Prediction of Concrete Compressive Strength Based on Combined Models

Luying Shu

December 2024

1 Abstract

Predicting the compressive strength of concrete is inherently challenging due to its complex composition, involving eight key ingredients. This study firstly compares linear techniques such as Linear Regression, Lasso, and Ridge; nonlinear methods like KNN; ensemble approaches including Random Forest, AdaBoost, and XGBoost; and neural networks represented by ANN. The top-performing models—Random Forest, XGBoost, and ANN—were selected and integrated using advanced combination strategies, including simple weighed, weighted averaging, optimized combinations and stacking techniques. The weighted average forecasting approach, initially limited by a single training split, was further enhanced with cross-validation(CV) to optimize model weights and improve robustness. The best-performing combined method was identified as Weighted CV, with model contribution ranked as Random Forest < XGBoost < ANN. This approach achieved a final R^2 value of 0.9596, demonstrating the potential of combining diverse models to enhance predictive accuracy and robustness.

Keywords: Concrete Compressive Strength, Machine Learning, Random Forest, XGBoost, ANN, Combined Model

2 Introduction

Concrete, a composite material consisting of aggregates bonded with fluid cement that hardens over time, is the most widely used material in civil engineering[1]. Its ability to withstand strong compressive loads without deformation makes it popular and reliable in building construction. As a result, compressive strength serves as a crucial property for optimizing concrete mixture designs and structural integrity[2]. However, predicting concrete compressive strength is a challenging task due to its nonlinear dependence on curing age and up to eight key ingredients. Usually, it requires repetitive laboratory testings with a compression machine to measure the value of compressive strength as it is difficult to capture the complex, interdependent relationships among these factors.

2.1 Problem of Interest and Significance

The problem our team tries to solve is finding the optimal predictive model for estimating concrete compressive strength based on available predictors. This would be significantly helpful, as in modern constructions, an accurate prediction model of compressive strength has many significant practical applications. Firstly, it allows for customized solutions tailored to specific project requirements, such as apartments, sculptures, or dams, ensuring optimal performance under varying conditions like atmospheric exposure and regional material availability. Secondly, an accurate predictive model plays a vital role in promoting environmental sustainability[3]. It fully utilizes resources and stimulates the development of low-carbon concrete formulations with eco-friendly alternatives to avoid cement overuse. Thirdly, it enhances cost efficiency by the most balanced mix proportions, reducing the reliance on costly and time-consuming physical tests, and prevents unnecessary material wastage. Lastly, it facilitates quality control through early decision-making, which allows teams to focus on problem prevention rather than correction. Overall, accurate predictive models address critical challenges in construction, contributing to diversified, environmentally sustainable, cost-saving and more efficient industrialized production.

2.2 Dataset description and Preprocessing

2.2.1 Data Description

We obtained a multivariate dataset of concrete compressive strength from the UC Irvine Machine Learning Repository[4]. It contains 1030 rows of concrete instances with no missing values and 9 column features, corresponding to 8 predictors and 1 response variable. The variable information is given in Table 1:

Table 1: Variable Table

Variable Name	Type	Units
Cement	Continuous	kg/m^3
Blast Furnace Slag	Integer	kg/m^3
Fly Ash	Continuous	kg/m^3
Water	Continuous	kg/m^3
Superplasticizer	Continuous	kg/m^3
Coarse Aggregate	Continuous	kg/m^3
Fine Aggregate	Continuous	kg/m^3
Age	Integer	day
Concrete compressive strength	Continuous	MPa

To begin with, we conducted basic descriptive analysis to understand the distribution of our dataset. Following that, we explored various data visualizations, including: Pairwise Scatter Plot, Histogram, Box Plot, Violin Plot, and Correlation Coefficient Heatmap.These visualizations helped us gain a clearer picture of the interactions between the dependent and independent variables, detecting outliers and linear correlations.

2.2.2 Data Preprocessing

After data exploration, we began preprocessing the data and performing linear modeling tests. Firstly, we randomly divided the dataset into two parts: 80% as the training set and 20% as the testing set. Secondly, we standardized the dataset into Z-scores to simplify hypothesis testing. Eventually, we are ready for predictions using some fundamental models.

2.3 Structure for Models of Prediction

In order to predict compressive strength in a logical sequence, we separate our model testing processes into two parts: In order to predict compressive strength in a logical sequence, we separate our model testing processes into two parts:

- Fundamental Model Forecasting
- Combined Forecasting

Specific methods we used are displayed in Figure 1:

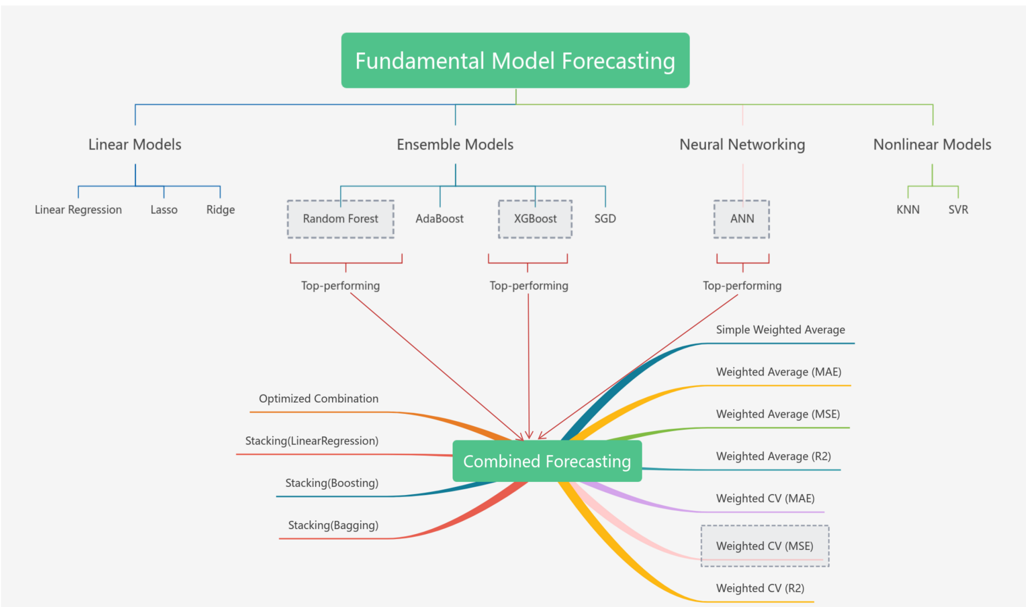

Figure 1: Modeling Structure

3 Fundamental Models for Machine Learning

To predict concrete compressive strength, we employed a variety of models and analytical tools, encompassing three linear models, two nonlinear models, four ensemble methods, and one neural network. The combination of these 10 models allowed us to compare their relative performance and determine several approaches with the most effective prediction for further combined model testing.

3.1 Linear Models

Before starting to build simple linear models, we need to verify linearity through the following tests[5]:

- **Heteroscedasticity Test:** verifies constant variance of residuals.
- **Normality Test:** checks whether residuals follow a normal distribution.
- **Autocorrelation Test:** examines independence of residuals.

To pass all three linear tests, we updated the dataset by taking the logarithm of y(response variable) and applying polynomial transformations to x(predictors). After these adjustments, the results showed as follows:

- White Test p-value: $0.334 > 0.05$, so passed.
- Durbin-Watson Test Statistic: 1.947 close to 2, so passed.
- Coefficient for β_1: $0.451 > 0.05$, so passed.

Under transformation, the dataset met all preconditions for simple linear regression model(2.1.1).

3.1.1 Linear Regression

Linear Regression models the relationship between input features X and the target variable y using a straight-line equation[6]:

$$y = w \cdot X + b \tag{1}$$

The model minimizes the Mean Squared Error(MSE) loss:

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

This model offers a simple and interpretable relationship between dependent and independent variables.

3.1.2 Lasso Regression

Lasso Regression[6] introduces an L1 regularization term to promote sparsity:

$$\text{Loss} = \text{MSE} + \lambda \sum_{j=1}^{p} |w_j| \tag{3}$$

Here, λ is the regularization parameter that controls shrinkage. Lasso regression eliminates irrelevant or less important features by driving their coefficients to zero.

3.1.3 Ridge Regression

Ridge Regression[6] extends linear regression by adding an L2 regularization term:

$$\text{Loss} = \text{MSE} + \lambda \sum_{j=1}^{p} w_j^2 \tag{4}$$

Unlike Lasso, Ridge regression penalizes the squared values of coefficients, making them closer to zero but not exactly zero. Ridge helps handle multicollinearity while keeping all features in the model.

3.2 Nonlinear Models

3.2.1 K-Nearest Neighbors(KNN)

KNN[7] predicts the target value based on the average of the k-nearest neighbors in the feature space. Using the Euclidean distance to identify neighbors, the predicted value is calculated as:

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{5}$$

This model is effective for classification and regression tasks, especially for datasets sensitive to noise.

3.2.2 Support Vector Regression(SVR)

SVR[8] fits a hyperplane to the data while minimizing errors within a margin of tolerance ϵ. The objective function includes regularization:

$$\text{Minimize} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{6}$$

Here, $||w||^2$ controls the smoothness of the regression curve, and ξ_i, ξ_i^* are slack variables for points outside the margin. C controls the trade-off between model complexity and margin violations.

3.3 Ensemble Models

3.3.1 Random Forest Regression

Random Forest[9] builds multiple decision trees on random subsets of data using bootstrap sampling(Bagging). The final prediction is the average of the predictions from T trees:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x) \tag{7}$$

Random sampling of data and features reduces overfitting and variance.

3.3.2 AdaBoost Regression

AdaBoost[10] combines weak learners(e.g., decision stumps) to form a strong predictor. The final prediction is a weighted sum of weak learners:

$$\hat{y} = \sum_{t=1}^{T} \alpha_t \cdot f_t(x) \tag{8}$$

The weights α_t are calculated as:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{9}$$

where ϵ_t is the error rate of the t-th weak learner. Lower error results in higher weights.

3.3.3 XGBoost Regression

XGBoost[10] improves Gradient Boosting Decision Tree(GBDT) with regularization and optimization techniques. Starting with an initial prediction=0.5, the model updates predictions as:

$$\hat{y}_i = \hat{y}_i^{(0)} + \eta\sum_{k=1}^{K} f_k(X_i, \theta_k) \tag{10}$$

where η=0.03 is the learning rate, f_k represents the k-th tree, and θ_k are tree parameters.

3.3.4 Stochastic Gradient Descent(SGD) Regression

SGD[11] optimizes linear regression parameters by minimizing the loss function iteratively. The weights are updated using:

$$w = w - \eta\nabla L(w) \tag{11}$$

Here, η is the learning rate, and $\nabla L(w)$ is the gradient of the loss function.

3.4 Artificial Neural Network(ANN)

To train the neural network, the model enhances its performance by continuously updating connection weights. This is achieved by comparing the predicted output for each input pattern with the corresponding target output, calculating the resulting error, and propagating the error function backward through the network to iteratively refine the weights across layers[12].Once the training process is complete, the network operates by performing inference based on input parameter values. During this stage, the network determines the output of its nodes using the weights and thresholds optimized during training.

In this study, we implemented an ANN model with three fully connected layers, including two hidden layers, each containing 8 neurons. The model was trained for 100 iterations using the ReLU activation function, the Adam optimizer, and a learning rate of 0.01. This configuration achieved an R^2 value of 0.955, demonstrating robust predictive accuracy. The coefficient R^2 quantifies the extent to which the independent variables account for the variability of the dependent variable, with values nearing 1 indicating stronger predictive capability.

The structure of ANN as shown in Figure 2.

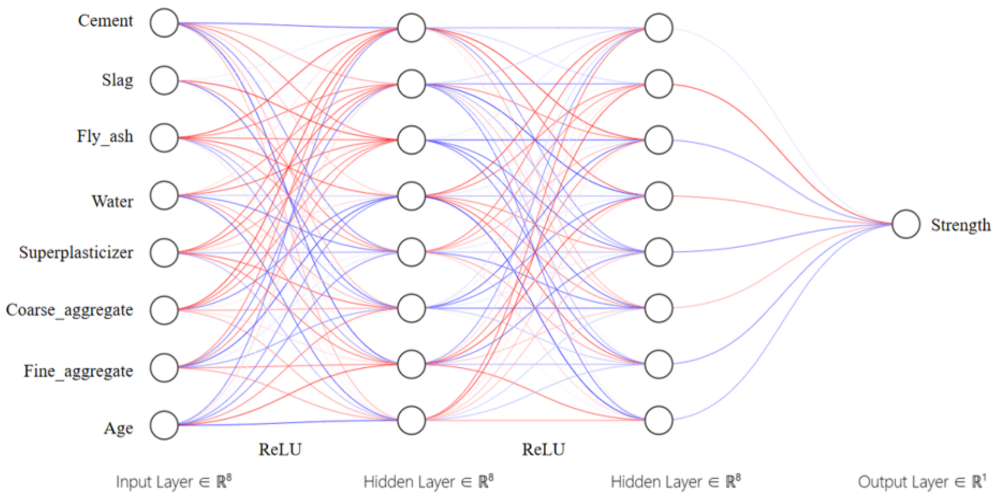

Figure 2: ANN Structure

3.5 Conclusion

We evaluated all 10 machine learning models using three evaluation metrics—MAE, MSE, and R^2[13]. The results are shown in Table 2:

Table 2: Performance of Fundamental Models

Metrics	Linear	Lasso	Ridge	KNN	SVR	RF	AdaBoost	XGB	SGD	ANN
MAE	6.544	9.210	8.211	6.688	8.280	3.221	6.583	2.684	8.356	2.479
MSE	73.826	126.148	103.513	73.168	106.274	21.108	62.578	16.573	106.126	13.338
R^2	0.753	0.578	0.654	0.755	0.644	0.929	0.790	0.944	0.645	0.955

According to the table, Random Forest, XGBoost, and ANN had relatively smaller MAE and MSE, and larger R^2. These models performed significantly better than others, making them suitable for combined forecasting to further improve performance.

4 Combined Forecasting

A total of 5 categories and 11 combined forecasting methods were implemented to improve the precision of predicting the compressive strength of concrete. These methods mainly combine predictions from **RandomForestRegressor**, **XGBRegressor**, and **ANN**, as these models demonstrated superior performance in individual evaluations.

4.1 Forecasting Methods

4.1.1 Simple Weighted Average

Final predictions are obtained by assigning equal or predetermined weights to individual models. We implemented both equal-weighted and user-defined weighted methods.

4.1.2 Weighted Averages Based on Metrics[14]

Weights are determined based on the performance metrics[13]:

- **MAE-based Weighting:** Models with lower Mean Absolute Error(MAE) receive higher weights.
- **MSE-based Weighting:** Models with lower mean square error(MSE) are prioritized.
- **R^2-based Weighting:** Models with higher R^2 scores are given greater importance.

Weights are typically proportional to the inverse of the error: $w_i = \dfrac{\frac{1}{E_i}}{\sum_{j=1}^{n}\left(\frac{1}{E_j}\right)}$.

Here, E_i represents the error term for the i-th model.

Weights can directly use the R^2 value or be normalized: $w_i = \dfrac{R_i^2}{\sum_{j=1}^{n} R_j^2}$.

The R^2 value measures the goodness-of-fit of the model.
Normalize the weights of all models so that the sum of the weights equals 1.

4.1.3 Weighted Cross-Validation(CV)

Using 5-fold cross-validation, weights are optimized for metrics such as MAE, MSE or R^2. For the training set, a predicted y sequence is generated for each fold. The performance is evaluated by calculating the metrics against the actual y sequence.

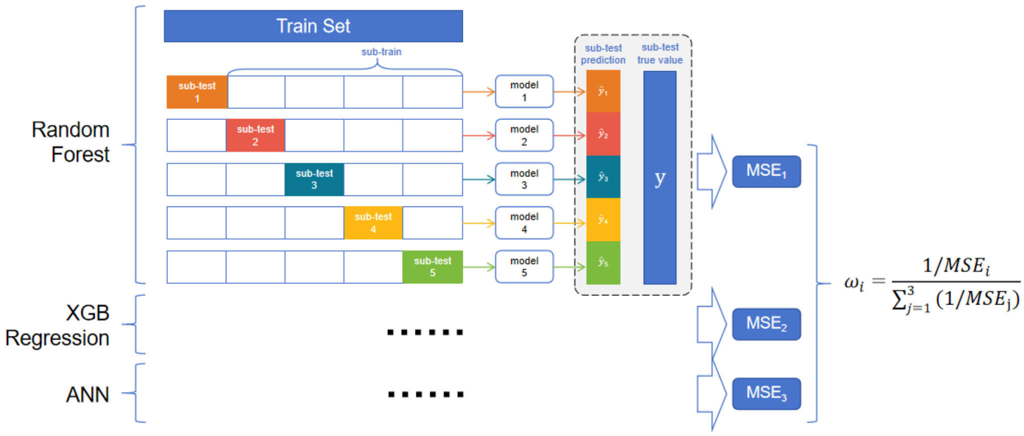

Figure 3: Weighted CV

- **Why We Choose CV:** The improvements brought by cross-validation(CV) have been demonstrated across various fields, including enhancing the robustness of model selection[15], improving evaluation reliability on small datasets[16], and boosting the predictive performance of time series models[17]. These applications show that CV is not merely a performance evaluation tool but also a key component of optimization strategies. Therefore, we aim to incorporate CV-based methods to develop combined forecasting models in the concrete domain.

- **Distinguish from Weighted Average:** The weighted average combination forecasting relies on a single training set split, which may limit its robustness and generalization. To address this, we incorporate 5-fold cross-validation(CV) to optimize the weights by evaluating model performance across multiple folds. In this process, we use the training set 5 times, ensuring that each fold serves as a validation set exactly once. This approach enhances stability and ensures the weights better reflect the models' overall predictive ability on unseen data.

4.1.4 Optimized Combination

An optimization process minimizes the mean squared error(MSE) in the training data set while restricting the weights between 0 and 1.

4.1.5 Stacking

Stacking is an advanced ensemble method that combines predictions from multiple base models using a meta-model(here we use LinearRegression, Boosting and Bagging).

The combined approach significantly outperformed individual models, demonstrating improved accuracy in predicting concrete compressive strength.

5 Conclusion

In this study, we adopted a **combination model approach** to predict concrete compressive strength instead of relying solely on an individual machine learning model. Our methodology focused on combining three robust models—**RandomForestRegressor**, **XGBRegressor**, and **ANN**—which demonstrated the best performance based on MAE, MSE, and R^2. The final results are displayed in Table 3:

Table 3: Summary of Combined Forecasting Results

Method	MAE	MSE	R^2
Simple Weighted Average	2.424	12.509	0.958
Weighted Average(MAE)	2.449	13.334	0.955
Weighted Average(MSE)	2.441	13.253	0.956
Weighted Average(R^2)	2.423	12.508	0.958
Weighted CV(MAE)	2.385	12.168	0.958
Weighted CV(MSE)	2.374	12.073	0.960
Weighted CV(R^2)	2.418	12.459	0.958
Optimized Combination	2.424	12.509	0.958
Stacking(Linear Regression)	2.700	16.729	0.944
Stacking(Boosting)	2.653	16.230	0.944
Stacking(Bagging)	2.706	16.798	0.944

5.1 Results and Best Performing Model

According to Table 3, the combined model based on **Weighted Cross-Validation(MSE)** performs the best. The final weights assigned to the three best models are:

- RandomForestRegressor: $w_1 = 0.281395$
- XGBRegressor: $w_2 = 0.323875$
- ANN: $w_3 = 0.394729$

5.2 Creativity and Achievements

- We adopted a combined model instead of relying on an individual model for concrete compressive strength prediction.
- Few studies in the concrete field have utilized a combination prediction method based on these three specific models(RandomForest, XGB, ANN).
- In contrast to previous studies using simple weighted average or weighted average methods, our CV-based combination prediction demonstrated clear advantages, achieving the lowest MSE and MAE, along with the highest R^2. These results suggest that the CV-based method effectively integrates information across training datasets, yielding more accurate and reliable predictions.

5.3 Limitations and Future Work

While the proposed combination approach offers high accuracy, it is computationally intensive. Future research could explore:

- **Parameter Optimization:** We fine-tuned hyperparameters beyond default settings for all models.
- **Optimizing and Advancing Neural Network Architectures:** Future work can focus on enhancing neural networks by first adding more layers to capture complex patterns, then exploring optimized activation functions such as ReLU or Sigmoid. Building on this, advanced architectures like Convolutional Neural Networks(CNNs) or Recurrent Neural Networks(RNNs) can be explored to further improve performance in predicting concrete compressive strength.
- **Exploring Other Preprocessing Methods:** Alternative preprocessing techniques to handle nonlinearities and extreme outliers. In our project, we applied polynomial and logarithmic transformations exclusively in Linear Regression. These techniques could also be explored in other models.
- **Validating Model Generalizability on Large Datasets:** Real-world testing on larger datasets to validate generalizability.
- **Sensitivity Analysis[18]:** Fixing a set of predictors while varying others to observe the response. This method allows us to understand how different predictors proportionally influence y, which helps in designing concrete mixtures to achieve desired strength.

5.4 Final Remarks

This research demonstrates the potential of ensemble learning in improving the accuracy of concrete compressive strength prediction. By leveraging the strengths of multiple models, our approach advances practical applications in construction engineering, optimizing material usage, enhancing material durability and environmental sustainability like reducing carbon emissions.

References

[1] Sakshi Gupta and Parveen Sihag. Prediction of the compressive strength of concrete using various predictive modeling techniques. *Neural Computing and Applications*, 34(8):6535–6545, April 2022.

[2] Muhammad Zeshan Akber. Improving the experience of machine learning in compressive strength prediction of industrial concrete considering mixing proportions, engineered ratios and atmospheric features. *Construction and Building Materials*, 444:137884, 2024.

[3] Ehsan Mansouri, Maeve Manfredi, and Jong-Wan Hu. Environmentally friendly concrete compressive strength prediction using hybrid machine learning. *Sustainability*, 14(20), 2022.

[4] I-Cheng Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C5PK67.

[5] SS Wu, BZ Li, JG Yang, and SK Shukla. Predictive modeling of high-performance concrete with regression analysis. In *2010 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 1009–1013. IEEE, 2010.

[6] LE Melkumova and S Ya Shatskikh. Comparing ridge and lasso estimators for data analysis. *Procedia engineering*, 201:746–755, 2017.

[7] Min Duan. Innovative compressive strength prediction for recycled aggregate/concrete using k-nearest neighbors and meta-heuristic optimization approaches. *Journal of Engineering and Applied Science*, 71(1):15, 2024.

[8] Prasenjit Saha, Prasenjit Debnath, and Paul Thomas. Prediction of fresh and hardened properties of self-compacting concrete using support vector regression approach. *Neural Computing and Applications*, 32(12):7995–8010, 2020.

[9] Hong Li, Jiajian Lin, Xiaobao Lei, and Tianxia Wei. Compressive strength prediction of basalt fiber reinforced concrete via random forest algorithm. *Materials Today Communications*, 30:103117, 2022.

[10] Yushuai Zhang, Wangjun Ren, Yicun Chen, Yongtao Mi, Jiyong Lei, and Licheng Sun. Predicting the compressive strength of high-performance concrete using an interpretable machine learning model. *Scientific Reports*, 14(1):28346, 2024.

[11] Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent for least squares regression. In *Conference On Learning Theory*, pages 545–604. PMLR, 2018.

[12] I-C Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

[13] K Sudheer Reddy, Niteesha Sharma, T Ashalatha, and B Ravi Raju. An intelligent ensemble architecture to accurately predict housing price for smart cities. In *International Conference on Intelligent Computing for Sustainable Development*, pages 110–122. Springer, 2023.

[14] Md Arif Istiek Neloy, Nazmun Nahar, Mohammad Shahadat Hossain, and Karl Andersson. A weighted average ensemble technique to predict heart disease. In *Proceedings of the Third International Conference on Trends in Computational and Cognitive Engineering: TCCE 2021*, pages 17–29. Springer, 2022.

[15] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[16] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[17] Vitor Cerqueira, Luis Torgo, and Carlos Soares. A case study comparing machine learning with statistical methods for time series forecasting: size matters. *Journal of Intelligent Information Systems*, 59(2):415–433, 2022.

[18] H Christopher Frey and Sumeet R Patil. Identification and review of sensitivity analysis methods. *Risk analysis*, 22(3):553–578, 2002.