knowledge transfer in a comprehensive manner. In general, R^2 is a good indication of the generalized model capabilities in comparison to other models in regression problems. Additionally, RMSE is widely used as one of the metrics that can indicate the model performance, especially whether or not the model is sensitive to outliers. An R^2 value closer to 1 and RMSE value closer to 0 indicate better performance and better generalization ability to predict the response variable(s). In addition, an R^2 value is not limited to a lower bound of zero, if the model prediction is worse than just using the mean value. The following equations show the definitions of MSE (Mean Square Error) and R^2. RMSE is the square root of MSE.

$$MSE = \frac{1}{n} \sum_{i-1}^{n} (y_i - \widehat{y}_i)^2 \tag{15}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} \tag{16}$$

where n is the number of samples, y_i is the actual response value, $\widehat{y}_i$ is the predicted response value, and $\overline{y}$ denotes the mean of the actual response value.

4. Experiment and results

4.1. Dataset description

In order to compare different TL techniques, a comprehensive

dataset should be acquired to train a model and appropriately evaluate its performance. In this study, a dataset consisting of 497 reinforced concrete (RC) column samples [54,55] is used to assess three different knowledge transferring techniques. One sample is excluded from the 498 columns of the original dataset since its experimental result is significantly outside of the range covered by the remaining columns. The dataset comprises two different section types and a wide range of shear reinforcement ratios and concrete compressive strengths. Twelve explanatory variables are extracted or selected from the 30 explanatory variables in the original dataset by calculating the feature importance scores. The relative scores from the calculation of the feature importance highlight which features may be more relevant to the response variable and which features are less relevant. The statistical information of 12 independent features and the response variable are summarized in Table 1.

Three experimental cases were considered for evaluating and comparing TL methods, as shown in Table 2. The cases are designed with similar but slightly different source and target domains. For *Case 1*, the criteria of the source and target domain are the shape of column section. For *Case 2*, the criteria is based on the equations in ACI 318–19 [46]. The following equations show $A_{v,min}$ defined in ACI 318–19, and the greater value should be chosen:

$$A_{v,min} = \begin{cases} 0.75\sqrt{f_c'}\,\dfrac{b_w s}{f_{yt}} \\ \\ 50\dfrac{b_w s}{f_{yt}} \end{cases} \tag{17}$$

where f_c' is the concrete compressive strength, b_w is the width, s is the spacing, and f_{yt} is the yield strength of transverse reinforcement.

For *Case 3*, the samples where the compressive strength is higher than 82.74 MPa (12000 psi) are classified as the target domain and the samples with less than or equal to 82.74 MPa (12000 psi) of the compressive strength are classified as the source domain. These three cases will illustrate the performance of the three TL techniques in knowledge transfer across section type, area of shear reinforcement, and concrete compressive strength values. For all cases, the maximum lateral shear strength is used as the response variable. Although the shear mechanism of RC structures can be divided into the contributions of concrete and steel reinforcement, this study focuses on the more comprehensive depiction of the shear behavior by directly estimating the maximum lateral shear strength of RC columns.

Table 1
Statistical information of the RC column dataset.

Description	Unit	Average	Standard deviation	Minimum	Maximum
Area	mm^2	121,432	112,703	6400	1,814,583
Effective depth	mm	294.13	117.09	62.99	1215.90
Shear span	mm	1179.32	837.69	80.01	9139.94
Yield stress of longitudinal rebar	MPa	423.54	62.20	239.94	586.90
Longitudinal reinforcement ratio	–	0.0242	0.0099	0.0046	0.0694
Clear length	mm	1351.03	817.12	160.02	9139.94
Transverse reinforcement legs parallel to primary load	–	2.60	0.88	2.00	6.00
Spacing of transverse rebar	mm	94.49	82.55	8.89	457.20
Yield stress of transverse rebar	MPa	450.76	185.75	199.95	1423.63
Transverse reinforce ratio	–	0.0059	0.0048	0.0004	0.0321
Concrete compressive strength	MPa	43.10	24.43	13.10	117.97
Axial load	KN	927.01	1160.99	0.00	7999.68
Maximum lateral load	KN	233.44	187.36	19.04	1338.69

Table 2
Experimental cases.

Case No.	Dataset partition	Section type	Number of samples
Case 1	Source	Rectangular	326
	Target	Circular	171
Case 2	Source	$A_v \geq A_{v,min}$	455
	Target	$A_v < A_{v,min}$	42
Case 3	Source	Normal strength concrete	443
	Target	High strength concrete	54

Note: A_v = area of shear reinforcement within spacing; $A_{v,min}$ = minimum area of shear reinforcement within spacing.

Table 3
Results of the discrepancy metrics.

Case No.	*MMD*	*EMD*	*HD*
Case 1	0.0119	11.1395	0.1483
Case 2	0.0305	25.4353	0.2731
Case 3	0.0254	12.4887	0.2191

The distances between the source and target domain in each case are different based on their physical characteristics. Thus, prior to analyzing the results of knowledge transferring techniques, the discrepancy metrics introduced in Section 3 should be calculated and compared. The discrepancy metrics calculated from the three cases are shown in Table 3. *Case 2* has the most different distributions between the source and target domains and *Case 1* has the most similar distributions, regardless of which discrepancy metrics are used. The number of samples used for training and testing a model in each target domain availability is summarized in Table 4. While typical ML models use only the target domain samples in the training process, the TL models use both the source and target domain samples. In order to effectively find the optimal hyperparameters for each model, the Bayesian optimization process, named Hyperopt, developed by James Bergstra [56] is adopted in this study. It provides an automated hyperparameter optimization process and generally requires a lower number of iterations when compared to the random search or grid search methods, assuming that the given search space is the same. For each case, the target domain availability for training a model is increased from 10% of the entire target domain up to 70% of the entire target domain. The remaining 30% of the target domain will be used to test the trained model. To accurately evaluate the trained model, training and test sets are mutually exclusive from one another. For each case, ten experiments are conducted with randomly selected training and test sets. Thus, the generalized performance of knowledge transfer across the domains can be well observed by averaging the results from those ten experiments. Training and validation loss have been monitored over the training process in every model and trial, and overfitting was never observed. The transfer learning model is trained with a knowledge transfer technique and tested on the target domain. On the other hand, the baseline model is trained and tested on the target domain without any source domain information. This enables us to directly and effectively compare the knowledge transferability across different domains.

4.2. Case 1: Knowledge transfer across RC column section type

Case 1 has the most similar distributions between the source and target domains, as shown in Table 3. It also has the largest number of target domain samples among the three cases, as is shown in Table 2.

Table 4
The number of samples used for training and testing a model in each case.

Target domain availability [%]			10	15	20	25	30	35	40	45	50	55	60	65	70
Case 1	Source	# of training samples	326	326	326	326	326	326	326	326	326	326	326	326	326
	Target	# of training samples	17	26	34	43	51	60	68	77	86	94	103	111	119
		# of testing samples	52	52	52	52	52	52	52	52	52	52	52	52	52
Case 2	Source	# of training samples	455	455	455	455	455	455	455	455	455	455	455	455	455
	Target	# of training samples	4	6	8	10	13	15	17	19	21	23	25	27	29
		# of testing samples	13	13	13	13	13	13	13	13	13	13	13	13	13
Case 3	Source	# of training samples	443	443	443	443	443	443	443	443	443	443	443	443	443
	Target	# of training samples	5	8	11	14	16	19	22	24	27	30	32	35	37
		# of testing samples	17	17	17	17	17	17	17	17	17	17	17	17	17

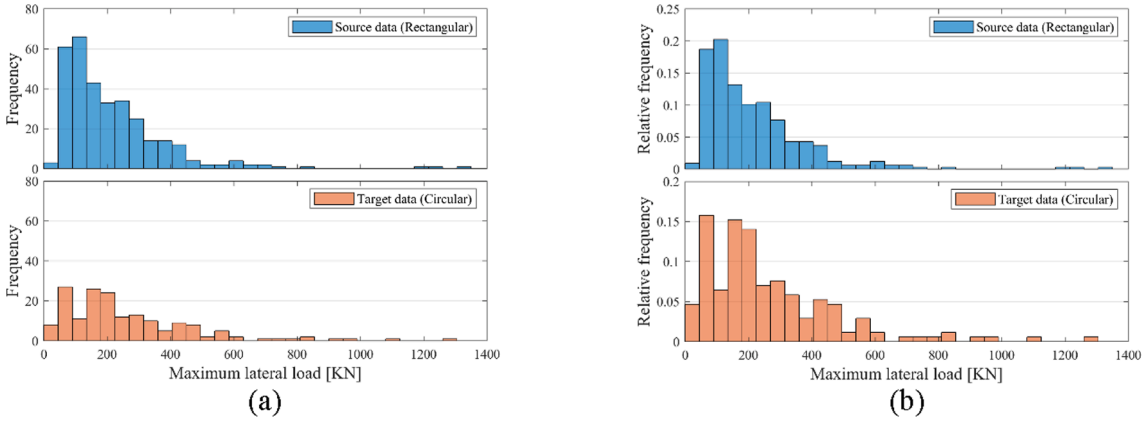


Fig. 2. Histograms of the source and target domains used in *Case 1*; (a) Frequency, and (b) Relative frequency (probability).

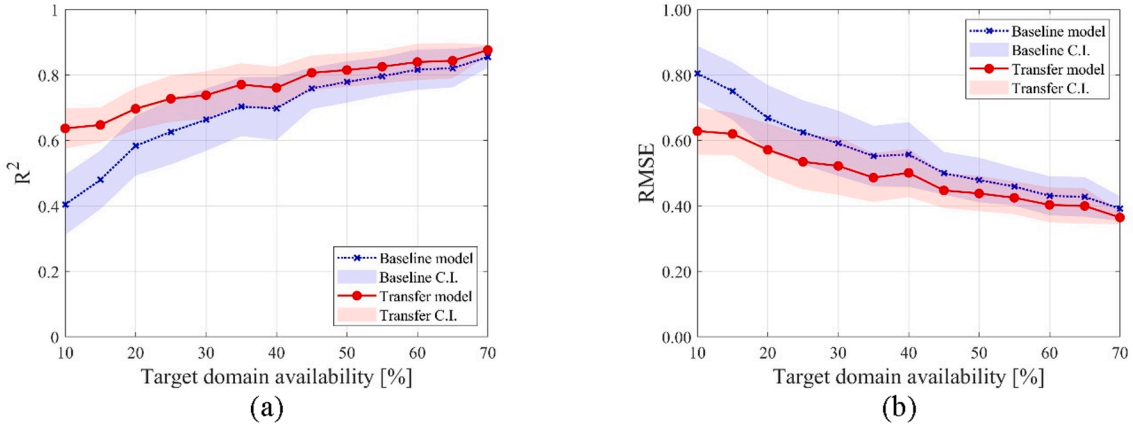


Fig. 3. IW-KRR performance in *Case 1* between the typical ML model and TL model; (a) The coefficient of determination, and (b) Root mean square error.

This characteristic can also be observed in Fig. 2, which shows the histograms of the source and target domains. As shown in Fig. 2, the ranges of the response variable between the source and target domains are similar, and their relative frequency also looks similar. A mutually exclusive set of 52 circular column samples is used to test the typical ML model and TL model. As the target domain availability increases from 10% to 70%, the number of samples for training a model increases from 17 to 119.

The performance comparisons of the typical ML model and TL model obtained from IW-KRR, Two-stage TrAdaBoost.R2, and DW-SVTR are depicted in Fig. 3, Fig. 4, and Fig. 5, respectively. The shaded areas in the figures indicate 95% confidence intervals from the 10 trials. Compared to the baseline model, it is observed that both R^2 and RMSE from the TL model are improved. Furthermore, the lower the percentage of target domain availability, the more significant the improvement between the baseline and TL model. Such trends can be observed in every TL technique used in this study. The prediction performance in 10, 40, and 70% of target domain availability from the three TL techniques are compared in Fig. 6(a). DW-SVTR shows the highest R^2 value not only from the case of low target domain availability but also for the case of high target domain availability. However, according to Fig. 6(b), which shows the differences between the typical ML model and the TL model, DW-SVTR has the smallest differences in all target domain availability, and the Two-stage TrAdaBoost.R2 model shows the most potent knowledge transferability. This is because DW-SVTR and typical LS-SVMR have good prediction performance, even for the low target domain availability.

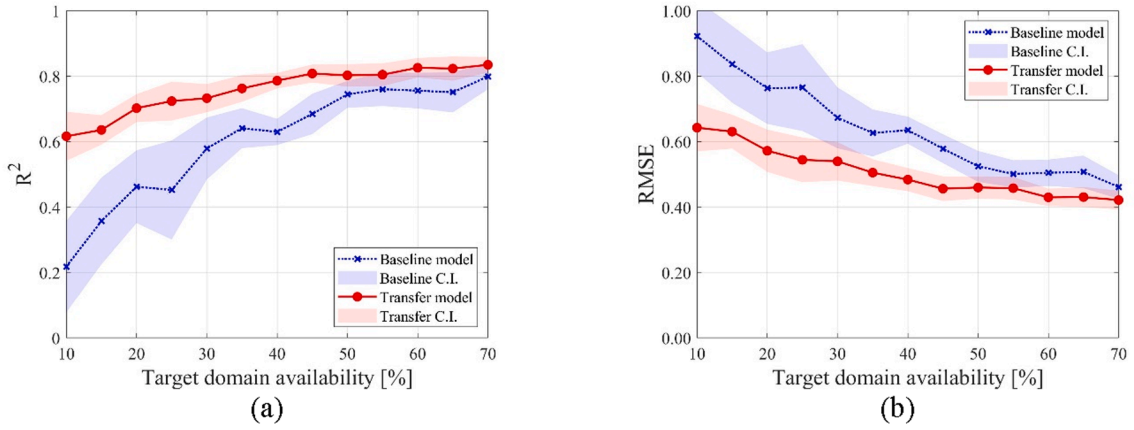


Fig. 4. Two-stage TrAdaBoost.R2 performance in *Case 1* between the typical ML model and TL model; (a) The coefficient of determination, and (b) Root mean square error.

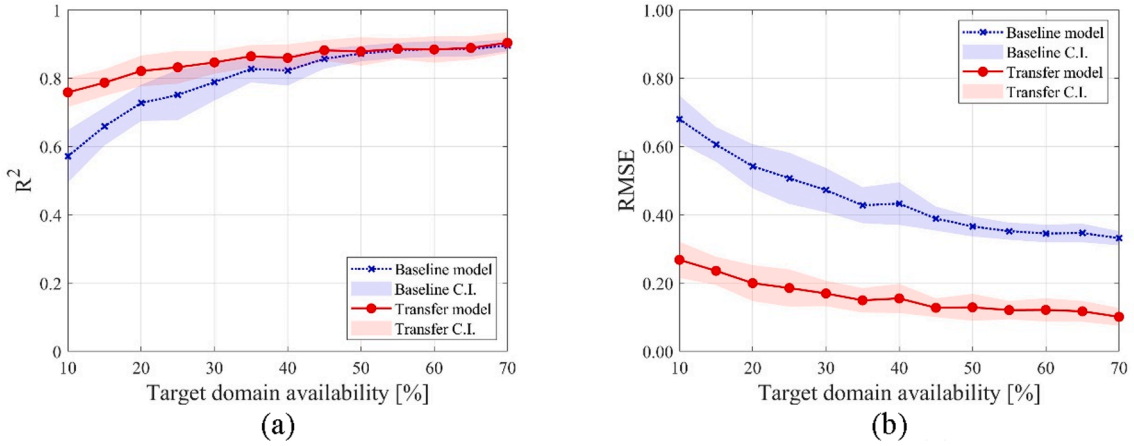


Fig. 5. DW-SVTR performance in *Case 1* between the typical ML model and TL model; (a) The coefficient of determination, and (b) Root mean square error.

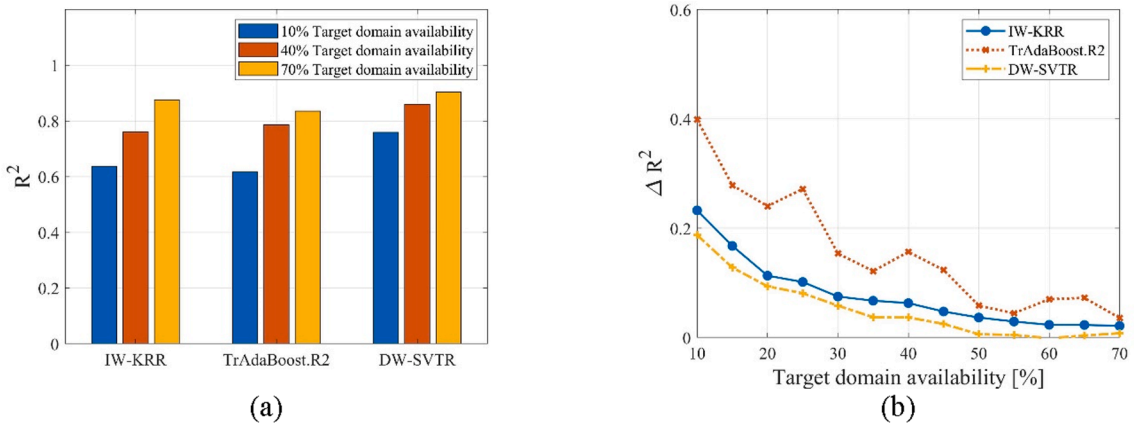


Fig. 6. Summary of knowledge transfer performance in *Case 1*; (a) bar chart in 10, 40, and 70% target domain availability from three different TL techniques, and (b) differences of R^2 between the typical ML model and TL model in terms of target domain availability.

4.3. Case 2: Knowledge transfer across shear reinforcement amounts

The source and target distributions for *Case 2* are the most dissimilar among the three experimental cases. In accordance with the results presented in Table 3, the shape of the distribution and the range of the response variable are different, as shown in Fig. 7. Furthermore, in this case, only 42 samples with lower shear reinforcement area are classified as the target domain, which is the smallest target domain size of all three cases. As the target domain availability increases from 10% to 70%, the number of samples for training a model increases from 4 to 29. The 10% of target domain availability is the most challenging scenario where only four samples are used to train a model. It is difficult, perhaps impossible, to get a well-generalized model if the typical ML model would be used. The performance comparisons of the typical ML model and TL model

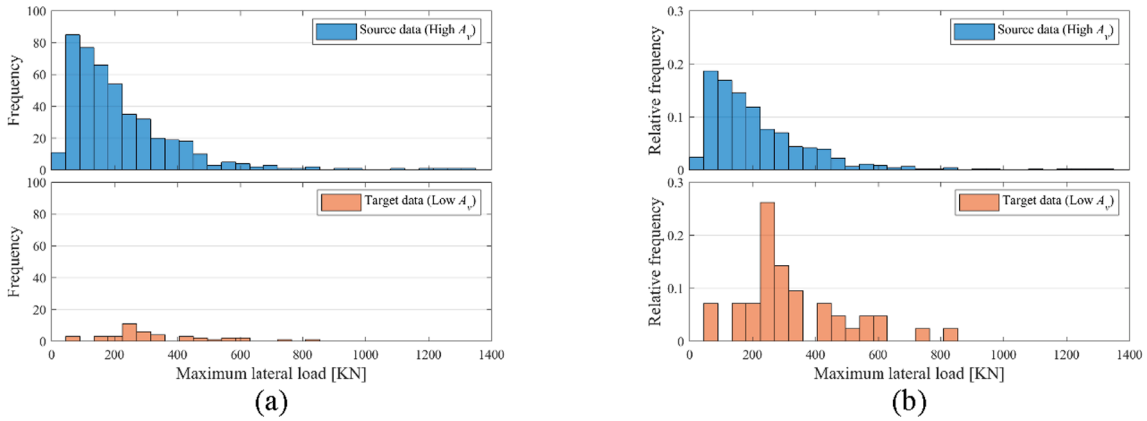


Fig. 7. Histograms of the source and target domains used in *Case 2*; (a) Frequency, and (b) Relative frequency (probability).

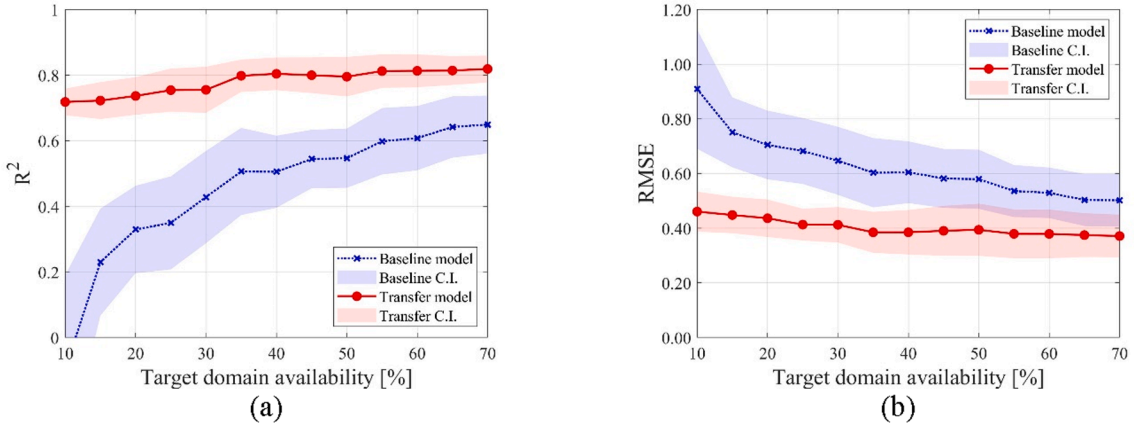


Fig. 8. IW-KRR performance in *Case 2* between the typical ML model and TL model; (a) The coefficient of determination, and (b) Root mean square error.

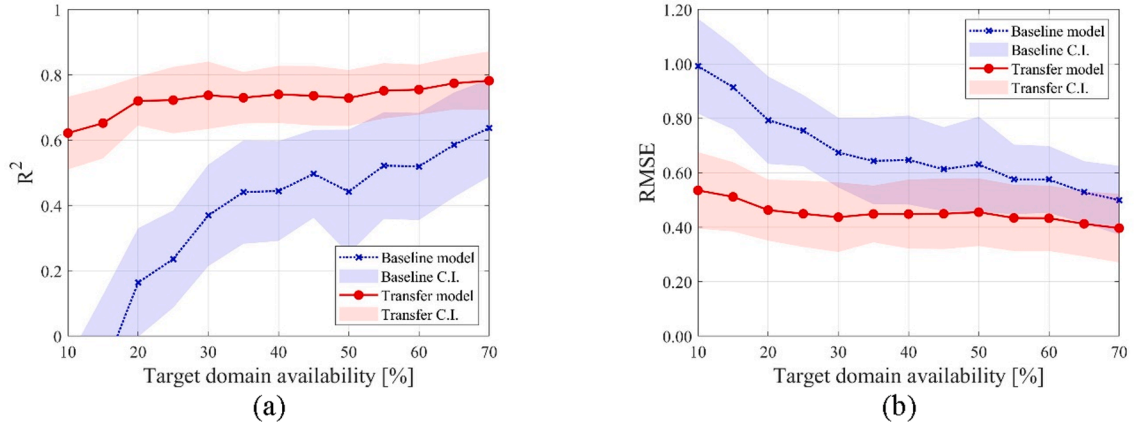


Fig. 9. Two-stage TrAdaBoost.R2 performance in *Case 2* between the typical ML model and TL model; (a) The coefficient of determination, and (b) Root mean square error.

obtained from IW-KRR, Two-stage TrAdaBoost.R2, and DW-SVTR are depicted in Fig. 8, Fig. 9, and Fig. 10, respectively. Similar to the trends observed in *Case 1*, R^2 and RMSE from the typical ML model and TL model are improved as the available data from the target domain increases. Also, the less target domain data the model uses, the more remarkable the improvement between the baseline and TL for every TL technique used in this case. Particularly, in *Case 2,* compared to the other cases, large areas are observed between the performance of the typical ML model and TL model. In other words, the most significant improvements by using knowledge transfer techniques occur in *Case 2*. This is because there is a physically strong correlation between the source and target domain, even if those two domains are the most dissimilar. The strong correlation is not just limited to the relationship between source and target domain. It is also strongly linked to each domain and the task learned by the model. The amount of shear reinforcement has a significant influence on the lateral capacity of RC columns, when compared to the shape of the section or concrete compressive strength. Even in the low target domain availability, the

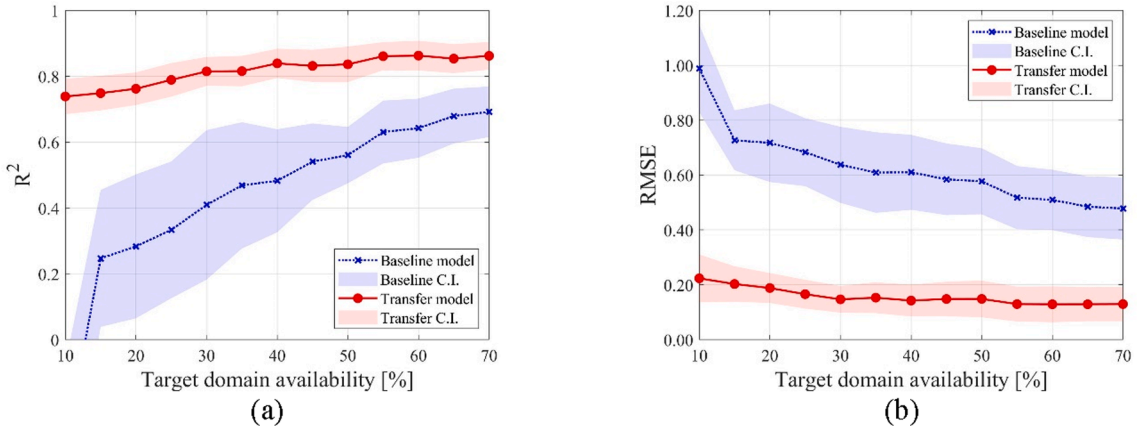


Fig. 10. DW-SVTR performance in *Case 2* between the typical ML model and TL model; (a) The coefficient of determination, and (b) Root mean square error.

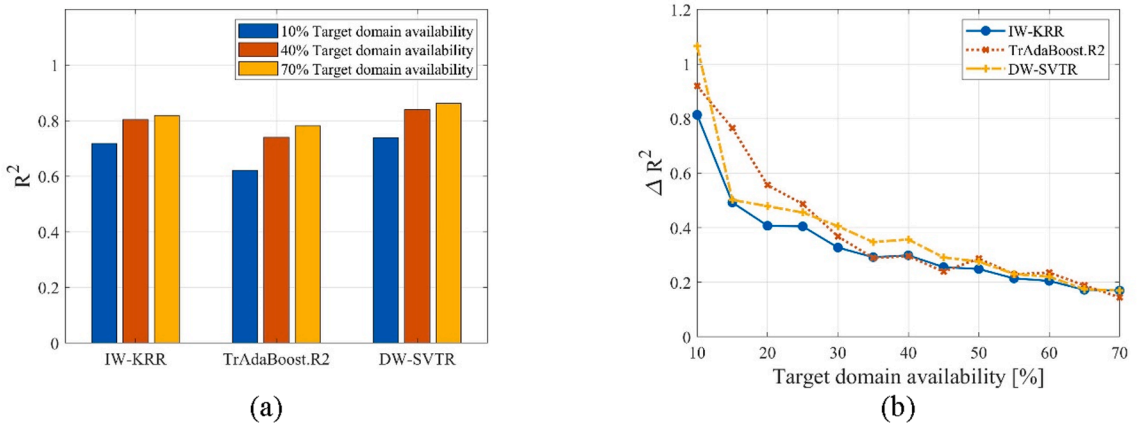


Fig. 11. Summary of knowledge transfer performance in *Case 2*; (a) bar chart in 10, 40, and 70% target domain availability from three different TL techniques, and (b) differences of R^2 between the typical ML model and TL model in terms of target domain availability.

prediction performance is reasonably good from all TL techniques in *Case 2*. This fact also supports the robust physical correlation between the source domain, target domain, and the lateral capacity of RC columns. The prediction performance in 10%, 40%, and 70% of target domain availability from the three TL techniques are compared in Fig. 11(a). DW-SVTR shows the highest R^2 value for the case of low target domain data availability and high target domain data availability. All the TL techniques in *Case 2* have good abilities to capture the behavior of the target domain even with a very small number of target samples, according to Fig. 11.

4.4. Case 3: Knowledge transfer across concrete compressive strength values

The calculated discrepancy metrics for *Case 3* are higher than *Case 1* and less than *Case 2*, as shown in Table 3. This means that *Case 3* has a moderate disparity between the source and target distributions, when compared to *Case 1* and *Case 2*. Fig. 12 shows the distributions of the source and target domains. In this case, 54 samples are used as the target domain, which is slightly larger than *Case 2*, but much lower than *Case 1*. Similar to the experimental setting used in *Case 1 and 2*, the target

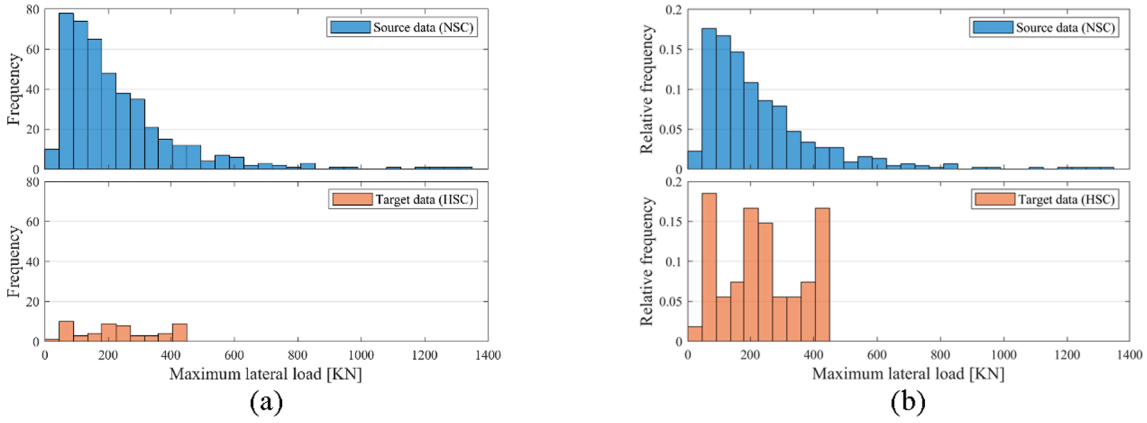


Fig. 12. Histograms of the source and target domains used in *Case 3*; (a) Frequency, and (b) Relative frequency (probability).

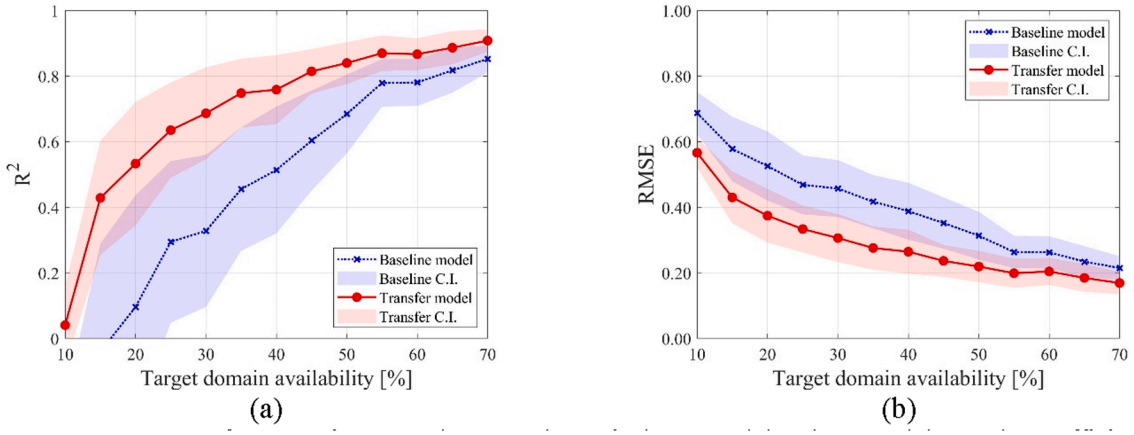


Fig. 13. IW-KRR performance in *Case 3* between the typical ML model and TL model; (a) The coefficient of determination, and (b) Root mean square error.

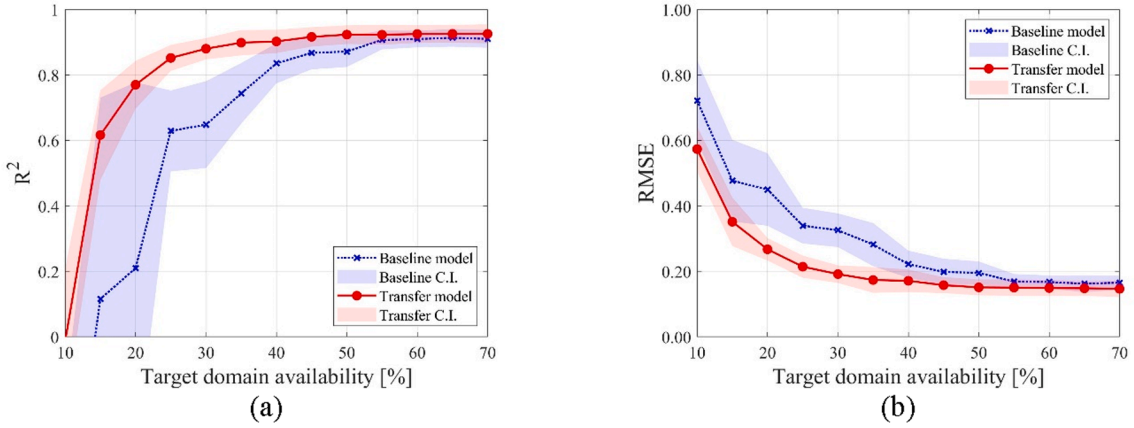


Fig. 14. Two-stage TrAdaBoost.R2 performance in *Case 3* between the typical ML model and TL model; (a) The coefficient of determination, and (b) Root mean square error.

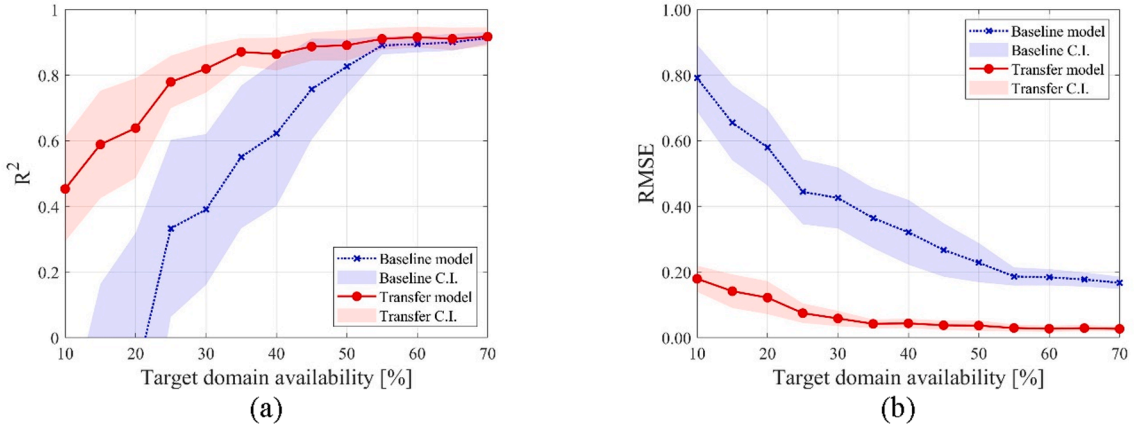


Fig. 15. DW-SVTR performance in *Case 3* between the typical ML model and TL model; (a) The coefficient of determination, and (b) Root mean square error.

domain availability increases from 10% to 70%, and the remaining 30% of target samples are used as a test set. The 10% of target domain availability corresponds to the scenario where only five samples are used for training a model. The performance comparisons of the typical ML model and TL model obtained from IW-KRR, Two-stage TrAdaBoost.R2, and DW-SVTR are depicted in Fig. 13, Fig. 14, and Fig. 15, respectively. Similar trends can also be observed in *Case 3*. Regardless of whether or not the knowledge transfer technique is employed, the prediction performance is improved as the available samples increase. Additionally, monotonic increasing behaviors for R^2 values of the typical ML and TL model are observed. Understandably, monotonic decreasing behaviors are observed for the RMSE values. Compared to the other two cases, *Case 3* struggles to obtain a well-generalized model, especially in the low target domain availability. Such results are attributed to the physically weak relationship between the source and target domain. The prediction performance for 10%, 40%, and 70% of target domain availability from the three TL techniques are compared in Fig. 16(a). Although the R^2 and

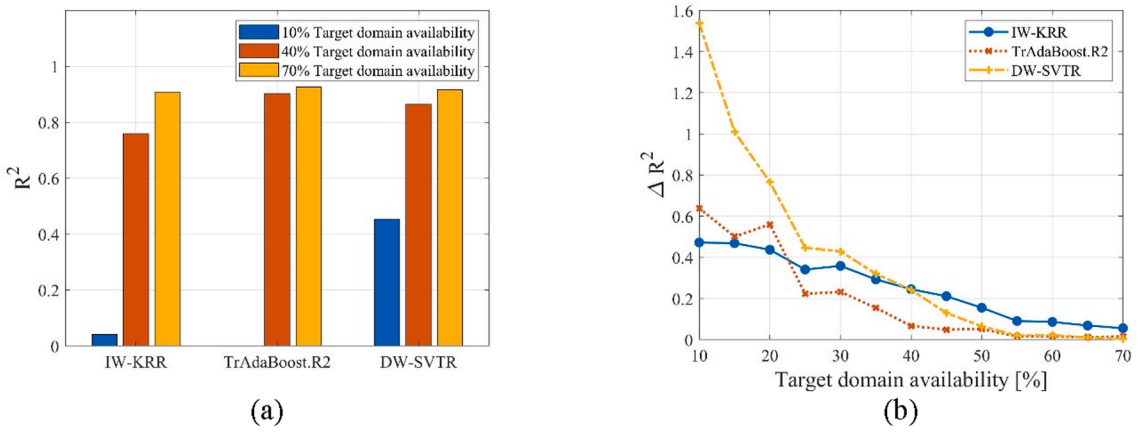


Fig. 16. Summary of knowledge transfer performance in *Case 3*; (a) bar chart in 10, 40, and 70% target domain availability from three different TL techniques, and (b) differences of R^2 between the typical ML model and TL model in terms of target domain availability.

RMSE values for the low target domain availability are not good compared to the other two cases, the results from DW-SVTR exhibit well-generalized prediction among the TL techniques in this case, as shown in Fig. 16(b).

5. Discussion

A relevant and well-established dataset is an essential aspect of any data-driven ML model. Training a good ML model with a very limited number of training samples is almost impossible. However, as reported in the earlier section, a better data-driven model can be obtained by adopting an appropriate knowledge transfer technique. For direct comparisons of the ML and TL models in terms of the required amount of training samples, the required target domain availability to get an acceptable data-driven model is summarized in Table 5. The required target domain availability is defined as the first target domain availability where the upper limit of R^2 confidence interval is equal to or higher than 0.85. For Two-stage TrAdaBoost.R2 in *Case 1* and every model in Case 2, the prediction performance of the ML model does not reach 0.85 until 70% target domain availability. An inequality sign denotes that the number of training samples is not enough to get a good ML model, and collecting or adding more training samples is required. According to Table 5, the required number of target training samples for a good TL model is less than that for a good ML model in all cases and models considered in this study. These results support the idea that transferring knowledge from the source domain can reduce the number of samples necessary to properly train a model; thus, TL can successfully

Table 5
Required target domain availability to get an upper limit of R^2 confidence interval equal to or higher than 0.85.

Case No.	Model name	Target domain availability [%]		Reduction of the required data [%], $((a-b)/a*100)$
		ML model (a)	TL model (b)	
Case 1	IW-KRR	55%	45%	18.18
	TrAdaBoost.R2	≥ 70%	60%	≥14.29
	DW-SVTR	35%	20%	42.86
Case 2	IW-KRR	≥ 70%	40%	≥42.86
	TrAdaBoost.R2	≥ 70%	65%	≥7.14
	DW-SVTR	≥ 70%	30%	≥57.14
Case 3	IW-KRR	55%	35%	36.36
	TrAdaBoost.R2	40%	25%	37.50
	DW-SVTR	40%	25%	37.50

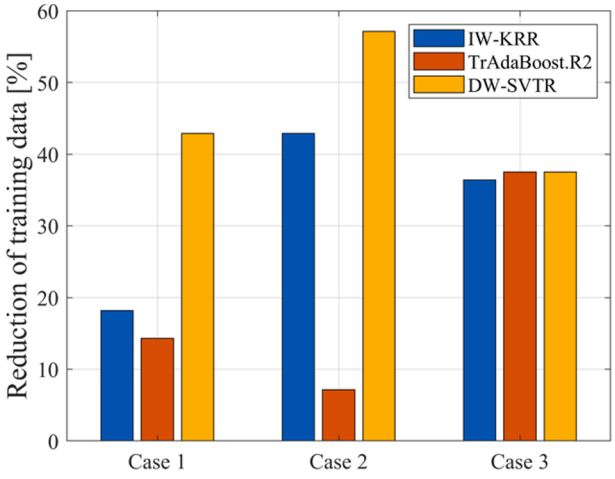


Fig. 17. Comparison of the knowledge transfer models in terms of reducing the required amount of training data.

alleviate the data scarcity problem. Fig. 17 shows a bar chart depicting the performance of knowledge transfer techniques in terms of the required amount of training data. As can be seen in Fig. 17, DW-SVTR has the greatest ability in terms of reducing the required number of training samples. It reduces the number of training data up to 57%, when compared to an ML model without any integrated knowledge transfer technique. This is because DW-SVTR uses two different instance weight functions simultaneously, which are KMM and a residual function. IW-KRR shows better abilities in terms of reducing the required training samples compared to the Two-stage TrAdaBoost.R2 algorithm.

The equations for estimating the shear strength of a non-prestressed reinforced concrete member are specified in ACI 318–19 [46]. Depending on the criteria introduced earlier in Eq. (17), the shear strength of a non-prestressed concrete member, V_n, is given as:

$$V_n = \begin{cases} \left[8\lambda(\rho_w)^{\frac{1}{3}}\sqrt{f_c'} + \dfrac{N_u}{6A_g}\right] b_w d + \dfrac{A_v f_{yt} d}{s} \left(A_v \geq A_{v,min}\right) \\ \left[8\lambda_s\lambda(\rho_w)^{\frac{1}{3}}\sqrt{f_c'} + \dfrac{N_u}{6A_g}\right] b_w d + \dfrac{A_v f_{yt} d}{s} \left(A_v < A_{v,min}\right) \end{cases} \quad (18)$$

where λ is the modification factor for lightweight concrete, ρ_w is the longitudinal reinforcement ratio, N_u is axial force, A_g is the cross-sectional area, d is the effective depth of the cross-section, A_v is the area of transverse reinforcement, and λ_s is the size effect modification factor.

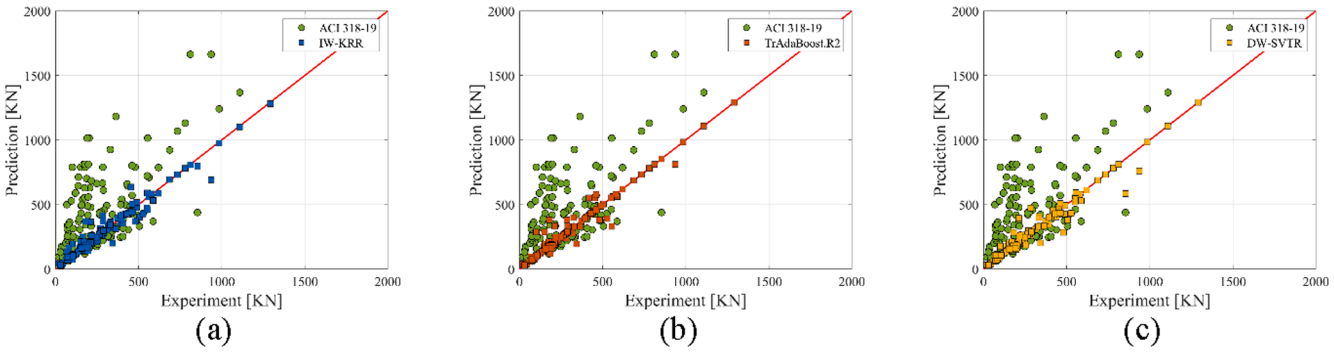


Fig. 18. Comparison of the predicted values in *Case 1* based on ACI 318–19 versus the TL models; (a) IW-KRR, (b) Two-stage TrAdaBoost.R2, and (c) DW-SVTR.

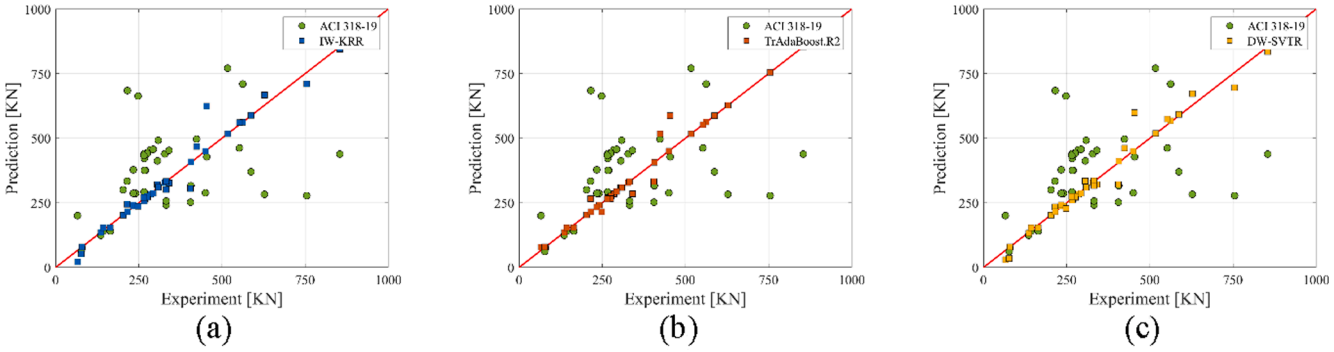


Fig. 19. Comparison of the predicted values in *Case 2* based on ACI 318–19 versus the TL models; (a) IW-KRR, (b) Two-stage TrAdaBoost.R2, and (c) DW-SVTR.

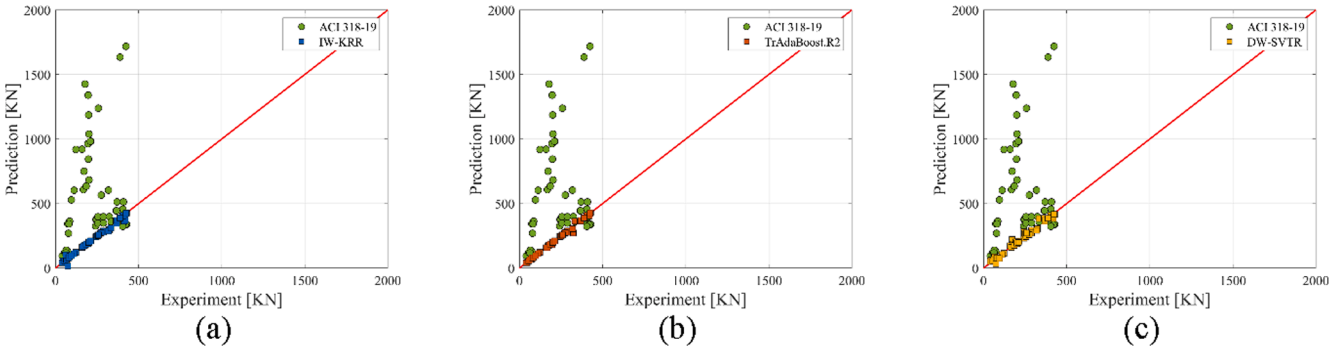


Fig. 20. Comparison of the predicted values in *Case 3* based on ACI 318–19 versus the TL models; (a) IW-KRR, (b) Two-stage TrAdaBoost.R2, and (c) DW-SVTR.

ACI 318–19 imposes a maximum value of 100 psi on $\sqrt{f_c'}$ for use in the calculation of shear strength of concrete members, because of a lack of test data and practical experience with concretes having compressive strengths greater than 10,000 psi [46]. Values of $\sqrt{f_c'}$ greater than 100 psi shall be permitted only for reinforced concrete members satisfying $A_v \geq A_{v,min}$. They also require the upper limit of 60,000 psi on the value of f_{yt} to control diagonal crack widths. With all these design specifications, the shear strength of RC columns in the target domain are calculated, and then compared with the shear strength values predicted by the three TL models in each case. The scatterplots comparing the predicted values based on ACI 318–19 and the TL models in *Case 1*, *Case 2*, and *Case 3* are depicted in Fig. 18, Fig. 19, and Fig. 20, respectively. Every TL model in all cases shows great ability to accurately predict the lateral capacity of RC columns with good R^2 values higher than 0.95. However, there are substantial differences between the calculated values from ACI 318–19 and the experimental results in the target dataset of each case. As can be seen in Fig. 20, it is noteworthy that the calculated values of RC columns in *Case 3* tend to have larger differences than those values from the other cases. Considering the perspective of precise estimations of the lateral strength of RC columns, the TL models considered in this study outperform the latest design standards.

6. Conclusions

This study has presented the feasibility of knowledge transfer techniques in a real-world structural engineering dataset and in three meaningful ways; across section type, shear reinforcement area, and compressive strength. Three different TL strategies are considered to more accurately predict the lateral capacity of the RC columns. This study uses a database consisting of 497 rectangular and circular RC column experiment specimens. The 12 explanatory variables are extracted and selected among the 30 explanatory features in the original dataset by calculating the feature importance scores. Ten repeated experiments are carried out with a randomly selected dataset to properly detect the generalized knowledge transfer performance. According to

the different source and target domains, three different experiments are designed to directly compare their performance across the different structural engineering domains. The discrepancy between the source and target domains is also calculated with several statistical distance metrics. The prediction performances of each TL technique are computed and compared with the results of the more traditional ML model. The following conclusions can be drawn:

- In all cases conducted in this study, the knowledge transfer techniques show better prediction performance, especially for the low target domain availability. The lower the percentage of target domain availability, the more significant the improvement between the baseline and TL model. Such trends can be observed in every TL technique used in this study. Therefore, we can conclude that transferring the pre-trained knowledge from the source domain enables a model to better explain the response variable in the target domain.
- The improvement of performance is particularly emphasized when the available data for the target domain is small. This indicates that a good ML model can be obtained by adopting a knowledge transfer technique, even with a small number of training samples. Therefore, TL can be successfully applied to the field of structural engineering as a means of alleviating the data scarcity problem.
- It is also demonstrated that DW-SVTR shows a good ability to transfer knowledge learned from the source domain, compared to IW-KRR and Two-stage TrAdaBoost.R2. Furthermore, the capability to accurately transfer the pre-trained knowledge is more associated with the underlying physical relationship between the source and target domains and less associated with the discrepancy between the source and target domain distributions.
- This study quantifies the amount of training data necessary to obtain the same level of prediction performance. In every model considered in this study, the required number of target training samples for a good TL model is less than that for a good ML model. Especially, DW-SVTR has the greatest ability in terms of reducing the number of required training samples.
- All TL models considered in this study show great ability to accurately predict the lateral capacity of RC columns with R^2 values higher than 0.95. From the perspective of precise estimations of the lateral strength of RC columns, the TL models considered in this study outperform the recent design standards.

Further research should be carried out to assess if the results obtained in this research can be extended to a broader range of applications, for example, arbitrary cross-sections, different materials, model-scale to full-scale extrapolations, or various sizes of structures. Additional research should be conducted on how to transfer knowledge more efficiently in structural engineering and how the injection of physics-based information in these approaches could further enhance the predictive performance. Furthermore, a generalized set of guidelines regarding the use of knowledge transfer can be developed based on more results from various datasets and tasks in the field of structural engineering.

Data availability

The datasets and results are accessible through the NSF NHERI DesignSafe-CI portal.

CRediT authorship contribution statement

Hongrak Pak: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Stephanie German Paal:** Conceptualization, Methodology, Writing – review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Science Foundation under Grant CMMI #1944301.

References

[1] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep Learning. Cambridge: MIT press; 2016.
[2] Pan SJ, Yang Q. A Survey on Transfer Learning. IEEE Trans Knowl Data Eng 2010;22(10):1345–59.
[3] Viola P, Jones M. Robust Real-Time Object Detection. Int J Comput Vision 2001;4(34–47):4.
[4] Xiang Y, Mottaghi R, Savarese S., Beyond pascal: a benchmark for 3D object detection in the wild. In: IEEE winter conference on applications of computer vision. 2014. IEEE.
[5] Yang B, Luo W, Urtasun R. Pixor: Real-Time 3d Object Detection from Point Clouds. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018.
[6] Pitts III AF, Dempsen SL, Che VW-Y. Natural Language Processing for a Location-Based Services System. 2008, Google Patents.
[7] Sarikaya R, Hinton GE, Deoras A. Application of Deep Belief Networks for Natural Language Understanding. IEEE/ACM Trans Audio Speech Lang Process 2014;22(4):778–84.
[8] Collins M., Duffy N. Convolution Kernels for Natural Language. In: Advances in neural information processing systems. 2002.
[9] Koopman P, Wagner M. Autonomous vehicle safety: an interdisciplinary challenge. IEEE Intell Transp Syst Mag 2017;9(1):90–6.
[10] Li Q, Zheng N, Cheng H. Springrobot: a prototype autonomous vehicle and its algorithms for lane detection. IEEE Trans Intell Transp Syst 2004;5(4):300–8.
[11] Kuderer M, Gulati S, Burgard W. Learning driving styles for autonomous vehicles from demonstration. In 2015 IEEE International Conference on Robotics and Automation (ICRA). 2015. IEEE.
[12] Erickson BJ, Korfiatis P, Akkus Z, Kline TL. Machine Learning for Medical Imaging. Radiographics 2017;37(2):505–15.
[13] Sanchez-Lengeling B, Aspuru-Guzik A. Inverse Molecular Design Using Machine Learning: Generative Models for Matter Engineering. Science 2018;361(6400):360–5.
[14] Wei J, Chu X, Sun XY, Xu K, Deng HX, Chen J, et al. Machine Learning in Materials Science. InfoMat 2019;1(3):338–58.
[15] Luo H, Paal SG. Machine learning-based backbone curve model of reinforced concrete columns subjected to cyclic loading reversals. J Comput Civil Eng 2018;32(5):04018042.
[16] Gao Y, Mosalam KM. Deep transfer learning for image-based structural damage recognition. Comput-Aided Civ Infrastruct Eng 2018;33(9):748–68.
[17] Mangalathu S, Jeon J-S. Classification of failure mode and prediction of shear strength for reinforced concrete beam-column joints using machine learning techniques. Eng Struct 2018;160:85–94.
[18] Pal M, Deswal S. Support vector regression based shear strength modelling of deep beams. Comput Struct 2011;89(13):1430–9.
[19] Siam A, Ezzeldin M, El-Dakhakhni W. Machine learning algorithms for structural performance classifications and predictions: application to reinforced masonry shear walls. Structures 2019;22:252–65.
[20] Huang J, Gretton A, Borgwardt K, Schölkopf B, Smola A. Correcting sample selection bias by unlabeled data. Adv Neural Information Process Syst 2006;19:601–8.
[21] Jiang J, Zhai C. Instance Weighting for Domain Adaptation in Nlp. In: Proceedings of the 45th annual meeting of the association of computational linguistics. 2007.
[22] Garcke J, Vanck T. Importance weighted inductive transfer learning for regression. 2014. Berlin, Heidelberg: Springer Berlin Heidelberg.
[23] Liao X, Xue Y, Carin L. Logistic Regression with an Auxiliary Data Source. In: Proceedings of the 22nd international conference on Machine learning. 2005, Association for Computing Machinery: Bonn, Germany. p. 505–512.
[24] Zadrozny B. Learning and Evaluating Classifiers under Sample Selection Bias. in Proceedings of the twenty-first international conference on Machine learning. 2004, Association for Computing Machinery: Banff, Alberta, Canada. p. 114.
[25] Argyriou A, Evgeniou T, Pontil M. Multi-Task Feature Learning. Advances in neural information processing systems 2006;19.
[26] Lee, S.-I., Chatalbashev, V., Vickrey, D. and Koller, D. Learning a Meta-Level Prior for Feature Relevance from Multiple Related Tasks. In: Proceedings of the 24th international conference on Machine learning. 2007.
[27] Pan SJ, Tsang IW, Kwok JT, Yang Q. Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 2010;22(2):199–210.
[28] Bonilla EV, Chai K, Williams C. Multi-Task gaussian process prediction. Adv Neural Information Processing Syst 2007;20:153–60.

[29] Li F-F, Fergus R, Perona P. One-shot learning of object categories. IEEE Trans Pattern Anal Mach Intell 2006;28(4):594–611.
[30] Lawrence ND, Platt JC. Learning to Learn with the Informative Vector Machine. In: Proceedings of the twenty-first international conference *on Machine learning*. 2004, Association for Computing Machinery: Banff, Alberta, Canada. p. 65.
[31] Schwaighofer A, Tresp V, Yu K. Learning gaussian process kernels via hierarchical bayes. Adv Neural Information Process Syst 2004;17:1209–16.
[32] Quattoni A, Collins M, Darrell T. Transfer Learning for Image Classification with Sparse Prototype Representations. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. 2008.
[33] Raina R, Battle A, Lee H, Packer B, Ng AY. Self-Taught Learning: Transfer Learning from Unlabeled Data. In: Proceedings of the 24th international conference on Machine learning. 2007, Association for Computing Machinery: Corvalis, Oregon, USA. p. 759–766.
[34] Azimi M, Pekcan G. Structural health monitoring using extremely compressed data through deep learning. Comput-Aided Civ Infrastruct Eng 2020;35(6):597–614.
[35] Goswami S, Anitescu C, Chakraborty S, Rabczuk T. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theor Appl Fract Mech 2020;106:102447.
[36] Luo H, Paal SG. Reducing the effect of sample bias for small data sets with double-weighted support vector transfer regression. Comput-Aided Civ Infrastruct Eng 2021;36(3):248–63.
[37] Feng C, Zhang H, Wang S, Li Y, Wang H, Yan F. Structural damage detection using deep convolutional neural network and transfer learning. KSCE J Civ Eng 2019;23 (10):4493–502.
[38] Żarski M, Wójcik B, Miszczak JA. Transfer Learning for Leveraging Computer Vision in Infrastructure Maintenance. arXiv preprint arXiv:2004.12337, 2020.
[39] Kim H, Kim H, Hong YW, Byun H. Detecting Construction Equipment Using a Region-Based Fully Convolutional Network and Transfer Learning. J Comput Civil Eng 2018;32(2):04017082.
[40] Li S, Zhao X, Zhou G. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. Comput-Aided Civ Infrastruct Eng 2019;34(7):616–34.
[41] Azadi Kakavand MR, Sezen H, Taciroglu E. Data-driven models for predicting the shear strength of rectangular and circular reinforced concrete columns. J Struct Eng 2021;147(1):04020301.
[42] Setzler EJ, Sezen H. Model for the lateral behavior of reinforced concrete columns including shear deformations. Earthquake Spectra 2008;24(2):493–511.
[43] Sezen H, Moehle JP. Shear strength model for lightly reinforced concrete columns. J Struct Eng 2004;130(11):1692–703.
[44] Priestley MJN, Verma R, Xiao Y. Seismic shear strength of reinforced concrete columns. J Struct Eng 1994;120(8):2310–29.
[45] Aboutaha RS, Machado RI. Seismic resistance of steel-tubed high-strength reinforced-concrete columns. J Struct Eng 1999;125(5):485–94.
[46] ACI, Building Code Requirements for Structural Concrete (Aci 318-19): An Aci Standard; Commentary on Building Code Requirements for Structural Concrete (Aci 318r-19). 2020, American Concrete Institute.
[47] Bayrak O, Sheikh SA. High-strength concrete columns under simulated earthquake loading. Struct J 1997;94(6):708–22.
[48] Hwang S-K, Yun H-D. Effects of transverse reinforcement on flexural behaviour of high-strength concrete columns. Eng Struct 2004;26(1):1–12.
[49] Jin C, Pan Z, Meng S, Qiao Z. Seismic behavior of shear-critical reinforced high-strength concrete columns. J Struct Eng 2015;141(8):04014198.
[50] Pardoe D, Stone P. Boosting for Regression Transfer. In: ICML. 2010.
[51] Dai W, Chen Y, Xue G-R, Yang Q, Yu Y. Translated learning: transfer learning across different feature spaces. Adv Neural Information Process Syst 2008;21: 353–60.
[52] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 1997;55(1):119–39.
[53] Drucker H. Improving regressors using boosting techniques. In: ICML. 1997.
[54] Ghannoum W, Sivaramakrishnan B, Pujol S, Catlin AC, Fernando S, Yoosuf N, et al. Nees: Aci 369 Rectangular Column Database 2015.
[55] Ghannoum W, Sivaramakrishnan B, Pujol S, Catlin AC, Wang Y, Yoosuf N, et al. Nees: Aci 369 Circular Column Database 2015.
[56] Bergstra J, Yamins D, Cox D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In: International conference on machine learning. 2013. PMLR.