Title

Concrete’s Strength Prediction using Machine Learning Method

Author

Ouyang, Boya

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Concrete's Strength Prediction using Machine Learning Method

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Materials Science and Engineering

by

Boya Ouyang

2024

ABSTRACT OF THE DISSERTATION

Concrete's Strength Prediction using Machine Learning Method

by

Boya Ouyang
Doctor of Philosophy in Materials Science and Engineering
University of California, Los Angeles, 2024
Professor Gaurav Sant, Chair

In this study, I present a comprehensive study that addresses the complex challenge of predicting concrete strength, leveraging the power of advanced machine learning techniques. Recognizing the limitations of traditional prediction models, I have introduced innovative methodologies to enhance accuracy and interpretability in this crucial aspect of construction.

Central to my approach is the development of the Ensemble-Based Outlier Detection (EBOD) algorithm. Recognizing the detrimental impact of noisy data on model performance, I designed EBOD to integrate multiple detection algorithms, thereby significantly reducing the bias associated with single-algorithm methods. This innovation ensures that the datasets used for model training and analysis are of the highest quality, laying a solid foundation for more accurate predictive modeling.

Moving forward, I explored the capabilities of Gaussian Process Regression (GPR) in predicting concrete strength. My work with GPR is not just about prediction; it's about understanding the intricacies of the data. I optimized the GPR model to not only forecast concrete strength with remarkable accuracy but also to quantify the uncertainties associated with these predictions. This dual capability of the GPR model enriches the interpretability

of the results, providing deeper insights that are invaluable for material engineering and construction management.

In my pursuit of transparency and interpretability in predictive modeling, I introduced symbolic regression into the study. I recognized the need for models that not only predict but also explain. Symbolic regression offered a solution, enabling me to construct interpretable models that shed light on the underlying physical phenomena governing concrete strength. To enhance the predictive power of these models, I incorporated advanced data augmentation techniques, such as the Synthetic Minority Over-sampling Technique (SMOTE), pushing the boundaries of prediction and understanding in unexplored domains.

A pivotal aspect of my study involved a meticulous analysis of the balance between data volume and the precision of machine learning models. I undertook a comprehensive evaluation of a vast dataset, assessing the performance of various algorithms in predicting concrete strength. This rigorous analysis highlights my commitment to not only advancing the accuracy of predictive models but also to understanding the practical challenges and limitations of employing machine learning in the field of concrete strength prediction.

Through the development of innovative algorithms, the application of advanced machine learning techniques, and a thorough analysis of extensive datasets, I aim to revolutionize the way we predict, understand, and apply concrete strength models in industrial applications, setting new benchmarks for accuracy and interpretability.

The dissertation of Boya Ouyang is approved.

Ali Mosleh

Jaime Marian

Amartya Sankar Banerjee

Gaurav Sant, Committee Chair

University of California, Los Angeles

2024

To my mother . . .

who—among so many other things—

saw to it that I learned to touch-type

while I was still in elementary school

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Professor Bauchy and Professor Sant for their invaluable guidance, support, and mentorship throughout my PhD study. Their extensive knowledge, insightful feedback, and unwavering encouragement have been instrumental in shaping not only this research but also my growth as a scholar.

Professor Bauchy's expertise and innovative approach to complex problems have significantly enriched my understanding and perspective on the subject. His meticulous attention to detail and commitment to academic excellence have been a constant source of inspiration and motivation for me.

Professor Sant's profound understanding of the field and his ability to elucidate intricate concepts have immensely contributed to the depth and quality of my research. His constructive criticism and patient guidance have been crucial in honing my analytical and problem-solving skills.

The journey through my PhD study has been challenging and rewarding, and I could not have navigated it without the steadfast support and mentorship of Professor Bauchy and Professor Sant. I am profoundly grateful for their generosity in sharing their wisdom, for their belief in my potential, and for their unwavering support at every stage of my research journey.

I extend my heartfelt thanks to both professors for their invaluable contributions to my PhD study and for the lasting impact they have made on my academic and professional journey.

VITA

2015	B.S. (Materials Science and Engineering), Central South University.
2017	M.S. (Materials Science and Engineering), Stanford University.
2018–present	Ph.D.(expected), (Materials Science and Engineering), UCLA.

PUBLICATIONS

1. Ouyang, Boya, et al. ”Using machine learning to predict concrete’s strength: Learning from small datasets.” Engineering Research Express 3.1 (2021): 015022..

2. Ouyang, N., et al. ”Predicting concrete’s strength by machine learning: Balance between accuracy and complexity of algorithms.” ACI Materials Journal 117.6 (2020).

3. Ouyang, Boya, et al. ”EBOD: An ensemble-based outlier detection algorithm for noisy datasets.” Knowledge-Based Systems 231 (2021): 107400.

4. Song, Y., et al. ”Decarbonizing concrete with artificial intelligence.” Computational Modelling of Concrete and Concrete Structures. CRC Press, 2022. 168-176.

5. Rongione, Nicolas Augustus, et al. ”High-performance solution-processable flexible SnSe nanosheet films for lower grade waste heat recovery.” Advanced Electronic Materials 5.3 (2019): 1800774.

CHAPTER 1

Introduction

The goals of the proposed project is motivated by the current need to develop new predictive models for concrete strength could accelerate the discovery of new concretes simultaneously offering higher performance, longer service life, lower cost, and lower carbon impact. Thanks to standardized, popular, and straightforward means of measurement, the compressive strength of concrete at 28 days offers a convenient metric of engineering performance that forms a key input in structural design and quality control. In addition, the development of other mechanical properties (e.g., stiffness, flexural/tensile strength, etc.) is correlated to compressive strength. Better predictability of concrete strength offers a means to reduce the extent of overdesign of field-produced concretes. This is a straightforward means of reducing its carbon footprint—since it allows for more efficient use of cement, without sacrificing performance. The overall goal of the project is to better understand the relationship between concrete composition and its strength. Although concrete's strength is largely governed by the water-to-cement ratio (w/c, mass basis), it is also affected by other features, e.g., chemical and mineral admixtures, cement type and quantity, aggregates types and quantity, entrained air, etc. Altogether, the high number of features influencing concrete's strength and the fact that the effects of individual features may be non-linear, competitive, and/or non-additive makes reliable prediction of the strength development in concrete extremely challenging. Although the development of physics- and chemistry-based predictive models would be desirable, in spite of decades of research, there is presently no available, robust, accurate models that can precisely, accurately, and reliably predict concrete's strength. As an alternative route to physics- and chemistry-based models, artificial intelligence and machine learning (ML) offer an attractive option to develop data-driven models by "learning from example" based on existing datasets. Here, we will use several classes of ML algorithms (e.g., polynomial regression, artificial neural network, random forest, and boosted tree) in predicting concrete's strength. Those models will be further used to design optimal concrete mixtures that minimize cost and embodied CO2 impact while satisfying imposed target strength.

CHAPTER 2

Research Challenges and Related Work

2.1 Noisy Datasets and Outlier Handling in Concrete Data

The introduction of machine learning (ML) techniques in the field of concrete strength prediction has been revolutionary, offering substantial improvements over traditional analytic methods, particularly with complex, nonlinear problems. The embrace of ML in recent research highlights its potential to significantly enhance predictive accuracy. However, the performance of ML algorithms is critically dependent on the availability of large and diverse datasets, which is a considerable challenge in the domain of concrete applications.

The limited availability of industrial concrete strength data, especially in the public domain, presents a significant barrier. This lack of data is compounded by the frequent omission of essential details in reported datasets. Parameters like mixing methods, curing temperatures, and types of aggregates—each vital to understanding and predicting concrete strength—are often missing, resulting in incomplete and inconsistent datasets.

The issue is exacerbated by the lack of standardization in data collection methods, leading to discrepancies that can severely distort the predictive models' outputs. Variability in testing protocols and specimen sizes, for example, can lead to marked variations in strength measurements, which are particularly problematic in smaller datasets typical of the engineering materials field.

The scarcity and unreliability of data not only affect the model's accuracy but also limit the broader application of ML in concrete strength prediction. For ML to realize its full potential in this critical area, addressing the issue of data availability and integrity is essential. This involves not only the collection of more comprehensive datasets but also the development of ML algorithms capable of dealing with data imperfections. The goal is to refine the precision of ML models and enable them to contribute meaningfully to advancements in concrete technology and construction practices.

2.2 Limited Availability of Reliable Concrete Strength Data

The advent of machine learning (ML) techniques ushers in a transformative approach to predicting concrete strength, offering significant advantages over traditional methods, particularly for nonlinear problems. Recent studies endorse ML as a formidable tool in this realm, attributing to its superior predictive capabilities. Nonetheless, the efficacy of ML hinges on the availability of extensive datasets to discern intricate input-output relationships, presenting a notable challenge in concrete applications. The scarcity of publicly accessible industrial concrete strength data further complicates this landscape.

Moreover, the integrity of the available data often falls short of the ideal. Critical parameters influencing concrete strength, such as mixing methods, curing temperatures, and aggregate types, are frequently unreported, leading to datasets that are both incomplete and inconsistent. The precision of ML models is fundamentally tied to the quality and consistency of the training data, a condition that concrete strength datasets struggle to meet due to unstandardized measurement techniques and recording discrepancies.

Inconsistencies in testing protocols or variations in specimen sizes can induce significant disparities in strength measurements, with their impacts profoundly magnified in smaller datasets. This is a recurrent scenario in engineering materials applications, where large, comprehensive datasets are a rarity. Consequently, the reliability of ML-driven concrete strength predictions becomes highly contingent on the volume of training data. Addressing these data limitations is pivotal, not just for enhancing model accuracy, but also for harnessing the full potential of ML in concrete strength prediction applications.

2.3 Mapping Data Uncertainty to Concrete Strength Prediction

Conventional machine learning models are often designed to deal with precise and well-defined data. However, when predicting concrete strength, one encounters a landscape rife with uncertainties, which these models are ill-equipped to navigate. The measurement variability is not a statistical anomaly but a recurring issue, rooted in fluctuating raw material qualities, measurement inaccuracies, and procedural inconsistencies such as changes in testing protocols or variations in sample sizes. This issue is further compounded in industrial settings, where concrete production seldom follows a uniform standard, leading to a broad dispersion of strength values and, consequently, a challenge for quality assurance.

Understanding these uncertainties transcends academic exercise; it has tangible, practical implications. It is the linchpin in achieving cost efficiency, enhancing the quality of mix designs, and formulating concrete proportions that are not merely theoretically sound but also practically viable for specific projects. Such an understanding promises a direct impact on the economic and structural integrity of construction endeavors.

Given these considerations, it is imperative to develop a model that doesn't just make predictions but also quantifies the confidence in those predictions. A model adept at capturing the nuanced spectrum of concrete strength uncertainties would serve as a powerful tool, enhancing the predictability and reliability of outcomes. Such advancements would not only bridge the gap between theoretical models and real-world applications but also provide a dependable basis for decision-making in the various spheres of concrete engineering and construction management.

In pursuit of this goal, the challenge lies in constructing a model that is sophisticated enough to interpret the complex, often incomplete data without losing sight of the practical realities it seeks to represent. This endeavor calls for a concerted effort that combines cuttingedge data science with concrete material expertise to create an analytical framework robust enough to withstand the unpredictability of the construction environment.

2.4 Extrapolating Concrete Strength with Unknown Feature Domain

When tasked with predicting the strength of concrete, engineers and data scientists often rely on well-established feature sets—variables that have been historically proven to influence the final strength of the material. These features include, but are not limited to, the water-cement ratio, aggregate size, curing time, and environmental conditions. However, a significant challenge arises when we must predict the strength of concrete using datasets where critical features are unknown or cannot be measured. This scenario is particularly common when dealing with novel materials, proprietary mixtures, or incomplete datasets.

Identifying the Core Challenge The primary challenge in extrapolating concrete strength in domains with unknown features lies in the unpredictability and variability inherent in the material's response to unmeasured influences. Concrete's behavior is notoriously sensitive to a multitude of factors, and the absence of information about any one of these can lead to substantial inaccuracies in strength prediction. Traditional machine learning models are trained on datasets with known feature spaces; when presented with data that falls outside this space, their predictive performance can degrade significantly.

Implications of the Unknown The inability to accurately predict concrete strength has profound implications for both the safety and economics of construction projects. Overestimating strength can lead to structural failures, while underestimating can result in overly conservative designs that waste materials and resources. Furthermore, the construction industry's increasing interest in sustainable and novel materials introduces additional unknowns into the equation, compounding the challenge.

Addressing the Unknown Addressing this challenge requires a multifaceted approach.

Strategies may include developing more sophisticated models capable of handling high degrees of variability and missing data, as well as investing in research to better understand the influence of lesser-known features on concrete strength. Additionally, fostering collaboration between material scientists, data analysts, and field engineers can lead to a more holistic understanding of how unmeasured features might correlate with those that are measured, indirectly informing strength predictions.

In essence, the challenge of extrapolating concrete strength in the face of unknown feature domains is a significant hurdle in the path toward more innovative and adaptive construction practices. Overcoming it will not only require advanced analytical techniques but also a deeper fundamental understanding of the material properties of concrete and the myriad factors that contribute to its strength.

CHAPTER 3

An Ensemble-Based Outlier Detection Algorithm for Noisy Concrete datasets

In this chapter, we introduce an innovative ensemble-based outlier detection [30] algorithm designed to cleanse noisy datasets, which are often plagued by outliers due to operational errors, intrinsic variability, and recording mistakes. Outlier detection is typically a dataset-specific task that lacks a universal solution due to the unique structure of each dataset. Our proposed EBOD method addresses this by integrating multiple outlier detection algorithms into an ensemble, where each algorithm is tasked with identifying the most conspicuous outliers from its perspective. This approach minimizes the detection bias often associated with single-algorithm methods.

We refine the selection of the optimal ensemble through a forward-backward search, ensuring that the combination of detectors is tailored to the structure of the dataset at hand. To validate our method, we apply it to a challenging dataset of concrete strength measurements. The results show that EBOD consistently surpasses individual detectors and their traditional combinations in performance.

Furthermore, we explore the implications of data cleansing on the complexity and performance of machine learning models, using an artificial neural network as a case study. The findings illustrate the profound impact of effective outlier detection on the training process and predictive accuracy of the network. This chapter will delve into the details of our EBOD algorithm, its computational complexity, and its influence on the subsequent machine learning analysis.

This work on EBOD, exemplified by the concrete strength dataset, marks a significant step forward in our ability to prepare data for complex machine learning tasks. It has been partially adapted from our conference paper:

Boya Ouyang et al., ”EBOD: An Ensemble-Based Outlier Detection Algorithm for Noisy Datasets,” presented in Knowledge-Based Systems 231 [31]: 10740.

3.1 Overview and realted works

The recent growth of machine learning approaches is rapidly reshaping our understanding of the unknown . As an alternative route to the long-established ways to develop our cognition based on the progressive accumulation of experience, machine learning algorithms approach a puzzle directly from an ensemble of existing data [2]. As such, machine learning has changed engineering practices and offered practical [36] solutions to problems that, previously, required experience, intuition, or theoretical knowledge . Since machine learning approaches solely rely on the analysis of data, their outcomes are unsurprisingly strongly affected by the size, distribution, and quality of the dataset. In particular, many studies have stressed the importance of data quality for the success of a machine learning analysis.

In that regard, unreliable, inaccurate, or noisy data often hinders the learning efficiency of a model and, in extreme cases, can even mislead the learning process and result in biased predictions . This is a serious issue as datasets for engineering applications are often based on experimental observations and, hence, can exhibit various types of imperfections, e.g., experimental errors, uncertainties regarding the tested system, variability resulting from the effect of missing features, data entry error, etc. Such datapoints are usually referred to as “outliers” [37]. Importantly, if numerous enough, the presence of outliers in a dataset, can negatively impact machine learning models, which, in turn, can reduce the trust of the public, industry, and governmental agencies in machine learning approaches. As such, proper data cleansing is often a prerequisite to any machine learning analysis. However, it should be noted that “extreme datapoints” that are simply far from the distribution of most of the observations are not always detrimental and can even be extremely informative—as they sometimes capture behaviors in regions of the features space that are poorly sampled.

To reduce the impact of outliers on machine learning models, two solutions are commonly adopted: [38] enlarging the dataset to minimize the weight of outliers and [43] identifying outliers and excluding them from the dataset [44]. The first option is often not practical for reasons of time, cost, or unavoidable uncertainties during data collection. Hence, it is of special importance to find efficient ways to detect outliers in datasets. To this end, various outlier detection methods are available. Simple approaches are based on identifying points that are far away from an expected pattern (e.g., Gaussian distribution) [48], further than n standard deviations away from the mean [52], or beyond the interquartile range, as defined with boxplots. More advanced outlier detection algorithms have been developed in the field of data mining. Many studies have focused on data clustering, where isolated clusters are considered as outliers. Alternatively, outlier detection approaches can be based on the analysis of the distance, density, or angle between datapoints. As non-parametric methods, the above detection methods facilitate the data cleansing process as they can identify outliers without assumptions of the data distribution. However, those detection algorithms are sensitive to hyperparameter settings such as number of neighbors for the cluster-based algorithms. Further, their performance can largely depend upon the spatial distribution of the data points, leaving it impractical to apply any single detection algorithm universally for various datasets. For instance it becomes much less advantageous to use those algorithms for detecting outliers in high dimensional space, as the distribution of the datapoints become highly sparse such that it is less common to have locally clustered datapoints. Alternatively, the issues associate with the individual detectors can be substantially addressed by combining a number of individual detector (i.e., base learners) into an ensemble-based detector.

In recent years, a number of ensemble-based outlier detection algorithms have also been proposed to improve the detection accuracy and robustness, especially for noisy datasets where the performance of a single detector tends to be less reliable. However, selecting the right base detectors has always been a tricky task that requires intuition or expert knowledge, due to the fact that outliers are not associated with a universal fingerprint . Further, poor-performing base detectors can substantially weaken the accuracy of the ensemble-based detector . As a result, the outlier detection algorithms usually need to be adjusted, replaced, or recombined dynamically from one dataset to the other. In particular, selecting a given outlier detection algorithm often requires some level of intuition or knowledge on the nature of the dataset—since each outlier detection algorithm comes with its own definition regarding how outliers differ from normal datapoints. In addition, outlier detection algorithms usually rely on the fairly arbitrary choice of a "threshold" value in discriminating outliers from non-outlier datapoints [53]. Selecting the optimal threshold is often a complex choice—as a loose threshold may not properly detect outliers, whereas, in turn, a strict threshold may result in the removal of valuable information from the dataset. For all these reasons, data cleansing is often highly subjective. Unfortunately, over the past years, far more attention has been placed on designing complex regression/classification machine learning algorithms than on developing robust, non-biased outlier detection methods—so that one may argue that outlier detection might be the actual bottleneck of many machine learning applications [54].

Here, as a steppingstone toward this end, we propose an unsupervised, ensemble-based outlier detection [57] approach that automatically determines the optimal the union of different outlier detection algorithms—wherein each outlier detector is used to solely detect the most extreme outliers [64]. Specifically, the based learners for EBOD are individual outlier detection algorithm, and these base learners are combined by taking the union of different outlier detection algorithms. The optimal selection of detectors is determined by forward-backward search. The use of such an ensemble of loose detectors reduces the risk of bias during outlier detection as compared to data cleansing conducted based on a single detector.

To illustrate this approach, we apply EBOD on a series of regression tasks and compare the regression accuracy of a base machine learning model before and after data cleaning. Firstly, we consider the example of a noisy dataset of production concrete strength measurements previously presented in Ref. (Young et al., 2019). This dataset comprises concrete mixing proportions [69] and associated measured strength after 28 days (as output) for more than 10,000 concrete samples. Beyond that, we further applied EBOD on ten benchmark regression datasets and evaluated its cleaning effect based on the regression accuracy of the same artificial neural network (ANN) models. We demonstrate that our EBOD outlier detection method systematically outperforms all detectors in terms of regression accuracy after data cleaning, when used individually or in combination (based on several other prevailing ensemble-based outlier detectors). Based on this new outlier detection method, we also explore how data cleansing affects the complexity, training, and accuracy of the ANN model.

3.2 Feature selection

To illustrate our EBOD outlier detection approach,concrete strength dataset is considered described in Ref. (Ouyang et al., 2020; Young et al., 2019). This dataset comprises a total of 10,264 concrete strength measurements, which are sourced from real concrete production without any pre-cleaning. It should be noted that concrete is by far the most manufactured material in the world and, hence, accurately predicting its strength is critical to ensure the integrity of the built environment (Kim et al., 2004). Concrete takes the form of a mixture of cement, water, sand (fine aggregates), stones (coarse aggregates), supplementary cementitious materials (e.g., fly ash), and chemical additives (Troxell et al., 1968). To the first order, the strength of a given concrete depends on the mixing proportions of these raw ingredients (Domone and Soutsos, 1994). To select the features to be used as inputs for the machine learning models considered herein, we conduct a permutation importance analysis.13 This analysis consists in determining the feature importance by independently randomly shuffling each feature and tracking the associated loss in accuracy (wherein important features results in more significant accuracy loss). Note that this analysis is conducted based on the ANN model presented in Sec. 3.4. Figure 1 shows the outcome of this analysis. As expected, the water-to-cement ratio (w/c) features the highest importance. We then select the features used herein to train the machine learning model based on their importance—while nevertheless limiting the dimensionality of the feature space by disregarding low-importance features. Based on this analysis, we select the following six most influential features controlling concrete's strength, namely (in order of decreasing importance), (i) water-to-cement ratio (w/c, mass basis), (ii) fine aggregate mass fraction, (iii) water-reducing admixture (WRA) dosage, (iv) coarse aggregate mass fraction, (v) fly ash mass fraction, and (vi) air-entraining admixture (AEA) dosage. In contrast, due to their low importance, the following other features are disregarded in this study: concrete load size, ambient temperature, and plant origin (categorial variable). Note that, for normalization purposes, all the relevant features are converted into a weight fraction. The cement mass fraction is not considered herein, as it is redundant with other features (i.e., the sum of all the weight-based features is 100). The 28-day strength of concrete is indicative of its long-term strength and largely dictates its performance (and price).

This concrete dataset exemplifies many difficulties associated with real-world datasets. For example, strength measurements exhibit some intrinsic variability, which makes it hard to discriminate outliers from legitimate measurement variabilities (Zhenchao, 2020). Outliers may also result from data entry typos or experimental errors, such as errors in mixing proportions (e.g., excess of water). Strength measurements can also be affected by external factors that are not captured by the present features (e.g., temperature, relative humidity, raw material quality, mixing protocols, etc.). As such, this dataset offers an ideal, archetypal, and challenging basis to illustrate our EBOD outlier detection approach.

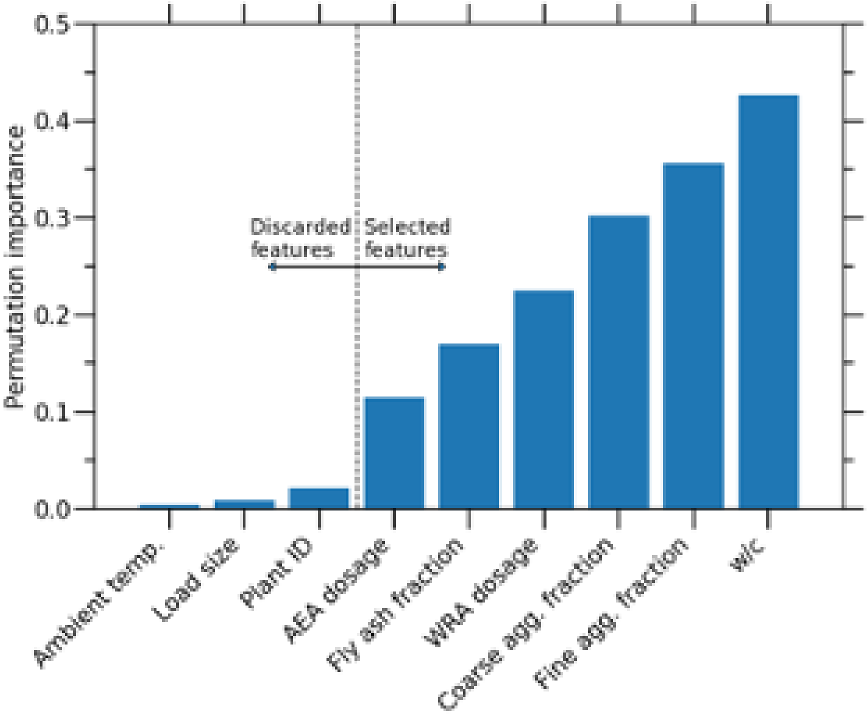

Figure 3.1: Permutation importance of each of the features considered herein, namely (in order of increasing importance), concrete load size, ambient temperature, plant origin (categorial variable), air-entraining admixture (AEA) dosage, fly ash mass fraction, coarse aggregate mass fraction, water-reducing admixture (WRA) dosage, fine aggregate mass fraction, and water-to-cement ratio (w/c, mass basis

3.3 Artificial neural network model

Based on the datasets, we train an ANN regression model aiming to predict concrete strength as a function of the mixture proportions. Although ANN may not offer the highest accuracy for this dataset (Ouyang et al., 2020), we select this regressor based on its sensitivity to outliers (Khamis et al., 2005). The ANN model is implemented and trained by using Scikit-learn (Pedregosa et al., 2011). We adopt resilient backpropagation to optimize the model parameters (Riedmiller and Braun, 1992). For simplicity, we restrict the ANN model to a single hidden layer and use a sigmoid as activation function. During training, the model is iteratively updated using a stochastic gradient descent optimizer until the relative change of the loss (here, the mean-square error) becomes minuscule (¡ 10-4). Once trained, the performance of the ANN model is evaluated based on the root-mean-square error (RMSE) and coefficient of determination (R^2). Here, the RMSE is the averaged Euclidian distance between predicted and measured data and R^2 quantifies their corresponding degree of scattering.

Prior to any training, the concrete strength dataset is randomly divided into a training set (80 percent of the datapoints), which is used to train the model, and a test set (remaining 20 percent of the datapoints), which is kept invisible to the model during its training and, eventually, is used to assess its ability to predict the strength of unknown concrete samples. For optimizing the choice of the model hyper-parameters, we implement five-fold cross-validation within the training set (Stone, 1974). In this study, the only hyperparameter that is considered is the number of neurons in the single hidden layer—wherein a deficit of neurons results in a simple model that is prone to underfitting (high bias), whereas an excess of neurons leads to an unnecessarily complex model that exhibits overfitting (high variance) and poorly generalizes to new samples that are not included in the training set. The optimal number of hidden neurons (and the dependence thereof on the presence of outliers) is determined based on the average cross-validation RMSE (see below).

3.4 Outlier detection algorithms

In this study, an ensemble-based outlier detection (EBOD) method is introduced that is based on an optimized combination of several base detectors. Our approach combines those base detectors and makes more reliable predictions to flag the outliers in various datasets. To this end, we select seven common outlier detection algorithms as the base learners for this ensemble listed on Table 2. This selection is based on the wide acceptance, simplicity of implementation, complementarity, and variety of these algorithms.

For the individual base learners shown in Table 2, the LOF, KNN, and SOS detectors can be classified as belonging to the family of distance-based algorithms, but differ in their approach and mathematical basis for identifying outliers. LOF approaches the problem from the concept of local density (which is estimated by the distance over which a point can be reached by its neighbors) since outliers tend to reside in low-density regions. Outliers are defined as the points that exhibit a density of neighbors that is low enough. Likewise, KNN evaluates the average distance between a central data point and its k nearest neighbors and scores its probability of being an outlier based this distance. The detection offered by SOS is based on the concept of affinity. This algorithm first computes the distance matrix of feature vectors for a datapoint, and then transforms this distance matrix into an affinity matrix. As such, outliers are defined as points showing a low affinity with the other datapoints.

The other algorithms are rooted in alternative viewpoints regarding what differs outliers from normal datapoints. In that regard, ABOD detects the outliers based on the weighted variance of the angles between a datapoint and its neighbors—wherein outliers are defined as datapoints that are far from the majority of the other data points in the hyperspace, with a low variance of the angles. This algorithm is efficient for identifying outliers in high-dimensional space by alleviating the curse of dimensionality (Kriegel et al., 2008). COF identifies outliers based on the degree of connection of a datapoint. IFOREST carries out the detection using a tree-based model, wherein outliers are more likely to be isolated near the root of the tree. Finally, OCSVM relies on a support vector machine to draw the boundary segregating true datapoints from anomalies.

Table 3.1: Individual outlier detection algorithm considered as the base learners for constructing the proposed EBOD outlier detector.

Detection algorithm	**Description**
KNN	k-nearest neighbors
LOF	Local outlier factor
COF	Connectivity-based outlier factor
OCSVM	One-class support vector machine
IFOREST	Isolation forest
ABOD	Angle-based outlier detection
SOS	Stochastic outlier selection

3.5 Alternative ensemble-based detectors

To illustrate the strength of the new EBOD data cleansing method introduced in Section 2.4, We compare it with alternative ensemble-based outlier detection methods listed on Table 3. The characteristics of each alternative ensemble-based outlier detector is briefly summarized as follows. The Averaging algorithm attributes an outlier score to each datapoint based on the average of the scores yielded by each individual detector (Aggarwal, 2013). In contrast, Maximization defines the final score as the maximum of the scores offered by the detectors (Aggarwal and Sathe, 2015). Building on these two ideas, AOM further introduces a bootstrap process, wherein the base individual detectors are first randomly divided into predefined subgroups and the final score is calculated by averaging the maximum scores within each subgroup (Aggarwal and Sathe, 2015). Similarly, MOA defines the final score as the maximum of the average scores within each subgroups (Aggarwal and Sathe, 2015). Feature Bagging combines the outcome of several base outlier detection algorithms by fitting them on random subset of features (Lazarevic and Kumar, 2005). LODA identifies outliers by modeling the probability of observed samples based on a collection of one-dimensional histograms. Each one-dimensional histogram is weak in detecting outlies, but the combination of these weak detectors eventually results in a strong anomaly detector (Pevný, 2016). LSCP is based on the idea that outliers located in distinct regions of the feature space are likely to be properly identified by different individual detectors. As such, this algorithm evaluates the competency of each individual base detector in identifying outliers within a given local region and subsequently combines the top-performing detectors for each region as the final output (Zhao et al., 2019). SUOD initially fits unsupervised base detectors on randomly projected feature space (like Feature bagging). It then evaluates the computational cost of each base model and replace the costly model with a faster supervised regression model, which can increase interpretability and reduce storage costs (Zhao et al., 2020). The last algorithm considered herein, a new outlier ensemble method AKPV (named after the authors of the source paper), combines individual detectors by averaging the scores of three detectors that have best performance (Alexandropoulos et al., 2020). For consistency, the implementation of all the above ensemble-based detectors relies on the same pool of individual base detectors, as introduced in Section 2.3. To ensure a meaningful comparison, we tune the detection parameters used the ensemble-based detectors such that they yield a number of outliers that is identical to that offered by our new EBOD method. In addition to these unsupervised detectors, many supervised ensemble-based detectors that have been developed over the past years, e.g., Bagged Outlier Representation Ensemble (BORE) (Micenková et al., 2015) or Extreme Gradient Boosting Outlier Detection (XGBOD) (Zhao and Hryniewicki, 2018). However, these supervised approaches are not considered herein since, in the case of the present dataset (as well as in many other engineering datasets), the nature of the outliers is not a priori known.

Table 3.2: A summary of the previously proposed ensemble-based outlier ensemble algorithms considered in the benchmark performance tests.

Ensemble-based detection algorithm	**Description**
Averaging	Simple combination by averaging the scores
Maximization	Simple combination by taking the maximum scores
AOM	Average of Maximum
MOA	Maximum of Average
Feature bagging	Combine multiple outlier detection algorithms using different set of features.
LODA	Lightweight On-line Detector of Anomalies
LSCP	Locally Selective Combination of Parallel Outlier Ensembles
SUOD	Large-Scale Unsupervised Heterogeneous Outlier Detection
New outlier ensemble	Average the scores of top three outlier detectors

3.6 Optimal outlier removal based on the union of detectors

As the essential part of our proposed EBOD approach, we implement a forward-backward search approach aiming to pinpoint optimal combination of the base detectors for flagging the outliers that varies across datasets. This method relies on the following steps. First, to avoid any bias regarding the choice of the threshold value to be used for each algorithm, the sensitivity of each outlier detector is tuned so as to systematically flag 10 percent of the datapoints as outliers. This aims to ensure that each detector identifies a small, constant fraction of the datapoints as being abnormal. Second, the performance of each single outlier detector (when used individually) is evaluated by comparing the test set R^2 of the base ANN model (see Section 2.2) before and after removing the detected outliers—wherein a good outlier detector is expected to notably increase the test set R^2. It should be noted that a single detector can either result in an increase in the test set R^2 (if it successfully removes abnormal datapoints) or, potentially, in a decrease in the test set R^2 (if it actually removes useful information, which harms the training of the model). The detectors are then ranked in terms of test set R^2 (i.e., from the best to the worse detector). Third, we conduct the forward-backward search to identify the optimal combination of these outlier detectors, as detailed below.

This general process of the forward-backward search is summarized in Figure 3.2. After determining the cleaning effect of the individual detectors (based on the test set R^2 of the ANN model), we first implement the forward search (Figure 1a), and it covers the following general steps: (i) assess the model accuracy by removing the outliers identified by each of the detectors in the algorithm pool, P, in a one-by-one fashion, (ii) move the best detector to the detector ensemble, U; (ii) remove the union of the outliers as identified by the selected detectors in U; (iii) calculate the model accuracy based on the cleaned dataset; and (iv) repeat the above steps iteratively until the model accuracy does not improve further. The backward search starts once the forward search is done, where the backward search basically mirrors the forward search, namely, we remove one detector at a time by systematically selecting the action that yields the largest increase in the model accuracy (or the lowest decrease). Thus, we track the evolution of the model accuracy during both the forward and backward processes, and the overall optimal detector combination is determined by averaging the two individual optimal detector combinations.

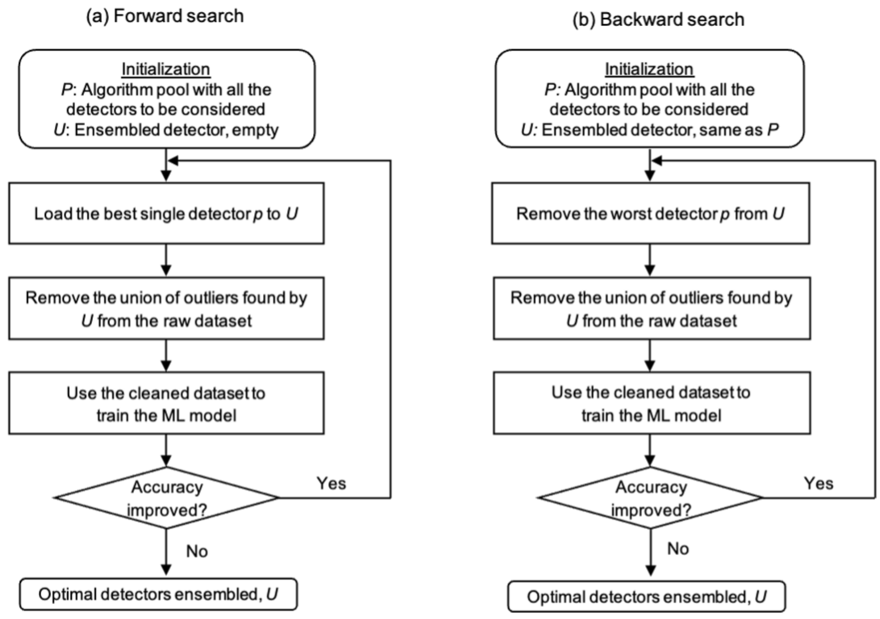
(a) Forward search
Initialization
P: Algorithm pool with all the detectors to be considered
U: Ensembled detector, empty
Load the best single detector p to U
Remove the union of outliers found by U from the raw dataset
Use the cleaned dataset to train the ML model
Accuracy improved?
Yes
No
Optimal detectors ensembled, U
(b) Backward search
Initialization
P: Algorithm pool with all the detectors to be considered
U: Ensembled detector, same as P
Remove the worst detector p from U
Remove the union of outliers found by U from the raw dataset
Use the cleaned dataset to train the ML model
Accuracy improved?
Yes
No
Optimal detectors ensembled, U

Figure 3.2: Flowchart of the (a) forward and (b) backward searching processes for determining the optimal combination of the detectors in our ensemble-based outlier detection (EBOD) method

The effect of combining detectors is assessed, which is at the core of our EBOD approach. To this end, Figure 3.3 shows the evolution of the accuracy of the ANN model during the forward-backward search. Interestingly, we note that the series of detectors that are iteratively selected at each step of the search does not systematically follow their ranking, when used as single detector (see Table 1). This highlights the existence of some combined effects, wherein a given detector may not exhibit notable benefits when used alone, but can positively complement other detectors when used in pairs. Overall, we find that the optimal ensemble of detectors consists in using the union of ABOD, COF, SOS, and LOF. Importantly, both the forward and backward search yield the same optimal ensemble, which confirms the robustness of the present EBOD approach. The optimal ensemble of detectors results in a significant increase in the accuracy of the ANN model, which increases from 0.53 to 0.63 and from 0.49 to 0.60 for the training and test set R^2, respectively.

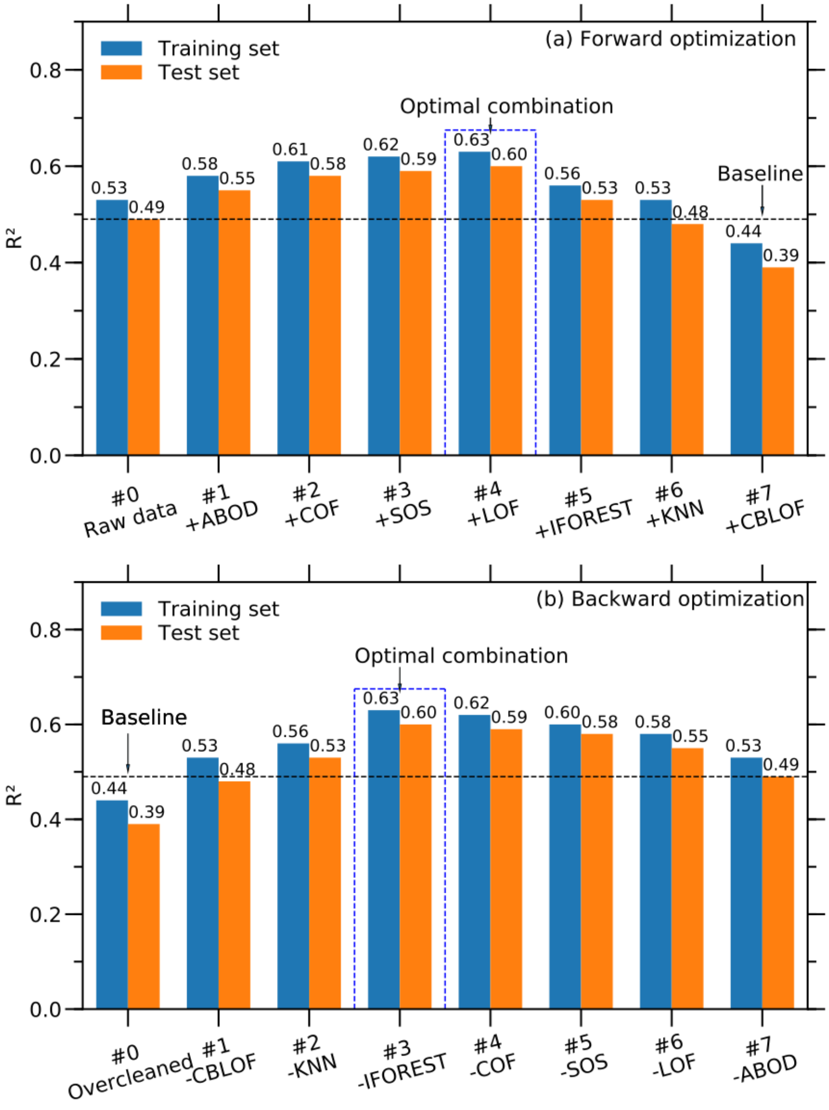

Figure 3.3: Coefficient of determination (R^2) achieved by the artificial neural network considered herein for the training and test sets at each iteration of the (a) forward and (b) backward searching process. Note that, here, we continue the forward and backward search after finding the optimal combination (which is slightly different from the procedures described in Figure 1) for the purpose of discussion

Figure 3.4 illustrates the combined evolution of model accuracy and dataset size (i.e., based on the number of removed outliers) during the forward-backward search. As expected, each iteration of the forward and backward search reduces and increases the size of the dataset, respectively. Nevertheless, we note that the number of outliers that are removed at each iteration is not constant. This illustrates the fact that, during the forward search, the first detector already removes most of the outliers, while each subsequent detector adds its own contribution. The contribution of each detector tends to decrease over time as the dataset gradually runs out of “true” outliers, which manifests itself by a gradual decrease in the number of outliers that are removed at each iteration, as well as a gradual decrease in the associated increment in accuracy. At some point, the search approach leads to excessive removal of outliers and, hence, results in the disappearance of some useful information from the dataset—which, in turn, negatively affects the accuracy of the ANN model. Overall, we find that optimal performance is achieved after removing 2645 data points (i.e., about 25

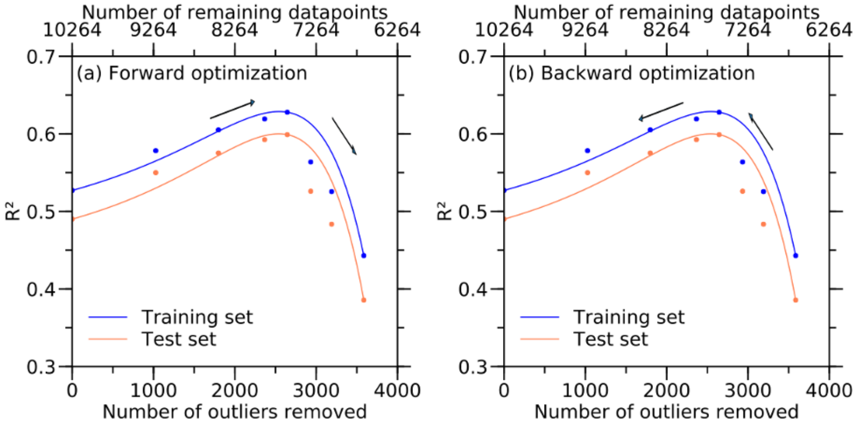

Figure 3.4: Coefficient of determination (R^2) achieved by the artificial neural network considered herein for the training and test sets at each iteration of the (a) forward and (b) backward searching process as a function of the number of removed outliers (bottom axis) and remaining datapoints in the dataset (top axis), wherein the solid line is solely meant to guide the eye

3.7 Influence of data cleansing on model complexity

Having established our EBOD data cleansing approach, we discuss how the removal of outliers affects the optimal degree of complexity of the ANN model. To this end, we conduct a comparative hyperparameter optimization, both before and after data cleansing. This is achieved by five-fold cross-validation, wherein we train a series of ANN models with varying number of hidden layers (the sole hyperparameter considered herein, for simplicity). Figure 4 shows the evolution of the training and validation set RMSE as a function of the number of hidden neurons. As expected, we note that, independently of whether data cleansing is conducted or not, increasing the number of hidden neurons systematically results in a decrease in the training set RMSE. This signals the fact that, as the model becomes more complex, it gradually manages to better interpolate all the details of the training set (Krishnan et al., 2018). Similarly, the validation set RMSE initially decreases upon increasing number of hidden neurons. In this regime, the model is underfitted and exhibits high bias, which is evident from the fact that the RMSE of the training and validation sets are both high and equal to each other. However, in contrast to the RMSE of the training set, the validation set RMSE eventually does not decrease any further and exhibits a plateau. In this regime, the difference between the RMSEs of the training and validation set suggests that the model becomes overfitted. Based on this analysis, we select the optimal number of neurons (a measure of model complexity) as the minimum number of neurons that yields a validation set RMSE that is less than one standard deviation away from the minimum RMSE, wherein the standard deviation is calculated based on the various RMSE values obtained during cross-validation.

Based on this analysis, we assess how data cleansing affects the optimal complexity of the model (see Figures 4a and 4b). We note that the ANN model trained with the cleaned dataset systematically achieves lower RMSE values than its counterpart trained with the non-cleaned dataset—both for the training and validation sets. This indicates that, at an equivalent degree of complexity (i.e., constant number of neurons), the model trained based on cleaned data systematically outperforms the one that is trained on the raw dataset. In addition, we find that the standard deviation of the validation set RMSE (represented by the light blue shaded area in Figure 3.5) is systematically larger when the model is trained based on the uncleaned dataset. This suggests that, based on the folds used for training, the presence of outlier greatly impact the training of the model and its ability to generalize well so as to reliably and consistently predict the validation set. Importantly, we observe that the plateau of the validation set RMSE occurs sooner in the case of the cleaned dataset. In fact, we find that the optimal number of hidden neurons prescribed by the present analysis is 15 and 9 before and after cleansing, respectively. This indicates that data cleansing reduces the optimal degree of complexity of the ANN model. This can be understood from the fact that, when trained from the raw dataset, the model does not properly capture the intrinsic relationship between inputs and output and tends to become more complex than necessary so as to capture some fluctuations in the training set induced by the presence of outliers (see below for more detail on this). In the following, we fix the number of neurons in the hidden layer to these optimal values.

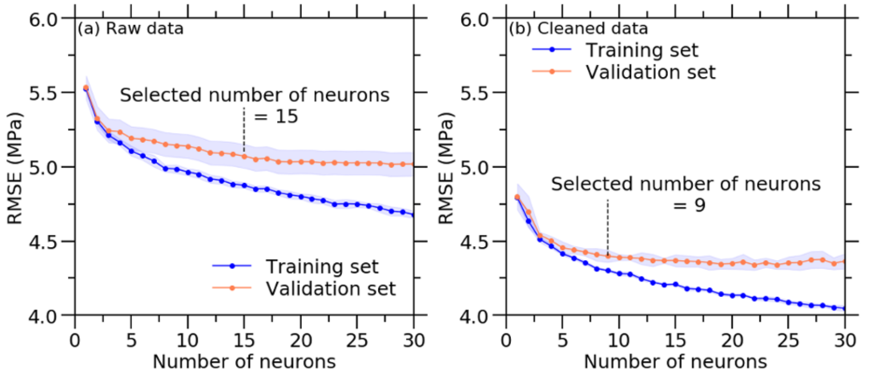

Figure 3.5: Root-mean-square error (RMSE) achieved by the artificial neural network considered herein for the training and test sets as a function of the number of neurons in the hidden layer when trained based on the (a) raw and (b) cleaned dataset

3.8 Influence of data cleansing on learning efficiency

Next, about how the presence of outliers negatively affects the learning process of the ANN model. To this end, we compute the learning curve of the model, both before and after data cleansing. This is achieved by iteratively training the ANN model based on increasing fractions of the training set and subsequently testing its prediction based on the same test set. To enhance the statistical significance of this analysis, the analysis is repeated five times (based on different random training-test splits). The resulting learning curves are shown in Figure 3.6. As expected, the training set RMSE is initially low and then gradually increases with the number of training examples. This is a consequence of the fact that it becomes harder and harder for the model to perfectly interpolate the training set (with a fixed, limited number of adjustable parameters). In contrast, the test set RMSE gradually decreases with the number of training examples—since the model gradually learns how to properly generalize to unknown observations. Irrespectively of whether data cleansing is conducted or not, we note that the training and test set RMSE eventually converge toward a fairly similar value, which confirms that these models do not exhibit any significant degree of overfitting. Note that the maximum size of the training set is smaller in the case of the cleaned data since a given fraction of the datapoints is flagged as outliers and removed.

By comparing Figures 3.5a and 3.5b, we observe that, at fixed number of training examples, the training set RMSE is systematically lower after cleansing. This confirms that, despite remaining fully unsupervised, our EBOD outlier detection indeed removes points that are far away from the interpolated model. This suggests that the points that are removed indeed act as true outliers. Similarly, the test set RMSE is systematically lower after cleansing, which suggests that the removal of the outliers enhances the ability of the model to learn how to properly generalize to unknown observations. Finally, we find that the ANN model converges faster toward its optimal accuracy (i.e., after being exposed to a lower number of training examples) when trained based on the cleaned dataset. This demonstrates that proper data cleansing effectively reduces the number of datapoints that is needed to train the ANN model.

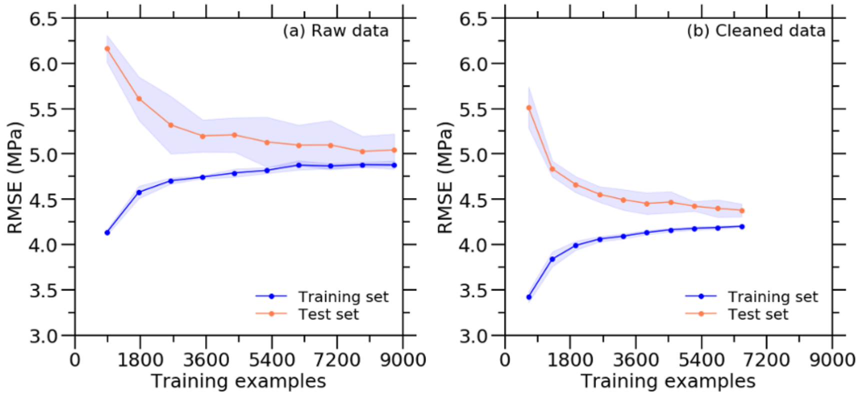

Figure 3.6: Learning curves showing the root-mean-square error (RMSE) achieved by the artificial neural network considered herein for the training and test sets as a function of the number of training examples when trained based on the (a) raw and (b) cleaned dataset

3.9 Influence of data cleansing on model accuracy

We now further discuss how data cleansing affects the final accuracy of the ANN model (i.e., after hyperparameter optimization). Figure 3.6 shows the strength values that are predicted by the ANN model for the test set (i.e., for unknown samples that are invisible to the model during its training) as a function of the actual strength values—wherein the y = x line indicates ideal agreement between predicted and true strength values. In these figures, the color of each pixel indicates the number of overlapped datapoints locally. Overall, we find that the ANN model trained with the cleaned dataset exhibits a higher accuracy—the test set RMSE decreasing from 5.07 to 4.40 MPa. A visual inspection of Figure 3.6 also reveals that data cleansing results in a distribution of the datapoints that is more sharply centered around y = x.

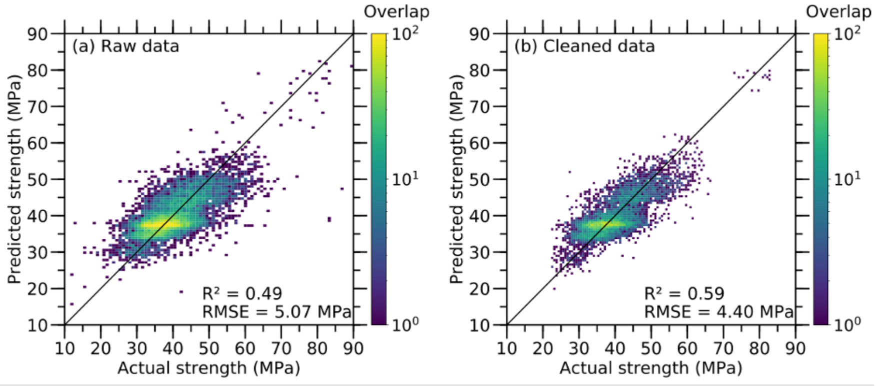

Figure 3.7: Test set concrete strength values predicted by the artificial neural network model as a function of the measured strength, when the model is trained based on the (a) raw and (b) cleaned dataset (note: the color here indicates the number of overlapped data points at each pixel)

To further assess the overall performance of the ANN model, we compute the error distributions (i.e., distribution of the deviation between the prediction and true strength values), which are displayed in Figure 3.7. We find that the ANN model trained on the uncleaned dataset is slightly biased as, overall, it tends to slightly underestimate concrete strength (which manifests itself by a negative mean error). In contrast, the mean error offered by the model trained based on the cleaned dataset is one order of magnitude smaller. Furthermore, we find that the error distribution becomes notably sharper after data cleansing. To quantify this change, we calculate the 90 percent and 95percent confidence intervals based on a Gaussian fit. We find that, after data cleansing, the 90 percent and 95 percent confidence intervals decrease from ± 8.3 to ± 7.2 MPa and from ± 9.9 to ± 8.6 MPa, respectively. This one more time illustrates that the outliers that are identified by our unsupervised EBOD approach are indeed far away from the interpolation model, which confirms their outlier nature.

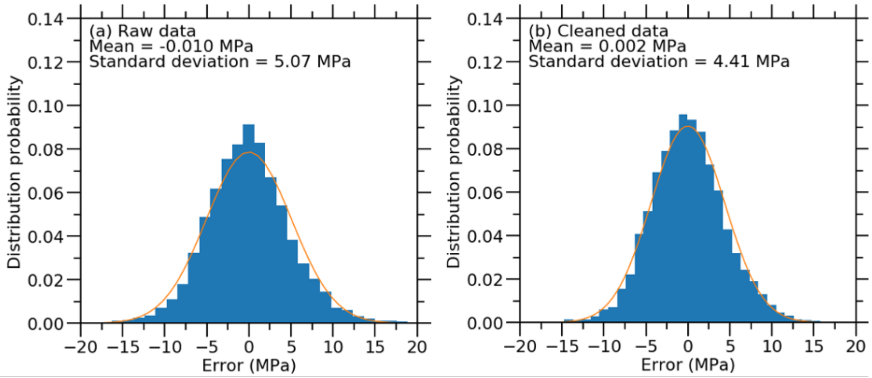

Figure 3.8: Test set error distribution shown by the artificial neural network models trained based on the (a) raw and (b) cleaned dataset (note: the distributions are fitted by some Gaussian distributions))

We further explore whether the presence of outliers "deforms" the trained model. This analysis aims to understand if the outliers that are present in the dataset simply increase the overall error of the model by lying far from the interpolated function or if the error of the model actually arises from the fact that the model itself is affected by the presence of outliers. To this end, since a direct data visualization is not possible in the entire feature space, we focus on the role of two select important features: (i) the water-to-cementitious (w/cm) ratio and (ii) the weight fraction of fly ash. These features are convenient since common concrete engineering knowledge suggests that concrete strength should monotonically decrease upon increasing w/cm and fly ash fraction (Popovics and Ujhelyi, 2008). For illustration purposes, Figure 8 shows the evolution of the strength that is predicted by the ANN models (with and without data cleansing) as a function of these two features. Note that, in this case, the other features are fixed to their average values. In both cases, the predicted values are compared with actual datapoints. Note that these datapoints are not exactly comparable to the predicted strength values as their features are not perfectly equal to the average values used to interrogate the ANN models. Nevertheless, these points are selected based on their vicinity within the feature space (with a relative variation of the features that is less than 10 percent). Among these datapoints, those that are flagged as outliers (and, hence, eventually removed) are highlighted in orange.

In both cases (see Figure 3.8a and 3.8b), we observe that datapoints that are identified as outliers are indeed far away from the rest of the datapoints (in terms of the output value, but not in terms of the input features). Importantly, we find that the presence of outliers significantly deforms the ANN model. First, in both cases, we find that the outliers tend to shift the model toward lower strength values. This echoes the fact the, without any data cleansing, the model exhibits a negative mean error (see Figure 3.8a). Further, we find that the presence of outliers tends to make the model less monotonic and more prone to fluctuations (see Figure 3.8b). This indicates that the model is locally deformed so as to attempt to fit the outliers. This illustration exemplifies why the model that is trained based on the non-cleaned dataset is associated with a higher optimal degree of complexity—since this increased complexity is required to fit the variability of the training set. In both cases, the non-monotonic behavior exhibited by the model trained based on the raw data is not supported by common concrete engineering knowledge and, hence, is solely a spurious effect arising from the outliers.

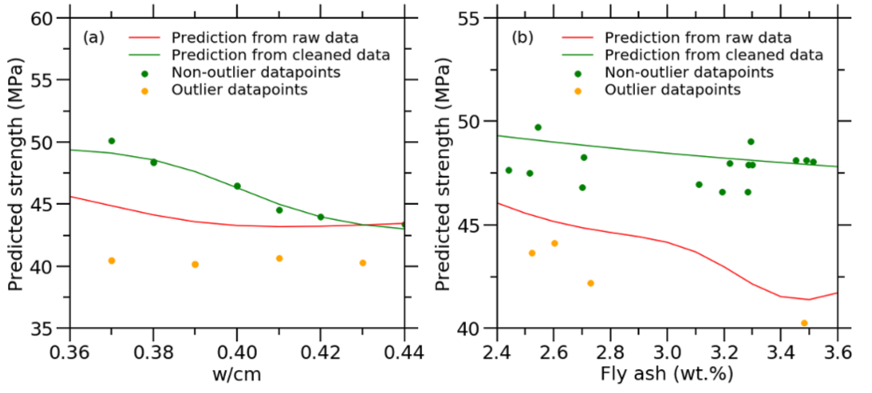

Figure 3.9: Concrete strength predicted by the artificial neural network models trained based on the raw and cleaned dataset as a function of two select features, namely, (a) the water-to-cementitious ratio (w/cm) and (b) the weight fraction of fly ash. Other features are fixed to their average values. The predicted values are compared with actual datapoints that are located at the vicinity of the predicted datapoints in the feature space (see text)

Table 3.3: Summary of the test set performance of the artificial neural network models considered in this study, both before and after using the proposed EBOD outlier removal method

Dataset	**Neurons**	R^2	**RMSE (MPa)**	**Bias (MPa)**	**Confidence interval (MPa)**	
					90%	**95%**
Original	15	0.49	5.07	0.010	± 8.3	± 9.9
Cleaned	9	0.59	4.40	0.002	± 7.2	± 8.6

3.10 Alternative ensemble-based detectors

To illustrate the strength of the new EBOD data cleansing method, We compare it with alternative ensemble-based outlier detection methods listed on Table 3. The characteristics of each alternative ensemble-based outlier detector is briefly summarized as follows. The

Averaging algorithm attributes an outlier score to each datapoint based on the average of the scores yielded by each individual detector . In contrast, Maximization defines the final score as the maximum of the scores offered by the detectors. Building on these two ideas, AOM further introduces a bootstrap process, wherein the base individual detectors are first randomly divided into predefined subgroups and the final score is calculated by averaging the maximum scores within each subgroup . Similarly, MOA defines the final score as the maximum of the average scores within each subgroups. Feature Bagging combines the outcome of several base outlier detection algorithms by fitting them on random subset of features . LODA identifies outliers by modeling the probability of observed samples based on a collection of one-dimensional histograms. Each one-dimensional histogram is weak in detecting outlies, but the combination of these weak detectors eventually results in a strong anomaly detector. LSCP is based on the idea that outliers located in distinct regions of the feature space are likely to be properly identified by different individual detectors. As such, this algorithm evaluates the competency of each individual base detector in identifying outliers within a given local region and subsequently combines the top-performing detectors for each region as the final output. SUOD initially fits unsupervised base detectors on randomly projected feature space (like Feature bagging). It then evaluates the computational cost of each base model and replace the costly model with a faster supervised regression model, which can increase interpretability and reduce storage costs. The last algorithm considered herein, a new outlier ensemble method AKPV (named after the authors of the source paper), combines individual detectors by averaging the scores of three detectors that have best performance . For consistency, the implementation of all the above ensemble-based detectors relies on the same pool of individual base detectors, as introduced in Section 2.3. To ensure a meaningful comparison, we tune the detection parameters used the ensemble-based detectors such that they yield a number of outliers that is identical to that offered by our new EBOD method. In addition to these unsupervised detectors, many supervised ensemble-based detectors that have been developed over the past years, e.g., Bagged Outlier Representation Ensemble

(BORE) or Extreme Gradient Boosting Outlier Detection (XGBOD). However, these supervised approaches are not considered herein since, in the case of the present dataset (as well as in many other engineering datasets), the nature of the outliers is not a priori known. Figure 3 illustrates the combined evolution of model accuracy and dataset size (i.e., based on the number of removed outliers) during the forward-backward search. As expected, each iteration of the forward and backward search reduces and increases the size of the dataset, respectively. Nevertheless, we note that the number of outliers that are removed at each iteration is not constant. This illustrates the fact that, during the forward search, the first detector already removes most of the outliers, while each subsequent detector adds its own contribution. The contribution of each detector tends to decrease over time as the dataset gradually runs out of "true" outliers, which manifests itself by a gradual decrease in the number of outliers that are removed at each iteration, as well as a gradual decrease in the associated increment in accuracy. At some point, the search approach leads to excessive removal of outliers and, hence, results in the disappearance of some useful information from the dataset—which, in turn, negatively affects the accuracy of the ANN model. Overall, we find that optimal performance is achieved after removing 2645 data points (i.e., about 25%) from the dataset.

Table 3.4: Ranking of the algorithms using the Friedman test.

Rank	**Algorithm**
6.69	EBOD
5.73	AKPV
5.25	Feature bagging
3.71	LSCP
2.03	SUOD
1.67	LODA

Table 3.5: Post hoc Dunn's test on EBOD vs. alternative ensemble-based outlier detection algorithms.

Comparison	**p-value**	**Result**
EBOD vs LODA	5.16×10^{-7}	H_0 is rejected
EBOD vs SUOD	3.14×10^{-6}	H_0 is rejected
EBOD vs LSCP	0.002	H_0 is rejected
EBOD vs Feature bagging	0.15	H_0 is accepted
EBOD vs AKPV	0.33	H_0 is accepted

3.11 Non-parametric statistical tests

In addition to investigating the concrete dataset, we further carry out non-parametric tests (also referred as distribution-free tests, which do not assume that the data is normally distributed) on ten benchmark regression datasets (see Table 1) to evaluate the statistical significance of our proposed EBOD versus nine alternative ensemble-based outlier detectors (see Table 3). We first implement Friedman test (Friedman, 1937). Herein, for data with ten groups (i.e., the ten regression datasets) and six treatments (i.e., the six ensemble-based detectors), the Friedman test first ranks the performance of each case, then computes the summed ranking for each treatment. The Friedman test statistics are then used for calculating the p-value. When the p-value is smaller than 0.05, it can be concluded that at least one of the treatments is different from the rest. To get a better understanding of the difference between EBOD of the other methods, Post-hoc Dunn's test (Demšar, 2006) is also performed based upon the mean rank differences observed from the Friedman's test. Dunn's test runs multiple pair-wise-comparison using Z-test statistics, which can be used to obtain p-value for each comparison (Kruskal and Wallis, 1952). If the p-value of an algorithm is smaller than a certain level (typically 0.05), it can be deemed there is significant performance difference between the compared pai

3.12 Conclusions

Overall, we find that our proposed EBOD outlier detection method improves the learning efficiency of the ANN model by decreasing the number of required hidden neurons, as well as the number of datapoints that are needed for the model to learn how to map inputs to output. Importantly, we find that our EBOD outlier detection approach considerably improves the accuracy of the trained ANN model, which is systematically illustrated by the test set R^2, RMSE, bias, and confidence interval. Importantly, our new EBOD method systematically outperforms alternative outlier detector algorithm, used either individually or in ensemble. Altogether, these results suggest that considering an optimized ensemble of outlier detection algorithms (rather than a single detector or simply an average of several detectors) offers a more robust cleansing of the data and, consequently, notably increase the performance of the subsequent machine learning model. Importantly, the EBOD approach does not require any intuition or knowledge regarding which type of detector (e.g., distance-based, angle-based, etc.) is best suited to tackle a given dataset. This approach also has the advantage of being fully unsupervised, that is, it does not require any expert-based examples of outliers or of any preexisting knowledge of what the typical signature of an outlier should be.

CHAPTER 4

Using Machine Learning To Predict Concrete's Strength: Learning From Small Datasets

In this chapter, we delve into the intricate relationship between the proportioning of concrete components and its resulting strength, a topic that has garnered substantial attention over recent decades. Despite extensive research efforts, the establishment of a robust, knowledge-based model that can accurately predict concrete strength remains an elusive goal. Traditional approaches often rely on physical or chemical principles to model this relationship; however, these methods fall short when addressing the complexities and nuanced interactions inherent in concrete formulations. In light of this, we pivot towards a contemporary paradigm—data-driven machine learning techniques—as a novel pathway to unravel this challenge.

Machine learning, with its inherent capacity to decipher complex, non-linear, and non-additive relationships, emerges as a compelling alternative. It promises to provide a fresh perspective on how concrete mixture proportions correlate with strength attributes. Nonetheless, this promising avenue is not without its constraints. A significant hurdle for machine learning models, as highlighted in our research [72], is their dependency on substantial datasets for training. This necessity poses a considerable challenge, given the scarcity of reliable and consistent strength data, particularly when it pertains to concrete used in industrial applications.

In response to these challenges, this chapter presents a comprehensive analysis of an extensive dataset, comprising over 10,000 observations of compressive strengths derived from industrially-produced concrete. We embark on a comparative study of selected machine learning algorithms, examining their proficiency in 'learning' and predicting concrete strength. This exploration is particularly focused on understanding the interplay between the volume of data required for model training and the achievable accuracy of the predictive models [4].

Through this investigation, we aim to shed light on the intricate balance between the precision a model can attain and the volume of data necessary to nurture it. In doing so, we aspire to advance the discourse on the practicalities and limitations of employing machine learning for concrete strength prediction.

This work has been partially adapted from our paper:

Boya Ouyang et al., "Ouyang, Boya, et al. "Using machine learning to predict concrete's strength: Learning from small datasets." Engineering Research Express 3.1 (2021): 015022."

4.1 INTRODUCTION

The 28-day compressive strength is one of the most widely accepted metrics to characterize concrete's performance for engineering applications. Indeed, although this standardized yet simple index is primarily used to evaluate the ultimate strength of concrete mixtures [1], it can also serve as an expedient measure to infer other critical mechanical properties such as elastic modulus, stiffness, or tensile strength [2]. Accurate strength predictions in concrete design have a profound impact on the efficiency and quality of construction projects. Indeed, for instance, an insufficient concrete strength can be the culprit of a catastrophic failure of civil infrastructures. Conversely, concretes exhibiting an overdesigned strength leads not only to higher material expenses [3], but also to additional environmental burdens—such as CO2 emissions in cement production [4]. Over the past decades, a substantial amount of effort has been devoted to developing predictive models for correlating a given concrete mixture proportion to its associated strength performance [5]. Beyond this, an ideal predictive model also provides important insights for designing new concrete with better constructability and durability, and/or at a lower cost [6,7]. Conventional approaches often seek to achieve these goals using physics or chemistry-based relationships [8–10]. Although the role played by major proportioning parameters (e.g., water-to-cementitious ratio, w/c, aggregate fraction, and air void content) has been extensively investigated, the influence of many other factors is not always negligible, e.g., chemical and mineral admixtures or aggregates gradation [11]. Due to the limited understanding of these complex property-strength correlations, it is still extremely challenging to get a robust and universal concrete strength model using conventional approaches [12]. As an alternative pathway, the recent development of ML techniques provides a novel data-driven approach to revisit the strength prediction problem. Importantly, ML-based predictions have been shown to significantly outperform those of conventional approaches, especially when handling non-linear problems [13]. Without the need for any physical or chemical presumptions, this new approach also further permits greater flexibility to extract hidden, non-intuitive feature patterns directly from the input data. As such, recent studies have established ML as a promising approach to predict concrete strength[14–17]. However, a major limitation of ML approaches lies in the fact that a large dataset is usually required for ML algorithms to "learn" the relationship between inputs and outputs [18,19]. This is a major concern for concrete strength applications, as strength data for industrial concretes are often difficult to access (i.e., data is not publicly available). In addition, reported concrete strength data are often incomplete, that is, some important features are often missing, e.g., curing temperature, additives, types of aggregates, etc. More generally, machine learning approaches require accurate and self-consistent data—which is often questionable for concrete strength data due to non-standardized measurements or inconsistencies in data recording [20]. For example, the strength of a given concrete material can significantly vary when the testing protocol or specimen size is changed [21–23]. Although such difficulties can be filtered out with sufficiently large datasets, their significance tends to be exacerbated in the case of small datasets. For all these reasons, it is critical to assess how the reliability of ML approaches for concrete strength prediction applications depends on the number of training data points. This study revolves around two core questions: [62] how much data is sufficient for training a machine learning model and [79] which ML algorithms are better suited to deal with small datasets. Here, by building on our previous studies [17,24], we explore the above questions by taking the example of three archetypal learning algorithms, namely, polynomial regression [40], artificial neural network [6], and random forest [61]. We compare the ultimate learning accuracy of these algorithms (i.e., based on the entire training set), as well as their learning efficiency as a function of data volume. These results are insightful for facilitating the adoption of machine learning techniques for small datasets—as relevant to concrete engineering.

4.2 Machine learning algorithms

To establish our conclusions, we assess the performance of three common, archetypical learning algorithms [59] as a function of the number of training data points. These methods are chosen as they belong to three distinct families of ML models, namely, polynomial, network-based, and tree-based [25,26]. Note that all the hyperparameters of the ML models considered herein were optimized in a previous study so as to achieve an optimal balance between under- and overfitting [16]. First, we consider PR, which is essentially based on linear regression, wherein the model parameters designate an n-degree polynomial function [27]. Based on our previous work [16], the PR model adopted herein features a maximum polynomial degree of 3. Second, we explore the potential ANN, which is a computational structure consisting of an input layer, an output layer, and one or several hidden layers bridging the two formers—wherein each layer comprises a collection of artificial neurons (i.e., computational units) [28]. Based on our previous work [16], the present ANN model exhibits 7 neurons in a single hidden layer. We adopt the sigmoid function as activation function to prioritize the importance of the input data and we use the backpropagation algorithm to optimize the model parameters [29]. Third, we consider RF, which is an enhanced bagging method since, by using the majority-voting concept, this approach is typically more predictive than conventional decision trees [30]. Here, based on our previous work [16], our RF model comprises 16 trees. Despite the different nature of these algorithms, their common goal is to predict a variable y (i.e., the 28-day strength) as a function of the input variables x (i.e., mixing proportions of concrete), while minimizing the difference between measured and predicted strength values (see Ref. [16] for details). 2.2 Feature selection The dataset used in this study includes the 28-day compressive strength of 10,264 commercial concretes and associated mixture proportions [17]. All the mixtures were cast using ASTM C150 compliant Type I/II cement [31] and Class F fly ash compliant with ASTM C618 [32]. The seven most influential features are considered in this study, namely, [56] the water-to-cementitious ratio [90], [83] cement %, [10] fly ash %, [85] fine aggregate %, [65] air-entraining admixture [15] dosage, and [49] water-reducing admixture [87] dosage. For normalization purposes, features (2-to-4) are taken as the solid weight fractions, wherein the fraction of coarse aggregates is excluded as it is redundant (i.e., the sum of all the weight fractions is 100%).

4.3 Model training

Following common practices in machine learning, 70% of the strength observations are randomly selected and used for model training (i.e., “training set”). The remaining 30% of the data are kept hidden to the model and assess the ability of the model to predict the strength of unknown concretes (i.e., “test set”). The hyperparameters of each mode are optimized by five-fold cross-validation [33]. In detail, the training set is randomly split into five smaller folds (each made of 20% of the training data). In each round of analysis, the model is trained based on four folds and validated based on the remaining fold (i.e., “cross-validation set”).

4.4 Accuracy evaluation

We evaluate the accuracy of each model by calculating their mean-square error [32] and coefficient of determination (R^2), wherein the MSE is the averaged Euclidian distance between predicted and true strength data in the test set. The relative MSE [8] is then calculated as the square-root of the MSE. The R^2 factor further quantifies the accuracy of the model predictions in terms of the degree of scattering around the fitted input-output relationship—wherein a perfect prediction would be associated with $R^2 = 1$. We further analyze the deviation between strength predictions and measurements by computing the error distribution—that is, the distribution of the differences between predicted and measured strength values for each concrete mixture in the test set. The error distribution yielded by each model then serves to calculate the 90 and 95% confidence intervals of a predicted strength falling into these ranges (see Ref. [16] for details). 2.5 Evaluation of the learning efficiency To investigate how each model "learns" how to predict concrete strength, we compute their "learning curve," which is often used to quantify the learning progress [34]. This approach consists of plotting the accuracy of the model as it is exposed to an increasing number of training examples. Here, we compute the MSE [51] while gradually increasing the size of the training set by 10% increments. To ensure consistent comparison, all the models are trained and evaluated based on identical training and validation sets.

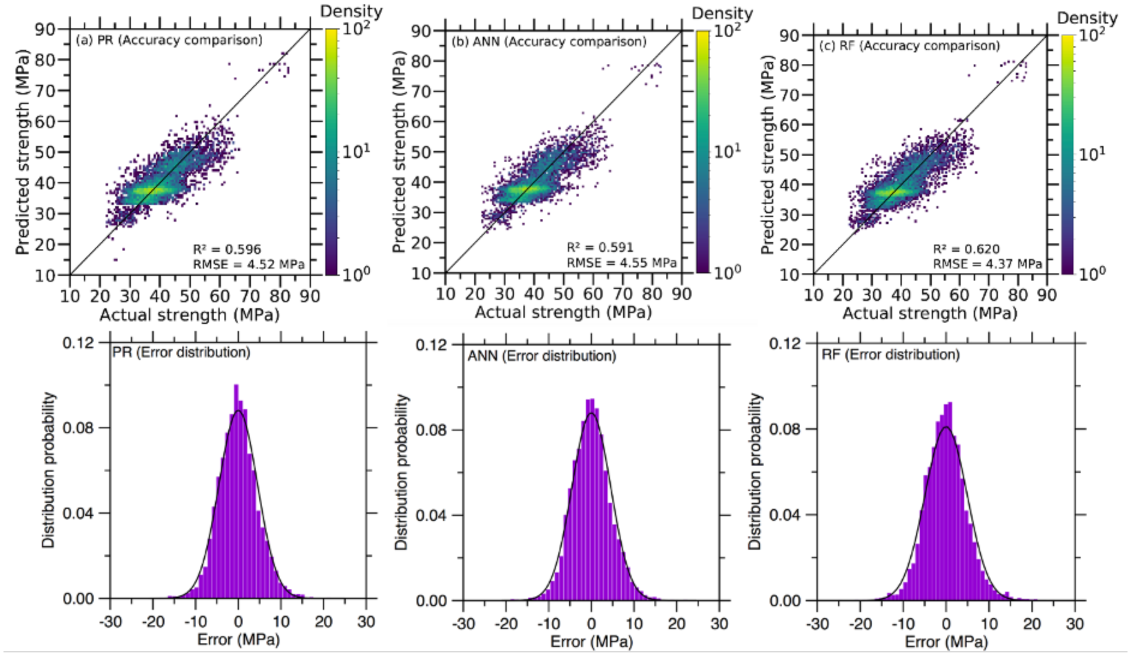

Figure 4.1: Comparison between predicted vs. ground-truth test set strengths [70] and error distribution [22] for the [17] PR, [45] ANN, and [28] RF models. Pixel colors in the left plots indicate the number of overlapped points. The error distributions are fitted by a Gaussian distribution function.

4.5 Accuracy of the machine learning models

We first compare the final accuracy offered by each ML model, that is, when trained based on the entire training set. To this end, Fig. 1 shows the predicted vs. ground-truth test set strength for each model, as well as the associated error distributions. The accuracy analysis is summarized in Tab. 1. In detail, we find that RF features the highest degree of accuracy, which manifests itself by a minimum RMSE, maximum R^2, and minimum confidence intervals.

Model Type	R^2	Confidence Interval (MPa)		Minimum Number of Training Data Points
		90%	95%	
PR	0.596	± 7.43	± 8.86	2680
ANN	0.591	± 7.45	± 8.88	3010
RF	0.620	± 7.22	± 8.60	4070

Table 4.1: Values of R^2 and confidence intervals over the test set for each model (when trained based on the entire training set) and minimum number of training data that is needed for each model to achieve an average validation set MSE that is less than one standard deviation away from its final validation set MSE

4.6 Gradual learning upon increasing training set size

Having shown that RF offers the best final accuracy when trained based on the entire training set, we now focus on the learning curve exhibited by each model—to assess their ability to quickly learn the input-output relationship as they become exposed to a gradually increasing number of training examples (see Fig. 2). As expected, all the models exhibit a fairly similar trend, that is, (i) the MSE of the training set increases with increasing training set size since it becomes increasingly difficult from the model to perfectly interpolate the training set and (ii) the MSE of the cross-validation set decreases with increasing training set size as the model gradually manages to learn the input-output relationship and, hence, eventually shows an increased ability to predict the strength of unknown concretes. Nevertheless, we find that, although the final accuracy offered by the models shows only minor differences (see Tab. 1), their learning curves exhibit more significantly distinct features. In detail, in agreement with the data presented in Tab. 1, we find that RF eventually features the lowest MSE for the validation set, as well as for the training set. However, we note that the MSE of the validation set exhibits a faster decrease in the case of PR and ANN. We further quantify this behavior by computing the minimum number of training data points that is needed for the model to achieve an average validation set MSE that is less than one standard deviation away from its final validation set MSE (i.e., when trained based on the entire training set), wherein the standard deviation is calculated based on the MSE obtained for each validation fold in cross-validation. Overall, we find that PR and, to a lesser extent, ANN features an increased ability to quickly learn how to predict concrete strength from small datasets as compared to RF (see Tab. 1).

4.7 Competition between model accuracy and need for large dataset

Overall, we find that the model offering the highest final degree of accuracy (i.e., RF) requires the largest training set to be trained, whereas, in turn, the models presenting the lowest final accuracy (i.e., PR and ANN) require the smallest training set to be trained. These results suggest the existence of a competition between (i) the final ability of a model to accurately learn the input-output relationship when trained based on an excess of training examples and (ii) the ability of a model to quickly learn this relationship when trained based on a small dataset. This competition can be rationalized in terms of the intrinsic "flexibility" of the model. On the one hand, PR and ANN are constrained, poorly-flexible models—since PR relies on a fixed analytical form, while the present ANN model exhibits a limited ability to capture complex input-output relationships as it comprises a single hidden layer. This lack of flexibility limits the final accuracy that is achievable by these models. Although the degree of complexity of these models (i.e., maximum polynomial degree for PR and number of hidden neurons for ANN) is already tuned to achieved the best balance between under- and overfitting (see Ref. [16]), the fact that the MSE of the training and validation sets both plateau toward the same value suggests that these models are too simple and lack some degrees of freedom. This limitation could potentially be mitigated by carefully increasing the complexity of these models (while avoiding overfitting)—for instance, by increasing the number of hidden layers in ANN [35]. In turn, the constrained nature of these models allows them to quickly achieve their maximum accuracy—since only a limited number of parameters (i.e., polynomial coefficients for PR and neuron-neuron connection weights for ANN) need to be parameterized [36]. This makes it possible for these algorithms to handle small datasets. However, it is clear from Fig. 2 that these models have already achieved their maximum accuracy and, hence, would not benefit from being trained with any additional data. On the other hand, RF is, in contrast, more flexible as it is not constrained by any analytical formulation. Indeed, in contrast to PR (which intrinsically yields a smooth, continuous, and differentiable relationship between inputs and output due to its analytical form), the tree-based structure makes it possible for the RF model to capture rough, less continuous/differentiable functions [37]. This flexibility enables RF to eventually reach a higher final degree accuracy once trained based on the entire training set. In turn, such complexity comes at a cost, namely, a large number of training data points is needed to properly parameterize the RF model. This is well illustrated by the facts that, unlike the cases of PR and ANN, (i) the validation set MSE of the RF model does not reach a plateau and continues to decrease upon increasing training set size and (ii) the final validation set MSE is significantly higher than the final training set MSE. Both of these learning curve features suggests that the RF model has not yet finished its training and, hence, could further by improved if exposed to an increased number of data—that is, unlike the PR and ANN models, the RF model still features some room for improvement

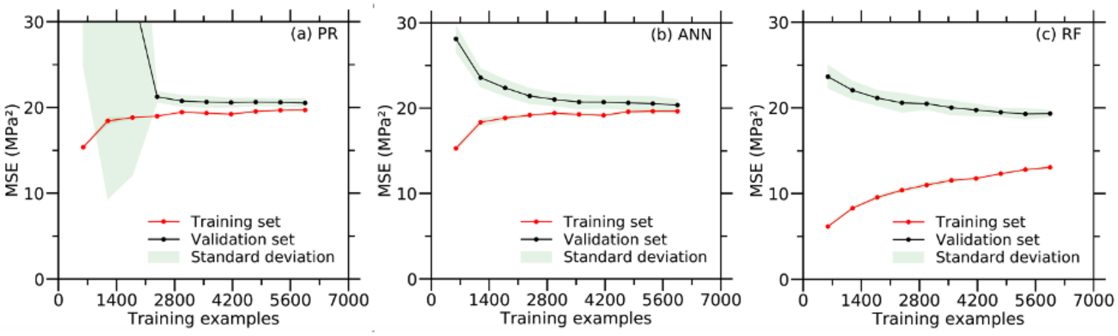

Figure 4.2: CLearning curves showing the MSE of the training and cross-validation sets as a function of the size of the training set for the (a) PR, (b) ANN, and (c) RF models.

4.8 CONCLUSIONS

Machine learning stands as a powerful tool in the realm of material science, particularly for predicting concrete strength—a critical parameter in construction and civil engineering. The application of various machine learning models presents a diverse range of capabilities and requirements, each suited to different scenarios and demands.

Simple models, such as Polynomial Regression (PR), offer a clear advantage in scenarios where the availability of data is limited. These models, due to their inherent simplicity and constrained nature, can rapidly reach their peak performance, making them a suitable choice when time or data is a constraining factor. However, this simplicity comes with a trade-off, as these models often reach a plateau in their predictive capabilities, offering limited final accuracy. This makes them less ideal for applications where precision is paramount, but they remain a valuable tool for initial analysis or when resources are constrained.

On the other hand, more complex and less constrained models, like Random Forest (RF), present a stark contrast. These models thrive on larger datasets, leveraging the increased information to build a more nuanced understanding of the underlying patterns and relationships within the data. This capacity to digest and learn from a substantial volume of data allows these models to achieve higher levels of prediction accuracy. However, this comes at the cost of requiring more substantial computational resources and a more extensive collection of training data. The need for larger datasets can be a limiting factor, especially in scenarios where data collection is challenging or expensive.

The choice between simpler models like PR and more complex models like RF hinges on a balance between the available resources, the urgency of the task, and the required precision of the predictions. For projects where quick, early insights are needed and the available data is sparse, simpler models may be the preferred choice. Conversely, in scenarios where the utmost accuracy is required and sufficient data and computational resources are available, investing in more complex models like RF can yield significant dividends, unlocking higher levels of prediction accuracy and providing more reliable guidance for decision-making processes.

In conclusion, the field of machine learning offers a versatile toolkit for predicting concrete strength, with different models catering to various needs, resources, and objectives. The key lies in carefully assessing the specific requirements and constraints of each project and choosing the model that best aligns with the project's goals, data availability, and resource constraints. Through this tailored approach, machine learning can significantly enhance our ability to predict concrete strength, thereby contributing to more informed decision-making and more robust constructions in the field of civil engineering.

CHAPTER 5

Mapping Data Uncertainty to Concrete Strength Prediction

In this chapter, we focus on the critical role of compressive strength as a definitive gauge of concrete quality, reflecting the material's capacity to resist loads. The prediction of concrete strength, a task of paramount importance, is notoriously complex due to the myriad of variables at play. Traditional models based on physics or chemistry often fall short in accurately predicting this essential property, owing to the intricate and intertwined nature of the influencing factors. It is in this intricate context that machine learning presents itself as a formidable tool, offering the ability to harness historical data to predict future outcomes.

Our investigation centers on the application of Gaussian Process Regression (GPR) [3], a method celebrated for its probabilistic approach to forecasting. GPR stands out by not only predicting concrete strength but also by quantifying the uncertainty associated with these predictions, thereby enriching our understanding and interpretation of the results.

We have honed our GPR model through a meticulous optimization process, setting a new benchmark in accuracy that surpasses established machine learning methods such as polynomial regression, neural networks, and random forests. The validity of our model is underpinned by a robust dataset, featuring around 13,000 samples collected from industrial concrete production. The model's adeptness in capturing the inherent uncertainties of concrete strength predictions is particularly commendable.

Beyond mere prediction, our refined GPR model offers a detailed analysis of how various constituents contribute to concrete strength. This analytical capability is invaluable, providing concrete, actionable insights into optimizing mix proportions. Such insights hold profound significance for material engineering and construction management, offering a pathway to more informed decision-making and optimization in these fields.

In sum, this chapter unfolds the narrative of our rigorous approach to enhancing the predictability and understanding of concrete strength through advanced machine learning techniques. It underscores our commitment to advancing the field of material engineering through innovative data analysis and modeling techniques, a commitment that is well documented in our conference paper and further elaborated in the discussions that follow.

5.1 Overview and realted works

The compressive strength of concrete at 28 day is the most important metric in measuring concrete's engineering properties. As the most produced material in the world, concrete is made of stones [5] and sand [9] that are glued by cement during the hydration process. Cementitious materials such as slag and fly ash are also added to enhance the durability of concrete [13]. Accurate prediction of 28-day compressive strength is essential in construction project since it can save time and material expense by avoiding overdesign [14]. It can also reduce the environmental burden by minimizing use of cement without sacrificing the engineering performance [16]. Although Concrete's strength is mainly controlled by cement fraction and the water to cementitious ratio (w/cm), it also depends on other features such as chemical admixtures, cement and aggregate types [18]. Therefore, it is challenging to model the nonlinear relationship between concrete components and its strength based on physical or chemical principles. As an alternative pathway, machine learning is capable of building models for concrete strength prediction by learning from data [19]. Machine learning algorithms can have various structures and use different strategies to find patterns inside data. The most commonly used learning algorithms include artificial neural network

[20], decision trees, and support vector machines [21]. ANN is a mathematical technique that uses inter-connected nodes called artificial neurons to model nonlinear data, which has been used in many studies to predict the concrete strength. For example, Ahmet Öztaş applied ANN model to predict the compressive strength of high strength concrete on 200 samples with test set R^2 of 0.999. Based on 50 data points Başyigit also used ANN model for compressive strength prediction on heavyweight concrete with test set R^2 of 0.998. Different from ANN, decision trees rely on tree-like model of decisions to find pattern inside data. Based on 49 samples, Palika Chopra used decision tree to predict compressive strength of concrete with test set R^2 of 0.8270 [23]. Halilerdal used hybrid ensemble of decision tree to predict the strength of concrete on 1030 samples with test set R^2 0.8179 [25]. SVM works by constructing hyperplanes in high dimensional data space, where different class of clusters are separated by these hyperplanes. A good separation is achieved when the hyperplane has the largest distance with its nearest class of data points [29]. Based on 500 samples, Ling used SVM to predict strength of concrete in marine environment with average relative error of 27.6 percent [33]. Even though these studies achieve high accuracy when applying machine learning model for concrete strength prediction, the size of dataset is small (less than 1000) and under well controlled lab condition, it is unclear whether these models can be reliable when applied to industry-mixed concrete since the mixture conditions may not be precisely controlled, thus bringing large variance to field concrete samples. Also, some important chemical admixtures such as air entraining agent, water reducing agent, retarder and accelerator are not all considered as input variables, which will limit the universal applicability of those models. Besides, regular machine learning models have difficulties to capture the uncertainties of the dataset. For the concrete strength dataset, the measured strength can have large uncertainties due to change of raw material quality, measure error, the change of testing protocol or sample size . Especially for concrete produced from industry, the measured strength can have large standard deviation due to the unstable production condition and quality control process [35]. A clear understanding

of concrete strength uncertainty will bring many benefits, including saving cost, refine mix design, project-specific concrete proportioning, etc. Thus, it is necessary to develop a model to capture the uncertainty of concrete strength measurement. Based on Bayes Theorem, gaussian process regression [1] model offers solution to provide uncertainty measurement on its predictions [47]. Unlike many popular machine learning algorithms that learn exact value for each parameter of the model, GPR is an algorithm that calculates the probability distribution over all admissible functions that fit the data. GPR is based on the assumption that data samples follow multivariant gaussian process, which is characterized by mean vector and covariance matrix. The using the conditional property of test data, the predictive distribution of test data can be computed. As such, GPR model can be particularly useful in capturing the intrinsic uncertainties in the concrete dataset. Moreover, GPR model's versatility allows it to reliably correlate the non-linear relation between concrete strength and composition by customizing its kernel parameters (Jäkel et al., 2007; Rasmussen and Nickisch, 2010) . This study applies GPR model for concrete strength prediction on a large dataset for industrial concrete production [48]. More importantly, we propose a methodology to optimize the GPR kernel based on both the model accuracy and previous knowledge about concrete strength and composition relationship. The resultant GPR model turns out to have unprecedent accuracy compared to our previous work where non-Bayesian ML models were used to predict the strength of industrial concrete (DeRousseau et al., 2018; Rafiei et al., 2017; Young et al., 2019; Ouyang et al., 2020). Besides, the GPR model provides the confidence interval of its strength prediction, which can be explained by the uneven distribution of concrete dataset. The GRP model also successfully measure the intrinsic uncertainties of the industrial concrete dataset. A reliable estimation of the concrete strength can reduce the concrete overdesign, which lower the production cost and associated CO2 production. Finally, we investigate the decomposed strength contribution from different concrete component. These results are insightful for facilitating the use of GPR model for concrete strength prediction and mixture optimization.

5.2 Dataset of concrete strength measurements

Training data is the key input for ML model to learn information for future direction. Without quality data, ML model cannot operate efficiently. The dataset used in this study comes from industry produced concrete consisting of measured strength of around 13,000 commercial concrete with their mixture components. The samples correspond to commercial mixtures and their compressive strength is measured after curing for 28 days at ambient temperature. These samples are cured for 28 days at which the field strength is measured under ambient temperature. Then nine components are selected as input features for ML model prediction based on permutation feature importance analysis [50]. These input features are major components during concrete mixing process [92]: [60] cement mass fraction, [63] fly ash mass fraction, [66] slag mass fraction, [68] water-to-cementitious ratio [67], [71] fine aggregate mass fraction, [75] air-entraining admixture dosage [74], [77] water-reducing admixture dosage (used for increasing concrete early-stage workability), [78] retarder dosage and [80] accelerator dosage (used for increasing concrete early-stage workability). Among all the mixture s, Type I/II cement [84] compliant with ASTM C150 and Class F fly ash compliant with ASTM C618 [86] are used. The features from 2 to 4 are convert to the solid weight fractions for normalization purposes. The fraction of coarse aggregates is excluded as it is redundant with features (2-to-4) since the four weight fractions always sum up to 100Initial data cleaning is conducted to remove potential outliers in the concrete dataset based on the ensemble of several unsupervised outlier detection approaches. The full details about the data cleaning can be found in another paper [88], which is under review. The relationship between concrete components and 28-day compressive strength is shown in figure 1. We can observe negative correlation between 28-day strength and w/cm and positive correlation between 28-day strength and fraction of cement. However, it is difficult to infer the the correlation of the rest components and 28-day strength based on the plot. This also highlights that the relationship between concrete components and 28-day strength is non-linear.

5.3 Gaussian Process Regression Model

GPR is a nonparametric [91], probabilistic ML approach that utilizes a multivariate Gaussian process to model the nonlinear relationship between input and output. Different from popular machine learning algorithms that learn a function with fixed parameters, GPR learns the probability distribution of functions that fit the training data. In GPR, the data samples are assumed to follow a Gaussian distribution which can be specified using a mean function, $m(x)$, and a covariance function, $k(x, x')$ (Rasmussen and Williams, 2006):

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{5.1}$$

Through selecting the mean (zero for most cases) and covariance functions (kernel), the prior knowledge can be incorporated about the space of functions. Gaussian noise, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, can also be added to the labels by summing the label distribution and noise distribution (Rasmussen and Nickisch, 2010):

$$y \sim \mathcal{GP}(m(x), k(x, x') + \sigma_{ij}\sigma^2) \tag{5.2}$$

Given the training data points and test points, GPR aims to learn the distribution of training data together with test data as a multivariate Gaussian distribution. The predictive distribution about the target variable can be made using conditional probability, which measures the likelihood of the prediction given the training data. Based on the predictive distribution, the mean value can be extracted as the prediction value and the standard deviation is used to infer the uncertainty of the prediction.

5.4 GPR kernels

Kernels are covariance functions that are used by GPR to fit the data. They define the prior knowledge of the function we want to use for fitting the data. The kernel function provides the similarity measurement between data points, which can be used to solve regression problems since the output of a function should be similar when inputs are close. There are various options for the GPR kernel function (Duvenaud, 2014) and each has its unique properties. Some common kernel functions include linear, Matern, rational quadratic (RQ), and radial basis function (RBF) kernel, as well as a composition of multiple kernels.

The linear kernel has the form of

$$k(x, x') = \sigma_b^2 + \sigma_v^2 \exp\left((x - c)(x' - c)\right) \tag{5.3}$$

Similar to linear regression, the linear kernel is used to capture the linear trend in the dataset for the GPR model.

The radial basis function (RBF) kernel has the form of

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{l^2}\|x - x'\|^2\right) \tag{5.4}$$

Where l is the length scale parameter and σ_f^2 is the output variance parameter. This kernel has a smooth function and is indefinitely differentiable. It is the most popular GPR kernel since it can be integrated against most functions.

The RQ Kernel has the form of

$$k(x, x') = \sigma_f^2 \left(1 + \frac{1}{2\alpha l^2}\|x - x'\|^2\right)^{-\alpha} \tag{5.5}$$

This kernel is equal to adding multiple RBF kernels together with different length scales. The α parameter determines the weighting of large length scale and small length scale variance. When $\alpha \to \infty$, the RQ kernel becomes the RBF kernel.

The Matern Kernel has the form of

$$k(x, x') = \frac{1}{\Gamma(v)2^{v-1}} \left(\sqrt{2v}l\|x - x'\|^v\right) K_v \left(\sqrt{2v}l\|x - x'\|\right) \tag{5.6}$$

Where K_v is the modified Bessel function and $\Gamma(v)$ is the gamma function.

One important metric of GPR model is that the kernels can be added or multiplied together to create new kernel structures. This will change the properties of kernel function (smoothness, isotropicity and periodicity), resulting a specialized form that is suitable for specific database. The GRP model is more likely to give optimal approximation to its target due to its enhanced expressivity. Thus, it is necessary to optimize kernel function though kernel combination process. The next section will introduce the kernel optimization process.

5.5 Model training and evaluation

Here we adopt GPR model to predict the 28-day strength the concrete based on GPy torch package. This package uses GPU acceleration and state-of-art of algorithms to reduce the computational complexity (Gardner et al., 2018). The behavior GPR model is controlled by the kernel, which represents the prior knowledge on the dataset. As the key parameter, the methodology of kernel selection will be discussed in detail in the following sections. Besides kernel, a noise level parameter is also specified to estimate the data uncertainty. The parameters of the kernel function and noise level are set as default values given in the package initially. During model training, the parameters of the GPR kernel and noise level will be optimized by maximizing the marginal likelihood using Adam optimizer during training process with learning rate set as 0.1. The accuracy of the model is measured by calculating the root-mean-square error (RMSE) and coefficient of determination (R^2). The RMSE represent averaged Euclidian distance between true strength and predicted strength, and R^2 measures the proportion of the variance between true strength and predicted strength. For model evaluation, we adopt five folds cross validation technique to make the result statistically meaningful. S Firstly we randomly shuffle the dataset and split the data into five groups (2600 data points for each group. Then one group is selected as the test set (the hidden data points from the model) with the remaining data points as the training set. The GRP model will be fit on the training set and evaluated on the test set. This process will be repeated five times until each group is evaluated as test set. The results are averaged iteratively after using each group for evaluation

5.6 GPR kernel optimization

The optimization process of GPR kernel is shown in Figure 5.1. The GPR kernel is selected based on not only the accuracy of the test set, but also our physical understanding of physical properties of concrete. Four kernels (Linear, RQ, Matern and RBF) are used during the optimization process. For each optimization steps, a performance score is calculated based on the accuracy (R^2) of the model and our physical understanding of concrete properties: the concrete strength should increase monotonically with increasing cement fraction or decrease monotonically with decreasing w/cm. To evaluate how well the GPR kernel matches the physical properties of concrete, the predicted strength as function of w/cm and cement is plotted. For each case one hundred concrete formulation is used with gradually increase w/cm or cement fraction. Then the ratio of data points that have higher strength than that of previous data point in strength vs cement plot is counted as a, the ratio of data points that have lower strength than that of previous data point in strength vs w/cm is calculated as b. Then the performance score of the GPR kernel is calculated by 50*R^2 + 25 *(a + b). Here, R^2 represents the accuracy of the GPR kernel, (a + b) represents how well the GPR model matches our understanding of concrete strength. Since the range of R^2 is between [0, 1] and the range of (a + b) is [0, 2], the formula 50*R^2 + 25 *(a + b) is used to make the overall performance score range from [0, 100]. It can also ensure equal weight is put on the accuracy of the GPR kernel and how well it matches the physical properties of concrete. For the first step we calculate the score for each kernel and choose the kernel with highest score as the optimal kernel. In the following steps, we make new kernel by either adding or multiplying the one kernel with the optimal kernel from last step and updated the optimal kernel according to the calculated score for this step.

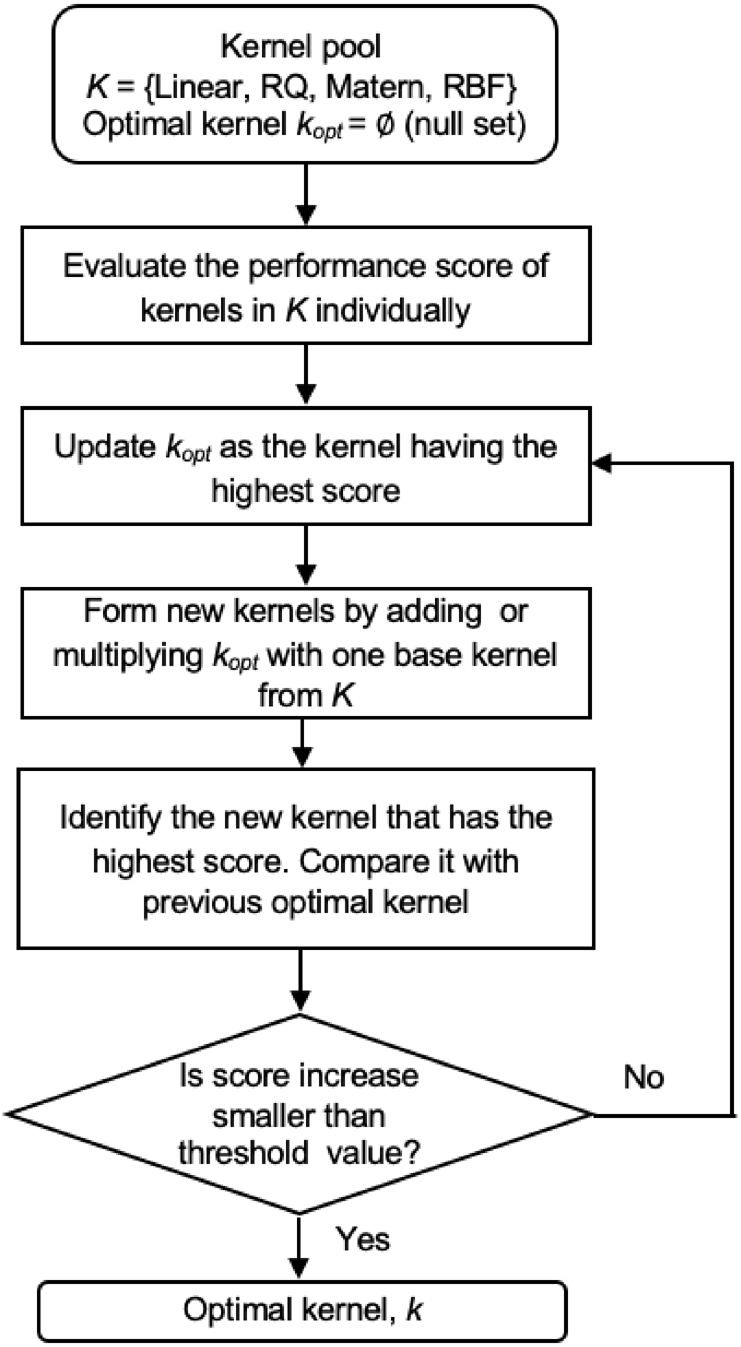
Kernel pool
K = {Linear, RQ, Matern, RBF}
Optimal kernel k_{opt} = Ø (null set)
Evaluate the performance score of kernels in K individually
Update k_{opt} as the kernel having the highest score
Form new kernels by adding or multiplying k_{opt} with one base kernel from K
Identify the new kernel that has the highest score. Compare it with previous optimal kernel
Is score increase smaller than threshold value?
No
Yes
Optimal kernel, k

Figure 5.1: Flowchart of the algorithm used to identify the optimal kernel for the Gaussian process regression (GPR) model.

5.7 GPR model performance based on individual kernel.

As the most important attribute of GPR model, the proper selection of GPR kernel is vital to build a robust GPR model. In order to find the optimal kernel that properly approximate concrete strength based on current dataset, we first investigate the GPR model performance based on individual kernel before the kernel optimization process. It turns out the GPR model based on individual kernel doesn't achieve optimal performance. As discussed before, we consider two criterions for the ideal GPR model. It should have both high accuracy and follow our physical understanding of concrete strength properties. Since it is known that concrete strength is proportional to the cement fraction, it is worth to checking the GPR model can capture such relationship by investigating how the model behave when only varying cement variable. We first investigate the GPR model performance based on individual kernel by plotting the predicted strength as function of cement fraction for GPR model (see Figure 5.2 For all the plots, the input features are set as the average composition and only cement fraction varies. Overall, most of the kernels can achieve an accuracy with R^2 of test set above 0.7 while linear kernel has the lowest accuracy (0.499 test set R^2). However, the GPR model based on Linear kernel follows the physical property of concrete strength (The predicted strength increases monotonically with increasing cement fraction). On the other hand, the GPR model based on the rest kernels exhibits higher accuracy (test set R^2 ¿ 0.7), while the predicted strength doesn't increase monotonically with increasing cement fraction, which contradicts our physical understanding of concrete strength. Even though the GPR model based on RQ kernel has the highest accuracy (0.745 test set R^2), the model is not optimal since it fails capture how concrete strength behave when varying cement fraction. Therefore, none of the individual kernels meet both the criterions for the GPR model. In the following section, we will discuss how we optimize the GPR kernel by combine these individual kernels together.

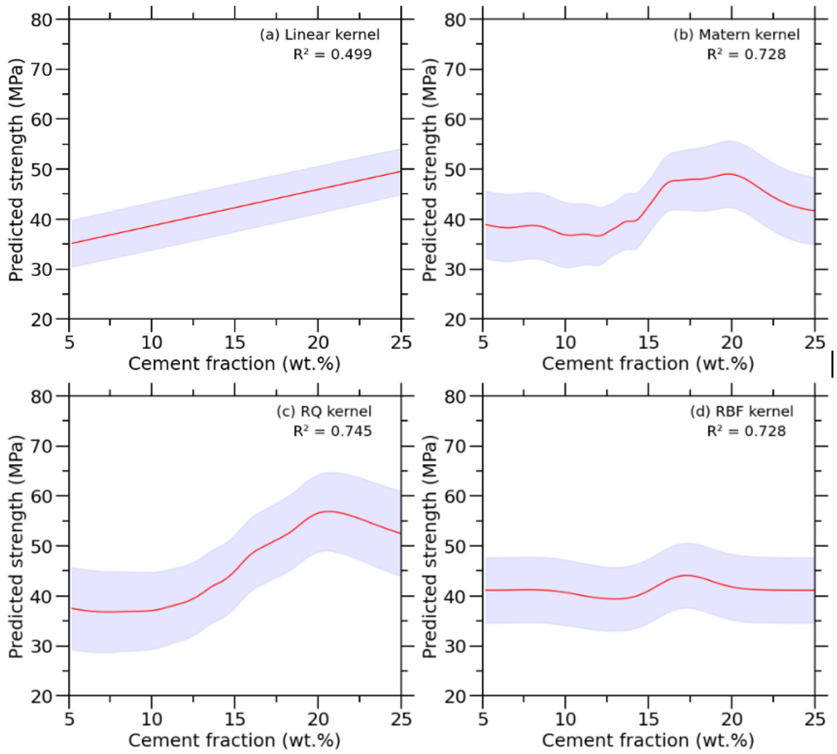

Figure 5.2: Predictions of the change on concrete strength as function of cement fraction for GPR model based on individual kernel ((a) Linear, (b) Matern, (c) RQ, (d) RBF).

5.8 GPR kernel optimization

Having established that GPR model based on individual kernel doesn't achieve optimal performance, we move on to test the GPR model performance on multiple kernel combination. we quantify the GPR model's behavior using a performance score, which is a based on both accuracy and the monotonicity of the strength vs cement or cement vs w/cm plot. Specifically, the monocity is characterized by the ratio of data points that increase with increasing cement on strength vs cement plot (the ratio is assigned as a) or the ratio of data points that decrease with w/cm on strength vs w/cm plot (the ratio is assigned as b). After assigning equal weight (50) to accuracy and monocity, the score is calculated by adding these two factors together (50*R^2 + 25 *(a + b)). In the first step only one kernel that has the highest score is chosen as the optimal kernel. In the following steps, one additional kernel will be combined with the optimal kernel from previous step until the increase of performance score is smaller than threshold (0.01). Figure 5.4a shows the score of best performance GPR kernel for each optimization step. Since for each step a new kernel is added or multiplied with the kernel in the previous step (start from single kernel in step one), the model becomes more complex after each optimization step. We can observe that the score has a large increase from step one to two while the increase of score becomes minimum after second step. Thus, the optimal kernel doesn't need to be the kernel with the highest score, it needs to achieve a balance between model accuracy and complexity. To better illustrate how the optimal kernel is chosen, Figure 3b exhibits the relative change of score for each optimization step. It can be noted that the change of score after step 4 is smaller than the threshold (i.e., 10-2), so the best GPR kernel generated from Step four (combination of four individual kernels) is chosen as the final kernel for the GPR model. The best performance GPR kernel and the error distribution for the corresponding GPR model at each optimization step is shown in figure 4. Starting with linear kernel in step 1, the distribution of the absolute error becomes narrow after adding or multiplying new kernel in each step. The final GPR model is based on the kernel with the structure: (Linear + RQ) * RQ + Matern. Through kernel optimization, the GPR model is able to modify its prior knowledge about the data and combine the strength of single kernels. As we can see from figure 2, the linear kernel is able to capture the monotonic trend of strength as function of cement or w/cm while the rest of the kernels have the advantage of high accuracy. Figure 5.5 and Figure 5.6 illustrate how the predicted strength vary with cement fraction or w/cm for the best performance kernel in each optimization step and their corresponding R^2. The range of cement fraction and w/cm in the plot is consistent with the input feature range from the dataset. From figure 5 we can see that the slop of predicted strength as function of cement fraction or w/cm first stays constant for all the range of cement fraction for linear kernel, then it starts to vary with cement fraction or w/cm as we combine more kernels with linear kernel. The GPR model based on linear kernel is equivalent to Bayesian linear regression, which is only capable of building linear model, which is not sufficient to capturing the nonlinear relation between concrete components and strength. After kernel optimization, the GPR models is able to capture the nonlinear relation between concrete components and strength by increasing R^2 from 0.499 to 0.747. The resultant GRR model also meets our physical knowledge of concrete strength since the predicted strength increases monotonically with increasing cement fraction and in figure 5.6 and the predicted strength decreases monotonically with increasing w/cm. Moreover, the shaded area represents the confidence interval of strength prediction. It remains constant at step one and step two, but it starts to vary when we continue to optimize the GPR kernel. For instance, the optimal GPR model has small standard deviation at intermediate strength level, which means the model is more confident in predicting median-strength concrete.

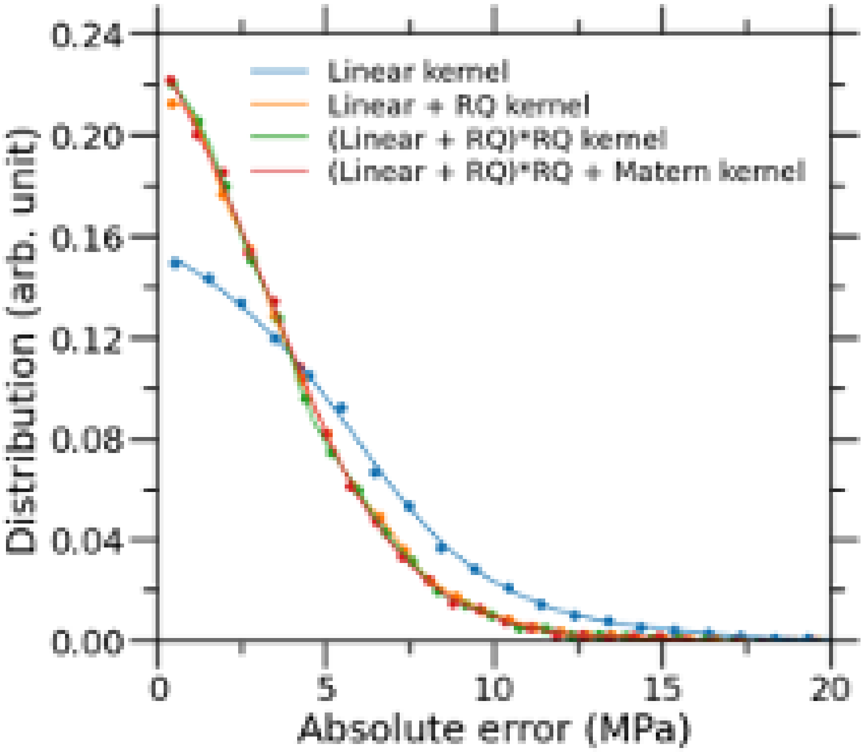

Figure 5.3: Distribution of the absolute strength prediction error of the GPR models with different kernel combinations

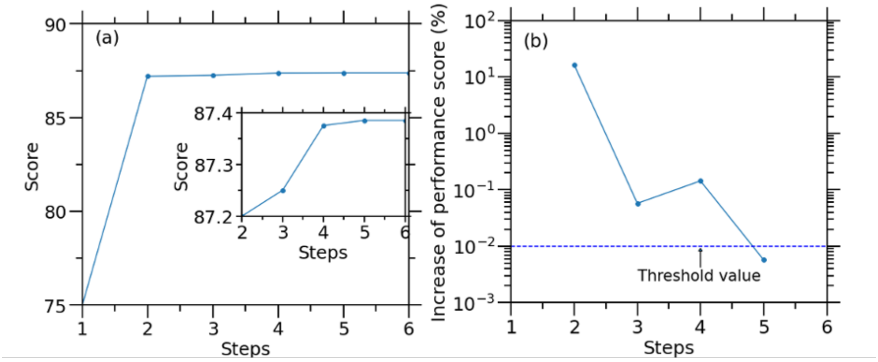

Figure 5.4: Predictions of the change on concrete strength as function of cement fraction for (a) Linear kernel, (b) linear + RQ kernel, (c) (Linear + RQ)*RQ kernel and (d) (Linear + RQ)*RQ Matern kernel , as yielded from the GPR models with different kernel combinations

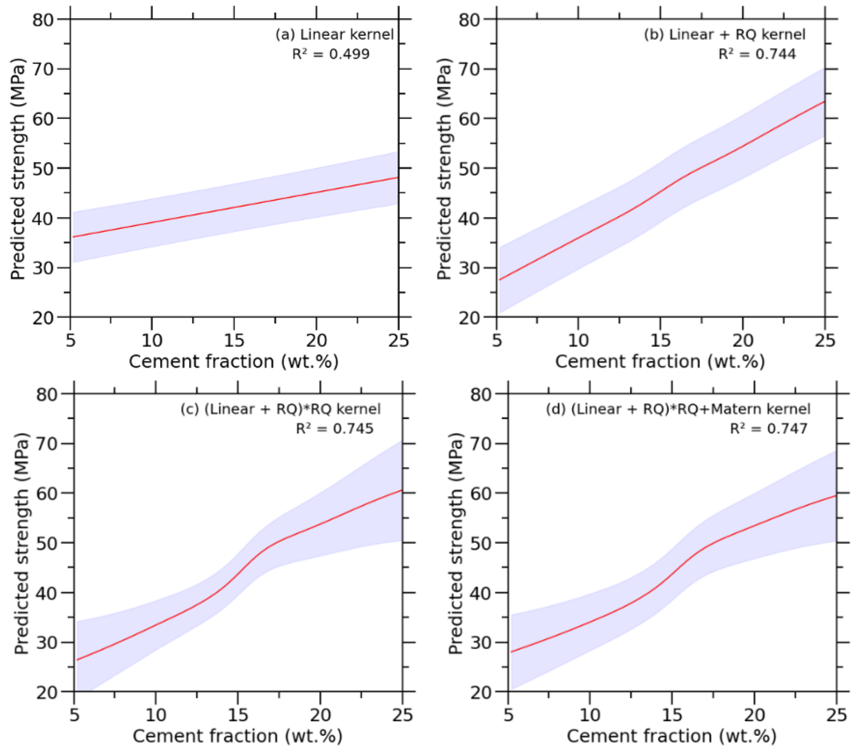

Figure 5.5: Predictions of the change on concrete strength as function of cement fraction for (a) Linear kernel, (b) linear + RQ kernel, (c) (Linear + RQ)*RQ kernel and (d) (Linear + RQ)*RQ Matern kernel , as yielded from the GPR models with different kernel combinations

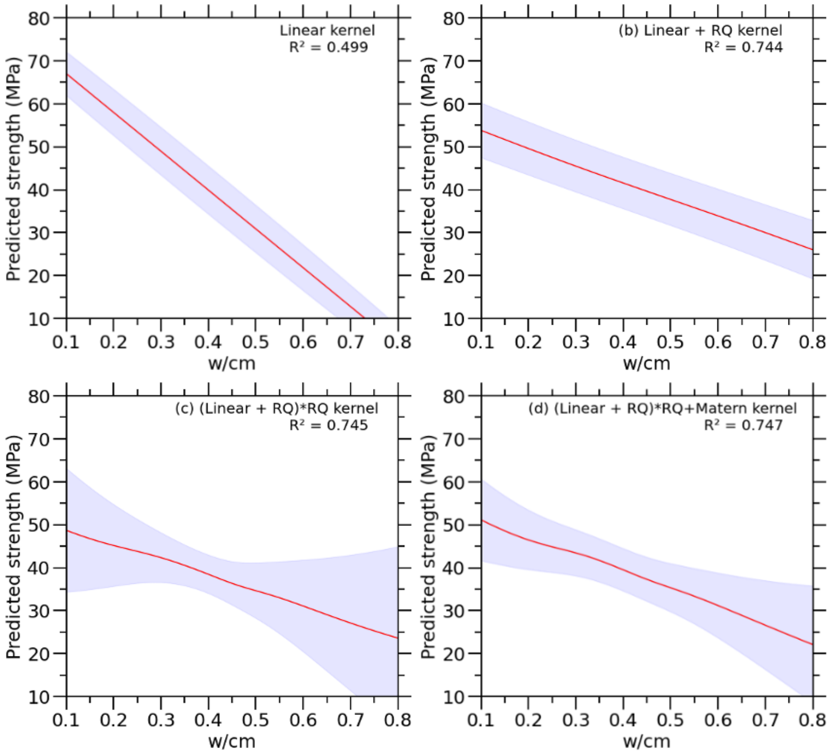

Figure 5.6: Concrete strength values predicted by GPR model as function of W/CM for (a) Linear kernel, (b) linear + RQ kernel, (c) (Linear + RQ)*RQ kernel and (d) (Linear + RQ)*RQ + Matern kernel , as yielded from the GPR models with different kernel combinations

5.9 Compare GPR model with other machine learning models.

After choosing the optimal GRP kernel, we further compare the GPR performance with other machine learning models. The detailed setting of ML models other than GPR is given in supplementary material. Figure 5.7 shows the predicted strength value for the test set (the unknown samples hidden from tranining) as funciton of the true strength values. The y=x line represents the ideal agreement between the predicted strength and actual strength values. The GRP model shows the highest accuracy with 0.75 R^2 and 3.98 MPa RMSE, proving its capability to to caputre the relation between concrete components and its strength. On the contraray, the PR model has the lowest accuracy due to its limitation as additive model. While ANN and RF have improved accuracy compared to PR model , they still can not achive the same accuracy as the GPR model. In addition, the color of the pixel exhibits the overlapping of the local datapoints. It can be noted that the data is mainly distributed in mdeian strength value (30MPa – 50MPa) range since the ovelapping over datapoints in that domain is much higher for the than the poins in other strength interval. For instance, the yellow region correponds to datapointsthat has the highest density (102 of overlap, which is located in the region of (35MPa – 40MPa) This can also explain the small standardeviation of median-vlaue predicted strength in figure 5.5 and figuere 5.6. The GPR model turns out to be the optimal model for concrete strength prediction with highest R^2 and lowet RMSE.

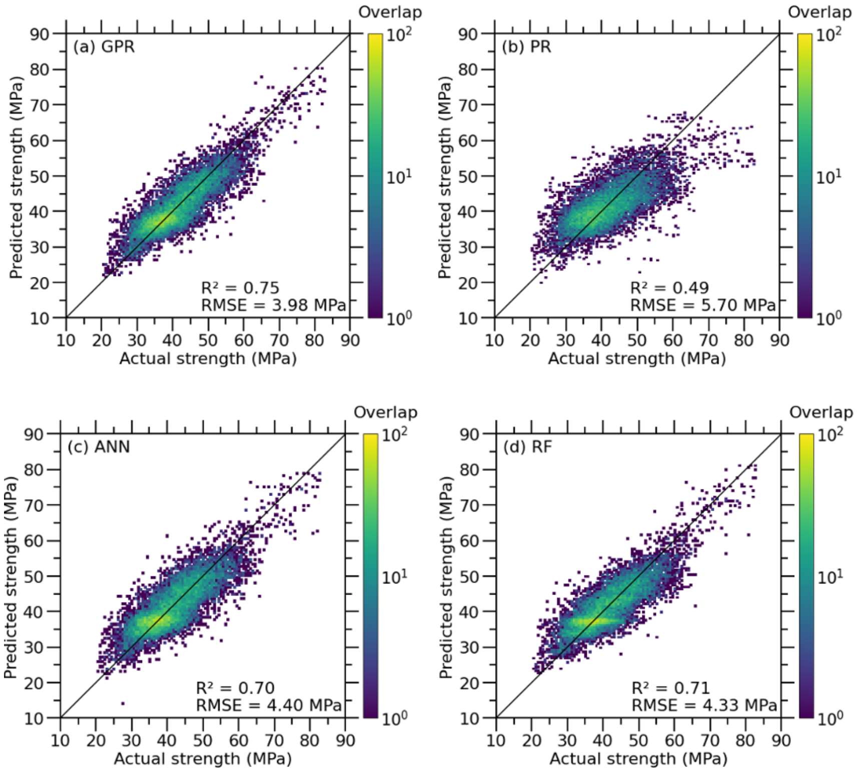

Figure 5.7: A comparison between the predicted and measured concrete strength for the test set samples, based on (a) polynomial regression (PR), (b) artificial neural network (ANN), (c) random forest (RF), (d) gaussian process regression models (GPR). Note that the color of the data points indicates the degree of overlapping in the plot

5.10 GPR model's ability to capture the data uncertainty

The advantage of the GPR model is that it not only predicts the concrete strength, but also captures the uncertainty by giving the confidence interval. To investigate whether the GPR model can capture the intrinsic uncertainties of the concrete dataset, figure 8a shows the strength distribution of concrete data set as function of 95 percent confidence interval of the predicted strength. We can see that the majority of the concrete data has 95 percentconfidence interval less than 5 MPa while less than two percent of data have 95 percent confidence interval larger than 10 MPa. Figure 5.8b shows the 95 percent confidence interval as function of 28-day strength of concrete, illustrating that the datapoints that have lowest 95 percent confidence interval (around 4 MPa) are located in the intermediate strength region (30-40 MPa). This confirms that the 95 percent confidence interval output by the GPR model is affected by the strength distribution of dataset. In addition, a polynomial fit of the data (solid line) is given in figure 5.9b. The uneven slop in both sides of the fitted line shows the uncertainty increment is different when deviating from lowest point. We further investigate the density of datapoints as function of 28-day strength (figure 5.9a), which shows the uneven strength distribution of the concrete datapoints. Also, figure 5.9b shows that the 95 percent confidence interval of the predicted strength is inversely proportional to the density of datapoints. It shows that the GPR model tends to have high confidence in the region of high-density data points. The solid line is made by polynomial fit of the data, which serves as the visual guide to the eye. The dashed line corresponds to lowest level of 95 percent confidence interval, which can be interpreted as the intrinsic uncertainty of concrete dataset. The intrinsic uncertainty value (3.8 MPa) is close to the literature report, proving its ability to capture the intrinsic uncertainty in the concrete data.

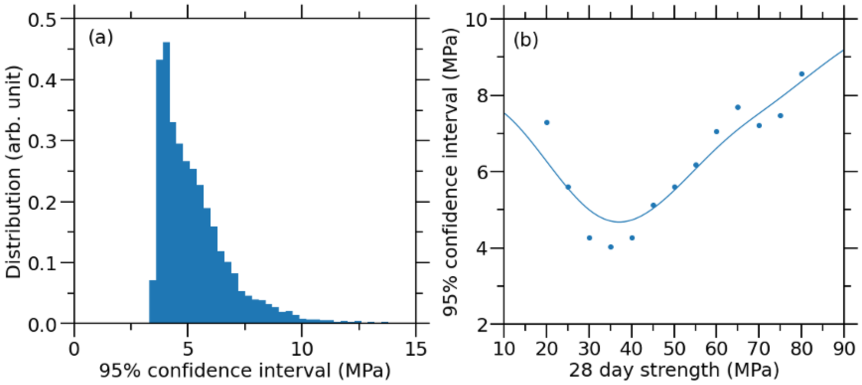

Figure 5.8: Variation of the uncertainty of the GPR model prediction: (a) Distribution of 95 percent confidence interval based on GPR model's prediction using concrete data set and (b) the 95 percent percent confidence interval as function of 28day strength

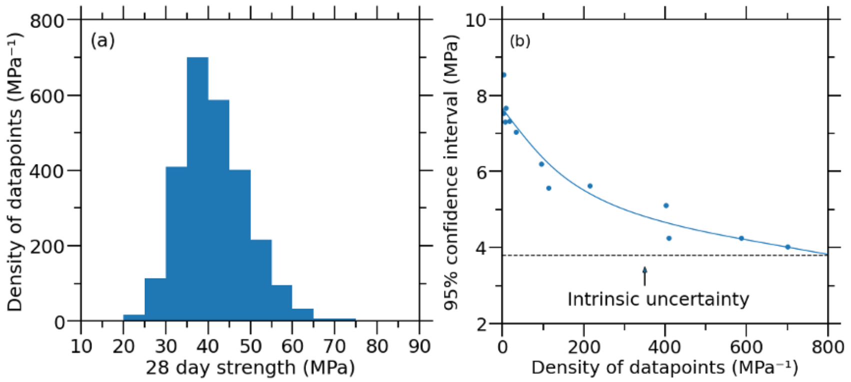

Figure 5.9: Density of concrete datapoints as function of (a) 28-day strength and (b) 95 percent confidence interval as function of density of concrete data points

5.11 Concrete strength development when varying the concrete composition

The results in this section are mainly involved with strength prediction based on extrapolation and multi-variant optimization problems. Specifically, the GPR model is used to predict the strength development by varying the concrete composition. Three different cases are investigated. In the following sections, we not only show the strength development as function of individual component, but also the combine effect of two or three concrete components using heat map

5.12 Replace cement with fly ash and slag

We now investigate the strength development when replacing cement with fly ash and slag. The initial values of cement fraction (20 wt.percent) fine aggregate (37 wt.percent) and coarse aggregate (43 wt.percent) fractions are determined by using the average composition of the dataset, with slag and fly ash fraction set to zero. Figure 5.10a shows that the strength decreases when replacing cement with class F fly ash, which is consistent with the experimental result in literature. On the contrary, replacing cement with slag can slightly increase the predicted strength, showing the feasibility of slag as cement replacement during concrete mixing process . Figure 5.11a shows the combined effect of fly ash and w/cm on strength development. As we see in figure 5.10a, the predicted strength decreases with increasing w/cm. Also, the w/cm variable only shift the predicted strength to lower value without changing the negative correlation between concrete strength and cement replacement by fly ash. Figure 11b shows the combined effect of w/cm and slag on strength development. Similar to fly ash, the w/cm doesn't affect the positive contribution of slag to concrete strength development. Finally, the combined effect of w/cm, slag and fly ash on concrete development is shown in figure 5.11c. Consistent with previous analysis, the concrete compositions that yield the highest strength are in the region contains high cement content and low fly ash content.

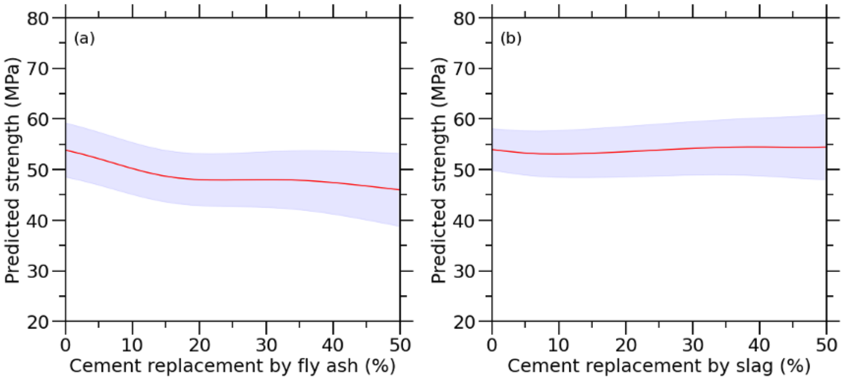

Figure 5.10: Variation of the predicted stre ngth with concrete composition: (a) predicted strength as function of cement percentage replacement by fly ash and (b) cement replacement by slag

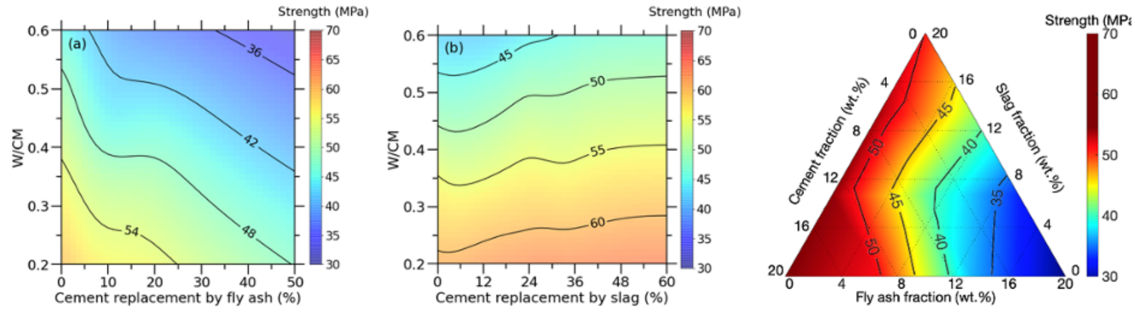

Figure 5.11: Variation of the predicted strength with concrete composition: (a) Contour map of predicted concrete strength as function of cement replacement by fly ash and w/cm, (b) predicted strength as function of cement replacement by slag and w/cm

5.13 Vary the content of aggregates

Next, we investigate the strength development when by varying the aggregates content. The first case is percentage of cement replaced by aggregates while fixing the ratio of fine and coarse aggregate (see figure 5.12a). Similar case is also found in literature .The trend is that replacing cement with aggregates will decrease the predicted strength due to the dilution of cement content. The second case is changing fine aggregate/coarse aggregate ratio while fixing the rest concrete components (see figure 5.12b). There exists one composition that has highest strength when changing fine aggregate/coarse aggregate ratio. Figure 13a shows the combined effect of w/cm and cement replacement by aggregate on strength development, again the predicted strength decreases with cement replacement by aggregates at various w/cm. Figure 5.13b shows the combined effect of w/cm and fine aggregate/coarse aggregate ratio on strength development. It can be noted that when w/cm decreases, the optimal fine aggregate/coarse aggregate ratio that yields highest predicted strength will also increase. Currently there is no experimental study about the effect of w/cm on optimal fine aggregate/coarse aggregate ratio, the finding from figure 13b can serve as new knowledge for concrete strength development. From figure 5.13c we can see the region that has high cement fraction and low aggregates fraction will yield high strength, which meets our expectation.

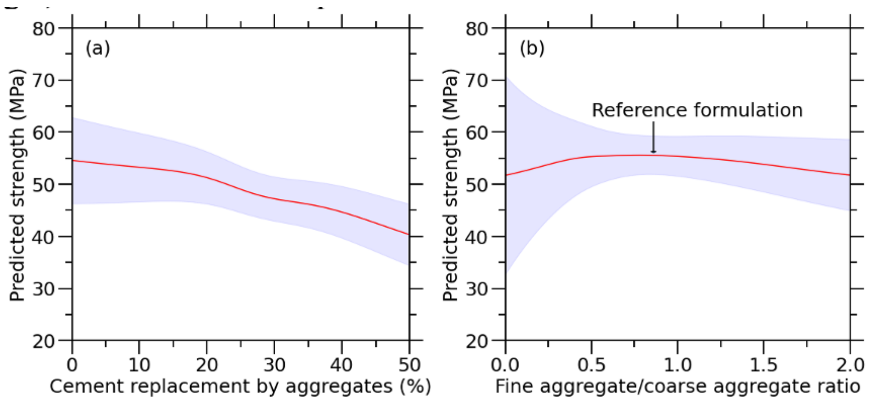

Figure 5.12: Variation of the predicted strength with aggregates: (a) predicted strength as function of cement percentage replacement by aggregates and (b) fine aggregate/coarse aggregate ratio

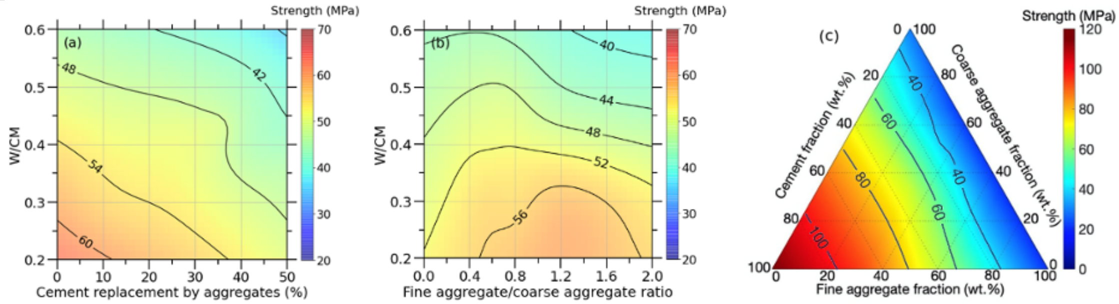

Figure 5.13: Variation of the predicted strength with aggregates: contour map of predicted strength as function of (a) cement percentage replacement by aggregates and w/cm, (b) predicted strength as function of coarse aggregate replacement by fine aggregate and w/cm, (c) ternary map showing predicted strength as function of coarse aggregate, fine aggregate and cement

5.14 Vary water reducing admixture and air entraining agent content

Finally, we investigate how the predicted strength change when we vary the composition of Air entraining agent or water reducing admixture. As expected, the predicted strength decreases with increasing strength of air entraining agent (see figure 14a) since it will introduce air inside concrete. the predicted strength also decreases with increasing strength of water reducing agent (see figure 5.14b) since it will increase the flowability of concrete under fixed w/cm, thus reducing the strength. For the binary plot, we don't observe synergic effect of w/cm with water reducing admixture (figure 5.14b).

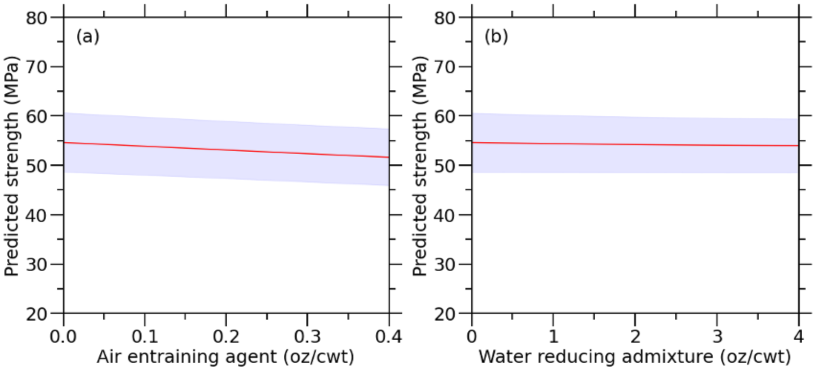

Figure 5.14: Variation of the predicted strength with chemical additives: predicted strength as function of (a) air entraining agent and (b) water reducing agent

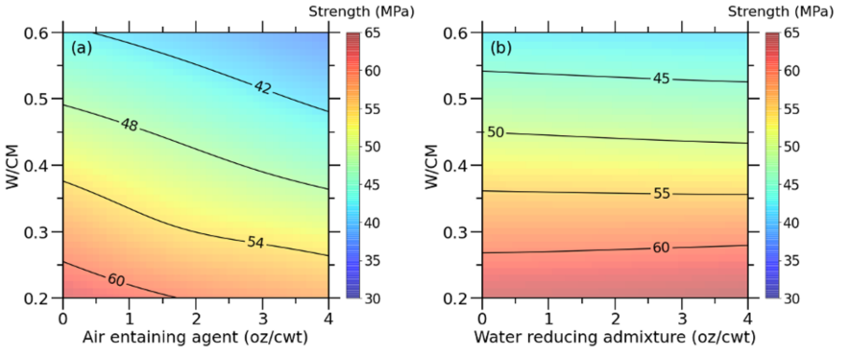

Figure 5.15: Variation of the predicted strength with chemical additives: contour map of predicted strength as function of (a) air entraining agent and w/cm, (b) predicted strength as function of water reducing agent and w/cm

5.15 Conclusions

Overall, we propose a strategy to optimize the GPR model based on both the accuracy and our physical knowledge of concrete strength development. The resultant GPR model exhibits unprecedented accuracy compared with other machine learning models (PR, ANN and RF). We also prove that GPR Model is able to capture the intrinsic uncertainties of the concrete dataset by giving the 95 percent confidence interval of predicted strength, which is inversely proportional to the density of datapoints. At last, we investigate the combined effects of various concrete components on the strength development, the overall trend still agrees with our physical knowledge of concrete properties. More importantly, the developed GPR model will serve as a guide in concrete mixture design process.

CHAPTER 6

Interpretable Concrete Strength Prediction and Extrapolation using Symbolic Regression

Machine learning [5] approaches have gained popularity in predicting the relationship between concrete strength and its composition. However, conventional ML models like artificial neural networks [7] and random forest [11] lack the ability to provide physical insights into concrete properties due to their black-box nature. In this work, we introduce symbolic regression as a powerful regression technique that searches for optimal mathematical equations and parameters. We employ symbolic regression to build an interpretable model for concrete strength prediction and mixture design. Furthermore, we demonstrate the superiority of symbolic regression over popular ML models in extrapolating concrete strength into unknown domains. To enhance the performance of ML models, we also employ data augmentation techniques such as SMOTE (Synthetic Minority Over-sampling Technique) for both interpolation and extrapolation.

6.1 Overview and realted works

Concrete, renowned for its low cost, durability, and wide availability, stands as the most widely used construction material globally [12]. The 28-day compressive strength of concrete serves as a convenient metric for quality control and assessing its engineering performance. Additionally, it exhibits correlations with other mechanical properties, such as elastic modulus and tensile strength. Accurately predicting concrete strength holds significant importance in reducing overdesign in concrete mixtures, resulting in reduced material usage during the construction process. Consequently, minimizing material expenses and associated CO2 emissions linked to concrete production becomes achievable.

Despite previous efforts, the accurate prediction of concrete 28-day compressive strength remains challenging. In early days, empirical models have been developed to correlate concrete strength given a concrete mixture of different components (Namyong et al., 2004; Popovics, 1990; Popovics and Ujhelyi, 2008; Zain et al., 2002). The most famous one is Abram's law which states the strength of concrete is determined by the ratio of water to cement. Other researchers further extend Abram's law to include other variables in the form of multilinear regression equation. However, empirical approaches are not sufficient to describe the nonlinear concrete strength -composition relationship.

On the other hand, with the development of artificial intelligence, ML approaches have been increasingly used to predict concrete strength. Through adopting learning algorithms, ML can directly learn from the data to model the relationship between concrete strength and mixture components. Up to date, several studies have shown that ML algorithms can give highly accurate results in terms of concrete strength prediction Ahmet Öztaş applied ANN model to predict the compressive strength of high strength concrete on 200 samples with test set R^2 of 0.999 (Öztaş et al., 2006). Based on 50 data points Başyigit also used ANN model for compressive strength prediction on heavyweight concrete with test set R^2 of 0.998 (Başyigit et al., 2010). Different from ANN, decision trees rely on tree-like model of decisions to find pattern inside data. Based on 49 samples, Palika Chopra used decision tree to predict compressive strength of concrete with test set R^2 of 0.8270 [13]. Halilerdal used hybrid ensemble of decision tree to predict the strength of concrete on 1030 samples with test set R^2 0.8179 [19]. SVM works by constructing hyperplanes in high dimensional data space, where different class of clusters are separated by these hyperplanes. A good separation is achieved when the hyperplane has the largest distance with its nearest class of data points [24]. Based on 500 samples, Ling used SVM to predict strength of concrete in marine environment with average relative error of 27.6 percent Ling et al., 2019). Nevertheless, those popular machine learning models are black box in nature, which makes it difficult to interpretate the results and gain physical insights from the model [26].

Accordingly, this research extends the body of knowledge by evaluating the capability of the Gaussian Process Regression [27] and symbolic regression [29] for concrete strength interpolation [34] and extrapolation [39]. GPR is a robust non-linear prediction model, it is a probabilistic, nonparametric, supervised, and unsupervised learning method that generalizes the non-linear and complex function mapping on the dataset [41]. Recently, GPR has increasingly attracted the attention of researchers from different engineering fields[42]. Due to the application of kernel functions, GPR can handle non-linear data.

On the hand symbolic regression is an algorithm that search over space for the optimal mathematical equations that best predict the outputs given the input variables [46]. In addition, different from traditional linear or polynomial regression, symbolic regression doesn't have fixed form of equation. Instead, it forms mathematical equation by searching simultaneously the parameters and the form of equation [92]. Therefore, symbolic regression has the potential to model the complex relationship between concrete strength and mixture components. The interpretability of symbolic regression model will also provide physical insight to concrete science and serve as guidance for concrete mixture design.

The contributions of this research lie in evaluating the capabilities of symbolic regression and ML models in concrete strength extrapolation. We investigate the combined use of symbolic regression and data augmentation techniques to enhance the performance of ML models. By comparing the results with popular ML models, we emphasize the advantages of symbolic regression in providing interpretable models while achieving accurate extrapolation of concrete strength.

This research extends the existing knowledge by demonstrating the advantages of symbolic regression in concrete strength extrapolation. By leveraging its ability to capture complex relationships, symbolic regression offers interpretable models that provide valuable physical insights into concrete science and guide concrete mixture design. Furthermore, by integrating data augmentation techniques such as SMOTE, we enhance the performance of ML models and improve their interpolation and extrapolation capabilities. This work holds significance in the construction industry, contributing to optimized concrete mixture design, reduced material waste, and minimized environmental impact.

6.2 Dataset

The dataset is comprised of 1184 concrete samples with various mixture composition and measured compressive strength. Those samples are produced in lab condition using ASTM C150 compliant Type I/II cement and Class F fly ash compliant with ASTM C618. The compressive strength measured is conducted at different curing age. Six most influential features are selected as model inputs based on permutation feature importance analysis [55]. Those features are: [58] water-to-cement ratio (W/CM, mass basis), [60] cement [67], [73] fly ash [76], [81] slag [82], [89] coarse aggregate (CA, in kg per m3 of concrete), (6) fine aggregate (FA ,in kg per m3 of concrete) and (7) curing time (T, in days). Since one sample can have strength measurement at 3, 7, 28 and 56 days, we repetitively use same sample composition but tested in different days as multiple model inputs. We end up having 4286 input data with concrete strength label measured at different days. The dataset used in this research consists of 1184 concrete samples that were produced in a laboratory setting. The samples were created using ASTM C150 compliant Type I/II cement and Class F fly ash compliant with ASTM C618. The compressive strength of the concrete was measured at different curing ages. To identify the most influential features, a permutation feature importance analysis was conducted. Based on this analysis, six features were selected as inputs for the model: Water-to-cement ratio (W/CM): The ratio of water mass to cement mass in the concrete mixture. Cement (CE): The mass of cement per cubic meter of concrete. Fly ash (FLA): The mass of fly ash per cubic meter of concrete. Slag (SL): The mass of slag per cubic meter of concrete. Coarse aggregate (CA): The mass of coarse aggregate per cubic meter of concrete. Fine aggregate (FA): The mass of fine aggregate per cubic meter of concrete. Additionally, the curing time (T) in days was included as a feature. Since each sample had multiple strength measurements at different curing ages (e.g., 3, 7, 28, and 56 days), the same sample composition was used as multiple inputs for the model. This resulted in a total of 4286 input data points, with the concrete strength measured at different curing ages serving as the label. By selecting these influential features and utilizing the multiple measurements at different curing ages, the dataset provides a comprehensive representation of the concrete mixture compositions and their corresponding strengths. This dataset will be used to evaluate the performance of the models, including symbolic regression and machine learning approaches, in accurately predicting concrete strength and extrapolating it to different curing ages.

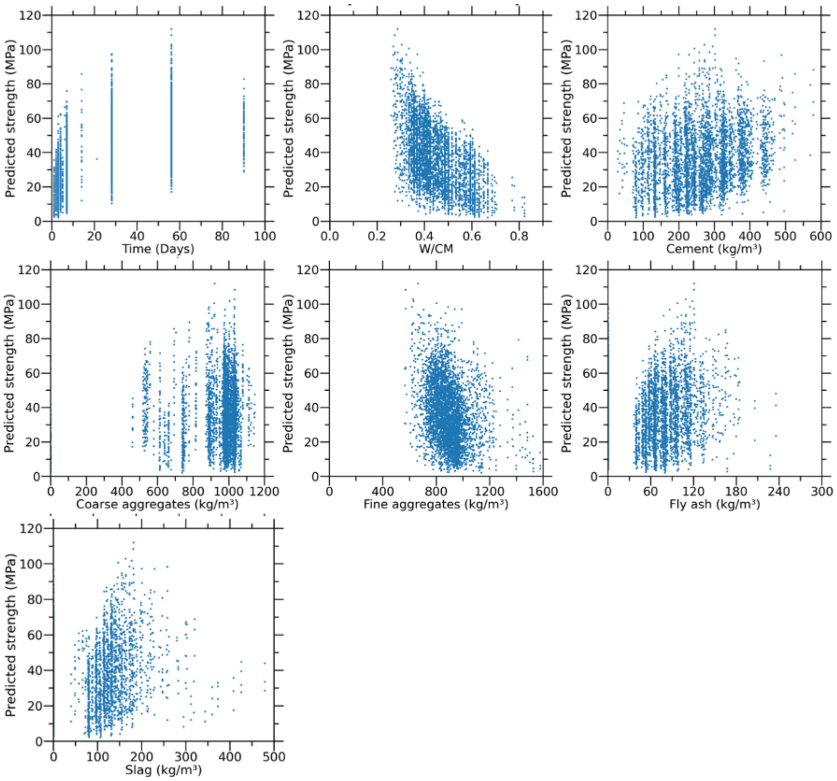

Figure 6.1: Scatter plot of lab data.

6.3 Data augmentation

To overcome the challenge that we only have a small dataset for training the machine learning model, we employ Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) to over sample the training set. SMOTE is a popular data augmentation technique used in machine learning to address class imbalance in datasets. Class imbalance occurs when one class of the target variable is significantly underrepresented in the dataset compared to others. This can lead to biased and inaccurate models that perform poorly in real-world scenarios. SMOTE works by generating synthetic samples from the minority class by interpolating between the existing samples. This is done by randomly selecting a minority class sample and its k-nearest neighbors, and then generating synthetic samples along the line segments connecting them. The algorithm works as follows: Identify the minority class samples in the dataset. For each minority class sample, find its k nearest neighbors. Select one of the k nearest neighbors randomly and generate a synthetic sample along the line segment between the minority sample and the selected neighbor. Repeat steps 2 and 3 until the desired level of minority class oversampling is achieved. The resulting dataset will have a higher proportion of the minority class, reducing class imbalance and improving model performance. In this work, we used SMOTE to generate synthetic training data points to improve the performance of machine learning models. As shown in figure 1, The constant distribution of the synthetic data (blue points) and the original lab data (orange points) shows SMOTE successfully generate synthetic data that resemble the lab data.

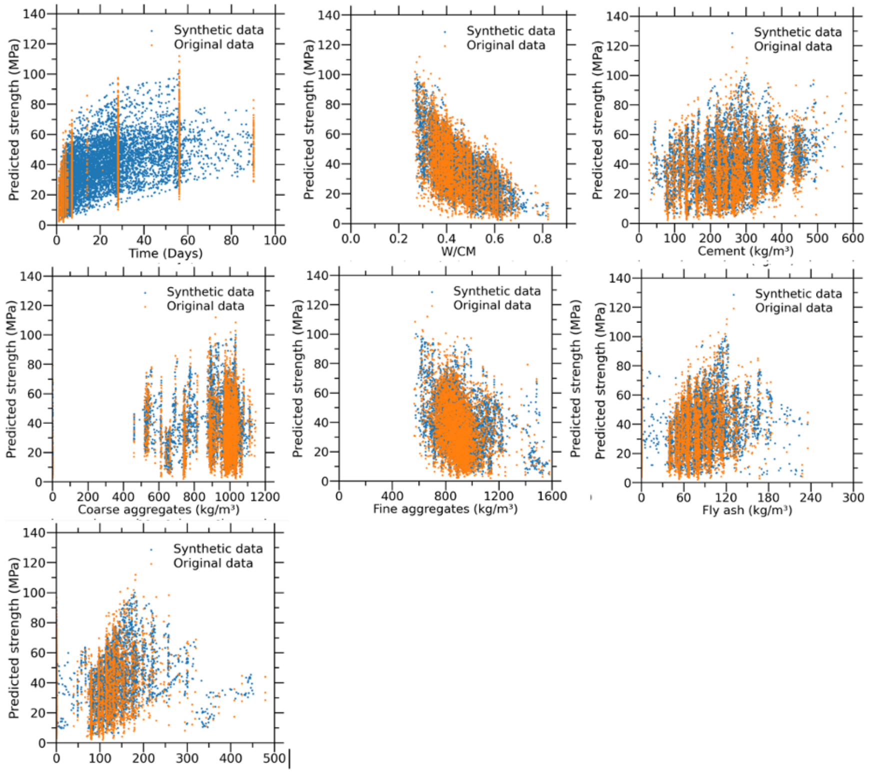

Figure 6.2: Scatter plot of lab data.

6.4 Data splitting for concrete strength interpolation

In regression tasks, interpolation refers to estimating the target variable within the range of the training data. To facilitate effective model learning across the entire data space, the dataset is split randomly into three subsets: training set (80 percent of the data), validation set (10 percent of the data), and test set (10 percent of the data). This random splitting ensures that the model captures the feature information comprehensively. Furthermore, the dataset contains data with the same features except for the age variable, which represents the curing time. It is essential to consider the uneven distribution of data with different age values. While the majority of concrete samples have 28-day strength measurements, only a few samples have measurements at other ages (e.g., 2, 4, 7, 56, etc.). Since curing age significantly impacts concrete strength, it is crucial to ensure an even distribution of data across the age feature in the training, validation, and test sets. To achieve this, stratified sampling is employed during the data splitting process. This ensures that each subset contains a proportional representation of concrete samples across different age values. By stratifying the data based on the age feature, the model will have sufficient exposure to concrete samples with various curing ages, enabling accurate interpolation of concrete strength. During model training, nine mathematical operators (e.g., addition, subtraction, multiplication, division, hyperbolic tangent, natural logarithm, exponential, square, square root, cube) are used to construct mathematical formulas that link selected concrete features to the compressive strength. The optimal mathematical equation is selected based on a comparison of model accuracy and complexity using the validation set. The optimal model is determined by its high accuracy and low complexity. The performance of the model in interpolating concrete strength is evaluated using the test set, which remains hidden from the entire training process to ensure that the data is unknown to the model. By evaluating the model's performance on unseen data, we can assess its ability to accurately interpolate concrete strength within the known data range.

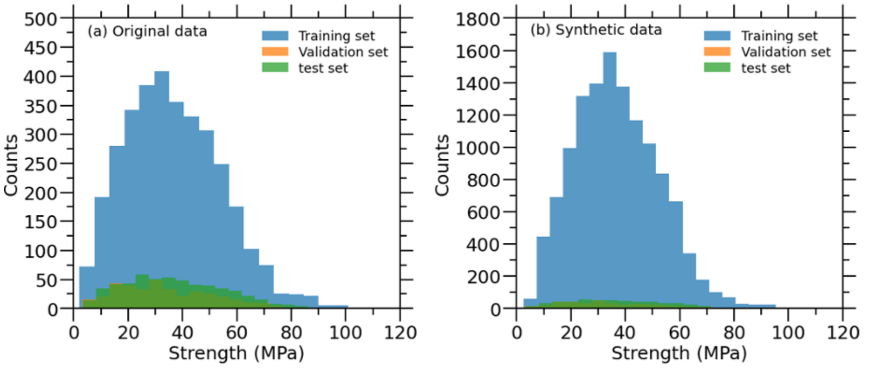

Figure 6.3: strength distribution of training, validation and test set for concrete strength interpolation.

6.5 Data splitting for concrete strength extrapolation

In extrapolation tasks, interpolation refers to estimating the target variable outside the range of the training data. In the context of concrete strength prediction, this means predicting concrete strength values that exceed the range of strengths present in the training data. For concrete strength extrapolation, a different approach is taken for data splitting compared to interpolation. The data is split based on the value of compressive strength, with high-strength concrete samples assigned to the test set, while low-strength concrete samples are assigned to the training and validation sets. The objective is to train the model using low-strength concrete data and evaluate its performance in extrapolating towards the high-strength region. Figure 6.4 illustrates the distribution of data in the training, validation, and test sets. It can be observed that the test set consists of concrete samples with compressive strength ranging from 50 MPa to 110 MPa, representing the high-strength region. In contrast, the training and validation sets predominantly contain concrete samples with strength values lower than 70 MPa, representing the low-strength region. It is important to note that there is a small overlap between the strength ranges of the training and test sets, creating a transitional area. By including this transitional area in the training set, the model can learn from the overlap and better extrapolate towards the high-strength region. This approach helps to improve the model's ability to generalize and accurately predict concrete strength values beyond the range of strengths present in the training data. The data splitting strategy for concrete strength extrapolation ensures that the model is trained on a representative range of low-strength concrete data while being tested on the challenging task of extrapolating towards high-strength values. This allows for a comprehensive evaluation of the model's performance in extrapolation tasks, providing insights into its capability to accurately predict concrete strength beyond the known data range.

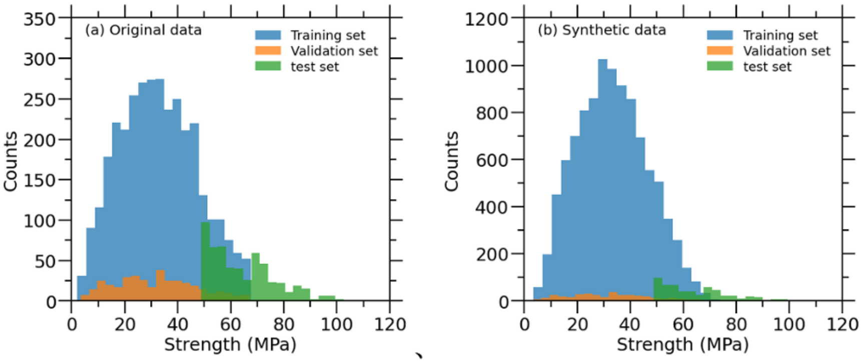

Figure 6.4: strength distribution of training, validation and test set for concrete strength extrapolation.

6.6 Symbolic regression

Symbolic regression is a technique that involves finding a mathematical model to fit observed data without making any assumptions on the function's specific form. Instead, a space of mathematical building blocks, such as mathematical operators, constants, state variables, and analytic functions, is provided, and the most suitable solution is sought by searching through this space (Wang et al., 2019). This means that both model structures and model parameters are optimized. In contrast to conventional regression techniques, optimization algorithms used in symbolic regression are different, and in this section, one of the most prevalent methods, genetic programming (GP), is introduced. In GP, solutions are represented as tree-structured chromosomes with nodes and terminals, and the solution to a given problem is evolved by following Darwin's theory of evolution. The process involves generating a set of initial terminal nodes and functions, forming individual trees with different sizes and structures. Specifically, these tree structures represent mathematical equations that are comprised of nodes randomly selected from a pool of mathematical operators, constants, and input features. Each tree is evaluated based on its performance of modeling the data and model complexity. The initialization process ends once the number of individuals reaches a user-defined population size, where the natural selection process comes into play. The fitness of each individual solution in the initial population is then evaluated by comparing their function output with the true value from the dataset, and GP evolves the current generation by randomly applying genetic operations to individuals. The higher the fitness score, the higher the chances of an individual being selected as a parent. The model with higher fitness score has higher probability to enter the next stage of evolution process through mutation, crossover and recombination operations. In the process of evolving these expressions, genetic operators such as mutation, crossover, and reproduction are used to generate new individuals from the current population. Mutation is a genetic operator that introduces random changes to an individual's genetic makeup(Karaboga et al., 2012). In symbolic regression, mutation involves randomly selecting a sub-tree of the individual's expression and replacing it with a randomly generated sub-tree. This can introduce new variables, constants, or functions into the expression, which can help the population explore new areas of the search space. Mutation is a key operator in GP as it can help prevent premature convergence and introduce novel solutions. Crossover is another genetic operator that combines genetic material from two individuals to create a new offspring. In symbolic regression, crossover involves selecting two individuals from the population and swapping sub-trees between them at a randomly selected node. This can produce offspring with combinations of the parents' features and can be an effective way to combine beneficial features from different individuals. Crossover can be used to explore new regions of the search space and improve the overall quality of the population. Recombination, or reproduction, is a genetic operator that duplicates an individual's genetic material to create an offspring that is identical to the parent. Recombination is a crucial operator as it helps maintain diversity in the population and ensures that good solutions are not lost from generation to generation. Overall, these genetic operators in symbolic regression allow the population to explore and evolve different mathematical expressions that fit the given dataset. By using a combination of mutation, crossover, and reproduction, the population can converge towards better solutions over time. Let's say we have two parent trees in our population: Parent 1: $x^2 + 3x - 4$ Parent 2: $\sin(2x) + 3$ Mutation: In the mutation operation, a random subtree in one of the parents is replaced with a new subtree. For example, we could mutate Parent 1 by replacing the constant term -4 with a new subtree, resulting in the following child: Child (mutation): x^2 + 3x + (cos(2x) + 1) Crossover: In the crossover operation, two parent trees are selected, and a random subtree from one parent is swapped with a corresponding subtree in the other parent. For example, we could perform crossover on Parent 1 and Parent 2 by swapping the constant term in Parent 1 with the sine function in Parent 2, resulting in the following child: Child (crossover): sin(2x) + 3x - 4 Recombination: In the recombination operation, a selected parent tree is duplicated and added to the next generation without any changes. For example, we could perform recombination on Parent 2, resulting in the following child: Child (recombination): sin(2x) + 3 These genetic operations are applied repeatedly to the population of trees, generating new child trees that are evaluated for their fitness, or how well they solve the problem at hand. The goal of symbolic regression is to evolve a population of trees that can accurately model a given dataset, and these genetic operations help to create variation and explore the space of possible solutions.

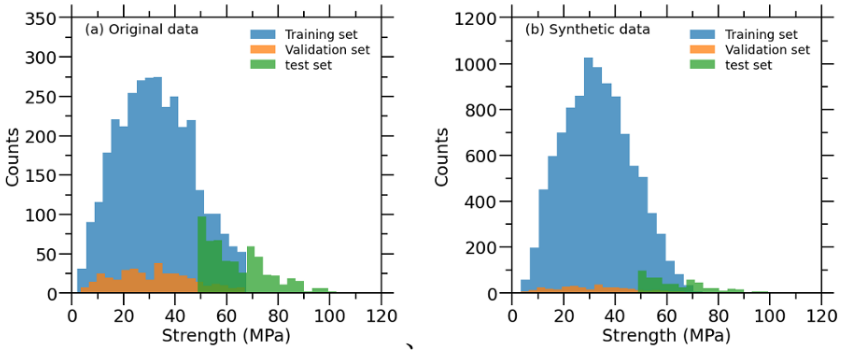

Figure 6.5: Tree structure chromosome representation of cross over operation in GP. (a) parent: $\sin(2x) + 3$; (b) child of genetic crossover operation: $x^2 + 3x + (\cos(2x) + 1)$

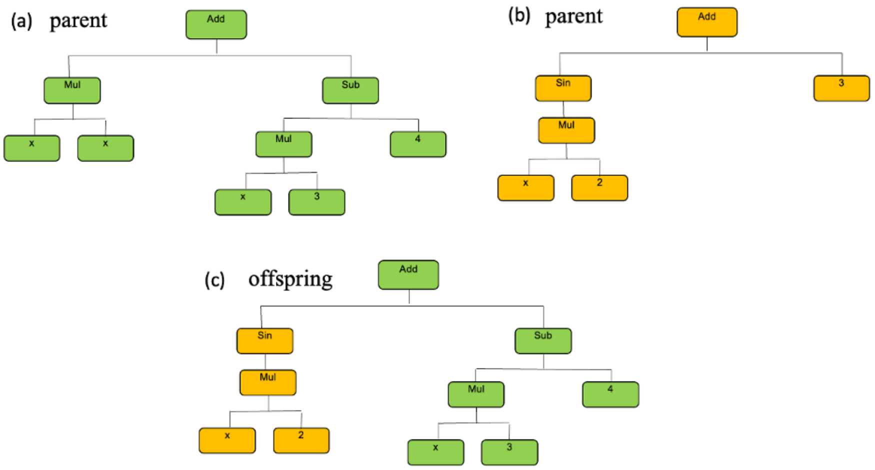

Figure 6.6: Tree structure chromosome representation of computer programs in GP. (a) parent 1: $x^2 + 3x - 4$; (b) parent 2: $\sin(2x) + 3$; (c) child of subtree mutation operation: $\sin(2x) + 3x - 4$

In this work, we employed a variant of symbolic regression algorithm named multigene symbolic regression based on GPTIPS2, an open source MATLAB toolbox for performing genetic programming and symbolic regression . This approach builds symbolic model through a linear combination of nonlinear tree expressions, each tree can be treated as a gene in this process. The coefficients of the symbolic regression model can be computed using ordinary lest square techniques, similar to the parameter optimization process for linear regression. Through combining the power of linear regression and genetic programming, the multigene symbolic regression is shown to be more accurate and computationally efficient than standard symbolic regression .

Here we adopt multigene symbolic regression to predict concrete strength at different age using Initially symbolic regression build a random generation of mathematical formulas. These mathematical formulas will compete and evolve to model the experimental data by evaluating the model accuracy and complexity. We select the optimal model by plotting the Pareto front of model accuracy and complexity for each model.

6.7 Gaussian Process Regression Model

Gaussian Process Regression (GPR) is a nonparametric, probabilistic machine learning approach that models the nonlinear relationship between input and output using a multivariate Gaussian process. Unlike other machine learning algorithms that learn a function with fixed parameters, GPR learns the probability distribution of functions that fit the training data. In GPR, data samples are assumed to follow a Gaussian distribution, which can be specified using a mean function, $m(x)$, and a covariance function, $k(x, x')$, as described by Rasmussen and Williams (2006):

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \quad \text{(Eq. 1)} \tag{6.1}$$

By selecting the mean (often zero) and covariance functions (kernel), prior knowledge about the space of functions can be incorporated. Gaussian noise, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, can also be added to the labels by summing the label distribution and noise distribution, as suggested by Rasmussen and Nickisch (2010):

$$y \sim \mathcal{GP}(m(x), k(x, x') + \sigma_{ij}\sigma^2) \quad \text{(Eq. 2)} \tag{6.2}$$

Given training data points and test points, GPR aims to learn the distribution of both as a multivariate Gaussian distribution. The predictive distribution about the target variable is derived using conditional probability, which measures the likelihood of the prediction given the training data. From the predictive distribution, the mean value is extracted as the prediction value, and the standard deviation is used to infer the uncertainty of the prediction.

6.8 Empirical Concrete Strength Prediction Model

Abrams' law, a pioneering empirical model introduced in 1918, elucidates the relationship between the water-cement ratio and concrete's compressive strength (Kargari et al., 2019).

This relationship is represented by:

$$f_c = AB^{-X} \quad \text{(Eq. 3)} \tag{6.3}$$

Here, f_c represents the compressive strength, while A and B are empirical constants. X signifies the water-cement ratio. For enhanced accuracy, especially considering the impact of mineral admixtures on concrete strength, the water-binder ratio is recommended. This ratio includes water content, cement content, fly ash content, granulated blast furnace slag content, and an efficiency factor, defined as:

$$x = \frac{w}{c + kf + s} \quad \text{(Eq. 4)} \tag{6.4}$$

In this equation, w stands for water content, c for cement content, f for fly ash content, s for granulated blast furnace slag content, and k represents the efficiency factor.

Beyond Abrams' law, there are other empirical models in concrete technology, such as the linear law (Moutassem and Chidiac, 2016) and the power law (Powers and Brownyard, 1946). They are represented as:

$$f_c = A\left(\frac{w}{c}\right)^{-B} \quad \text{(Eq. 5)} \tag{6.5}$$

$$f_c = A\frac{c}{w} + B \quad \text{(Eq. 6)} \tag{6.6}$$

In both equations, f_c is the compressive strength, and A and B are empirical constants. To adapt these empirical models to varying ages, it's often postulated that strength is proportionate to the logarithm of the concrete age (Mannan et al., 2002).

6.9 Empirical Concrete Strength Prediction Model

Abrams' law, a pioneering empirical model introduced in 1918, elucidates the relationship between the water-cement ratio and concrete's compressive strength (Kargari et al., 2019). This relationship is represented by:

$$f_c = AB^{-X} \quad \text{(Eq. 3)}$$

Here, f_c represents the compressive strength, while A and B are empirical constants. X signifies the water-cement ratio. For enhanced accuracy, especially considering the impact of mineral admixtures on concrete strength, the water-binder ratio is recommended. This ratio includes water content, cement content, fly ash content, granulated blast furnace slag content, and an efficiency factor, defined as:

$$x = \frac{w}{c + kf + s} \quad \text{(Eq. 4)}$$

In this equation, w stands for water content, c for cement content, f for fly ash content, s for granulated blast furnace slag content, and k represents the efficiency factor.

Beyond Abrams' law, other empirical models in concrete technology include the linear law (Moutassem and Chidiac, 2016) and the power law (Powers and Brownyard, 1946), represented as:

$$f_c = A\left(\frac{w}{c}\right)^{-B} \quad \text{(Eq. 5)}$$

$$f_c = A\left(\frac{c}{w}\right) + B \quad \text{(Eq. 6)}$$

In both equations, f_c represents the compressive strength, and A and B are empirical constants. To adapt these empirical models for varying ages, it's often postulated that strength is proportional to the logarithm of the concrete age (Mannan et al., 2002).

6.10 Machine learning model optimization

To determine the optimal configuration of the machine learning model, we employed a stratified sampling approach to split the training set into five folds (Pedregosa et al., 2011). This allowed us to evaluate the model's performance on different subsets of the training data. We then performed grid search to identify the optimal model hyperparameters, considering the averaged model performance across the five folds. For the symbolic regression model, the following parameters were optimized: population size (ranging from 10 to 300; selected as 100), number of generations (ranging from 10 to 100; selected as 30), tournament size (ranging from 5 to 50; selected as 20), and maximum number of genes (ranging from 4 to 12; selected as 6). These parameter ranges were chosen based on their relevance to the symbolic regression algorithm and their impact on model performance. For the Gaussian Process Regression (GPR) model, the optimized parameters included the kernel function (specific choice depends on the specific implementation and requirements of the problem), noise level (ranging from 0.1 to 0.6; selected as 0.2), and the epoch number (specific value depends on the training process and convergence criteria). These parameters were tuned to enhance the model's performance in concrete strength prediction. During both model training and testing, we assessed the machine learning model's performance using the coefficient of determination (R^2) for strength prediction and mean squared error (MSE). These evaluation metrics provide valuable insights into the accuracy and precision of the models in predicting concrete strength. For more detailed information on machine learning model optimization, we refer readers to our previous studies (Ouyang et al., 2020; Song et al., 2022), where we discuss the optimization process and provide in-depth explanations of the methodology employed. By optimizing the machine learning model's hyperparameters, we aimed to enhance its performance and ensure that it can effectively capture the complex relationship between concrete composition and strength. The optimized models were then used for further analysis and comparison with other approaches in the study.

6.11 Concrete strength interpolation

In this section, we compare the performance of the symbolic regression model on both original and synthetic data for concrete strength interpolation. We synthesized a total of 4,286 training data points using the data augmentation method described in section 2.2. The results of our analysis, illustrated in Figure 6.7, provide valuable insights into the model's performance. The symbolic regression model exhibited slightly better performance when trained on the synthetic data, achieving a mean squared error (MSE) of 48.09, compared to the model trained on the original data set, which had an MSE of 52.75. This improvement suggests that the data augmentation method effectively enhanced the model's ability to accurately interpolate concrete strength within the known data range.

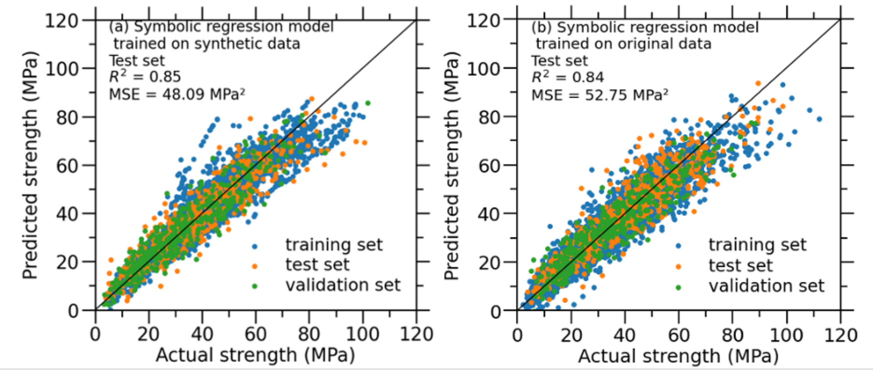

Figure 6.7: Comparison between predicted vs. ground-truth strengths for symbolic regression models trained on 4300 synthetic data (a) original data(b) for concrete strength interpolation.)

Furthermore, we aimed to determine the optimal number of synthetic data points by evaluating the model's performance on the test set using varying numbers of synthetic data points. As shown in Figure 6, the coefficient of determination (R^2) of the test set reached a plateau at an R^2 value of 0.88 when the number of synthetic data points increased from 4,286 to 12,000. This analysis allowed us to identify that employing 12,000 synthetic data points as the training set for concrete strength interpolation would yield optimal performance. These findings highlight the effectiveness of the data augmentation method in improving the performance of the symbolic regression model for concrete strength interpolation. The slight improvement in model performance on the synthetic data suggests that the augmented samples successfully captured the underlying patterns and relationships in the data, leading to more accurate predictions within the known data range. By utilizing an optimal number of synthetic data points for training (12,000 data points), we ensured that the symbolic regression model was better equipped to handle concrete strength interpolation. The inclusion of these synthetic data points provided the model with a broader range of samples, allowing it to capture the variations and complexities present in the concrete strength-composition relationship and ultimately enhancing the accuracy of the interpolation predictions.

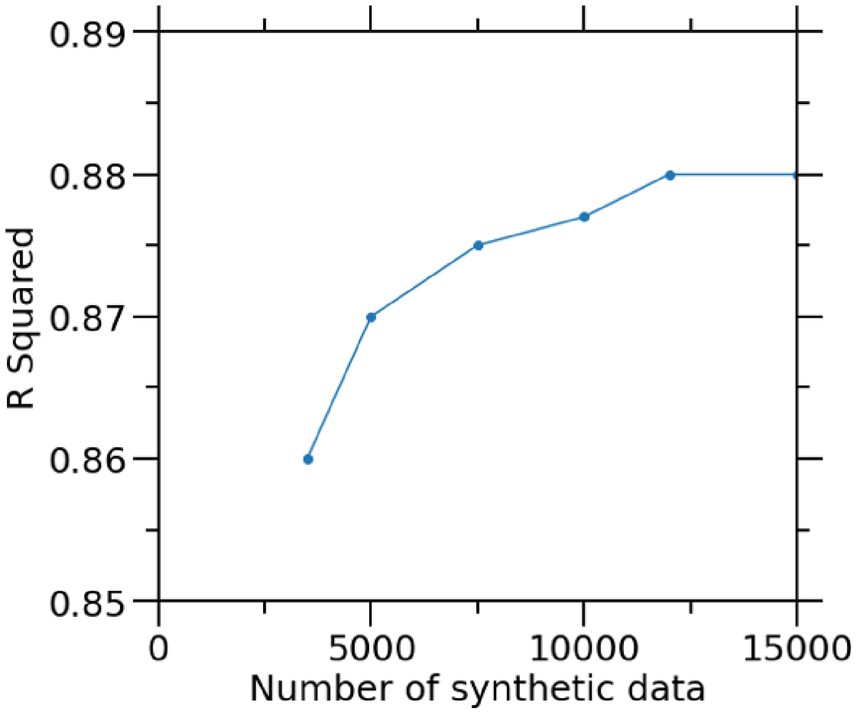

Figure 6.8: R squared of the test set as function of number of synthetic data)

After identifying the optimal number of data points for training the symbolic regression model for concrete strength interpolation, we proceeded to evaluate its performance and compare it with Gaussian Process Regression (GPR) models. Figure 6.9 presents the predicted versus measured strengths for the training, validation, and test sets, specifically focusing on the test set, for each model trained on synthetic and original data. The GPR models demonstrated good performance on the test set, achieving R^2 scores above 0.94 and a mean squared error (MSE) of 22.46 MPa

2

. The GPR models exhibited superior interpolation performance compared to the symbolic regression model due to the ML models' enhanced ability to identify patterns within the data as probabilistic models. However, it is worth noting that the GPR models are considerably more complex than the symbolic regression model. GPR models require the specification and tuning of hyperparameters such as length scales and noise levels, which are associated with the kernel functions. Additionally, GPR models involve intensive matrix operations and computations, particularly as the dataset size increases, resulting in a computational complexity of $O(n^3)$. On the other hand, the symbolic regression model maintains a simpler structure and parameterization. It searches for the optimal mathematical equation that best represents the relationship between the input variables and the output variable. The identified optimal model for symbolic regression only consists of 9 parameters, contributing to its simplicity. The simplicity and transparency of the symbolic regression model enable it to be highly interpretable and provide insights into concrete strength. Its simple and transparent form facilitates easy interpretation and understanding, offering valuable insights into concrete properties.

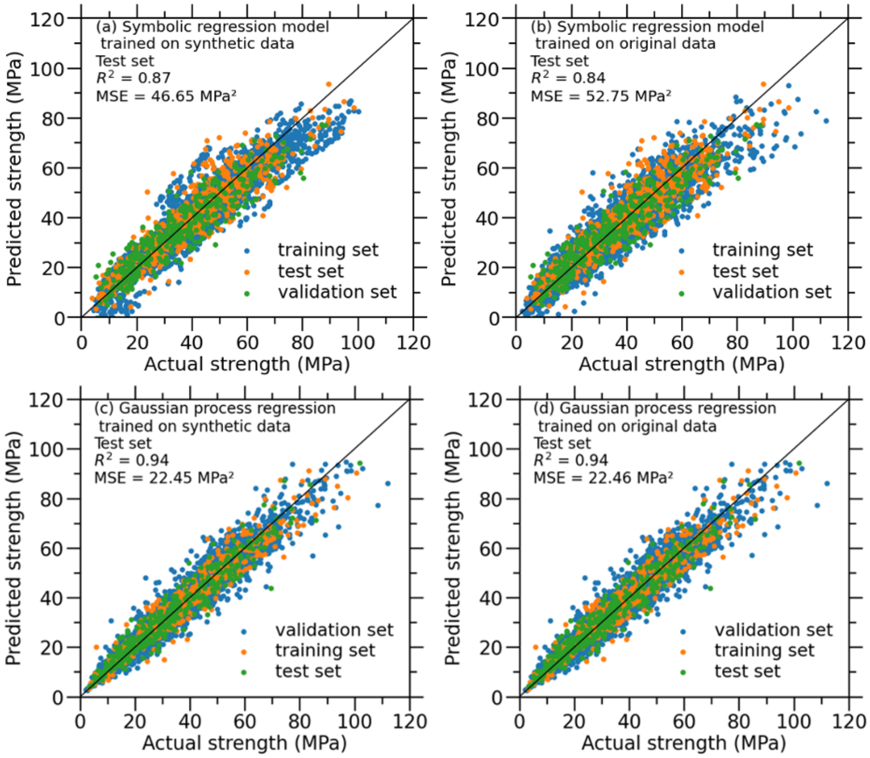

Figure 6.9: Predicted vs. ground-truth strengths comparison for symbolic regression and gaussian process regression on concrete strength interpolation with synthetic (12,000 data points) and original Data (4300 data points)

Then we compare symbolic regression model with empirical models for concrete strength interpolation, we now continue to compare symbolic regression model with empirical models and show the predicted vs. measured strengths for the training, validation and test sets in figure 6.10. Overall, symbolic regression achieved the best performance with minimum RMES, maximum R^2 and minimum confidence interval. In this evaluation approach, the data is randomly split into training, validation, and test set as shown in figure 1. Thus, the model can better interpolate the test set data which lies in the same strength domain of training set.

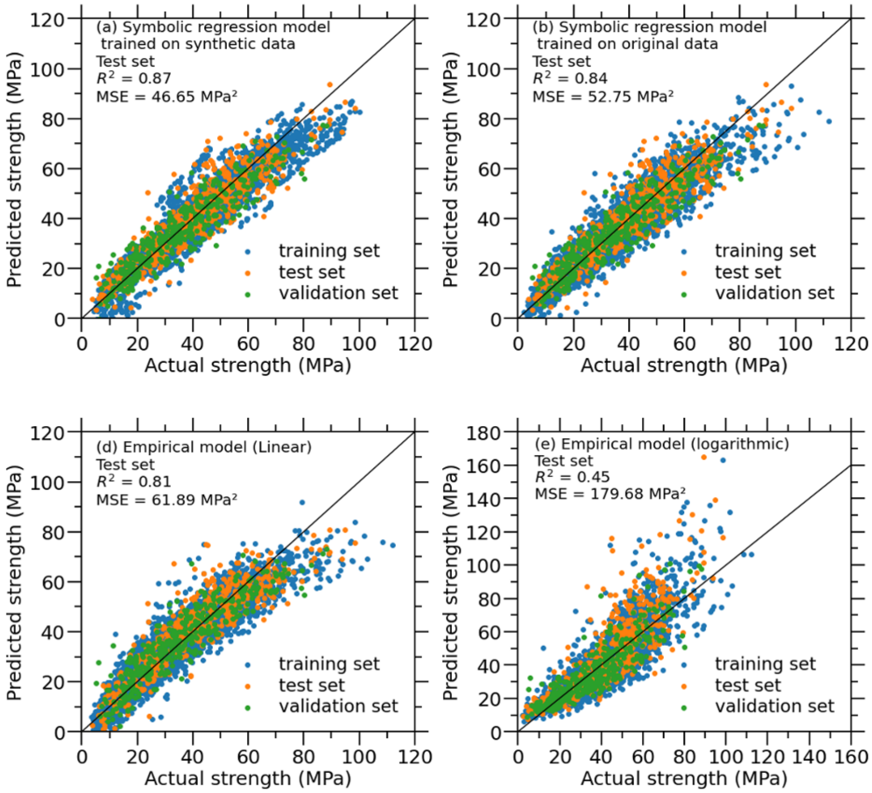

Figure 6.10: Comparison between predicted vs. ground-truth strengths for symbolic regression models and empirical models on concrete strength interpolation with synthetic (12,000 data points) and original Data (4300 data points)

. The generated mathematical equations for symbolic regression are presented in the density map shown in Figure 11. These equations were evaluated based on their coefficient of determination (R^2) on the validation set and their model complexity. The optimal models are those that lie on the Pareto front, striking a balance between high accuracy and low complexity. Figure 6.11 illustrates the density map of the generated symbolic equations, indicating the associated accuracy and complexity of the validation set model for concrete strength interpolation. It is evident from the figure that the model accuracy initially improves with increasing model complexity but eventually reaches a plateau, suggesting diminishing returns in accuracy as complexity increases. From the Pareto front, we selected five mathematical equations represented by points A, B, C, D, and E, as listed in Table 6.1. Among these equations, the one at point E was identified as the optimal model, achieving the highest R^2 value for the validation set. These findings demonstrate the analysis of the generated mathematical equations using symbolic regression. By comparing the R^2values and model complexity, we identified the optimal model from the Pareto front. The selected equation at point E exhibited the highest R^2 value for the validation set, showcasing its accuracy in predicting concrete strength.

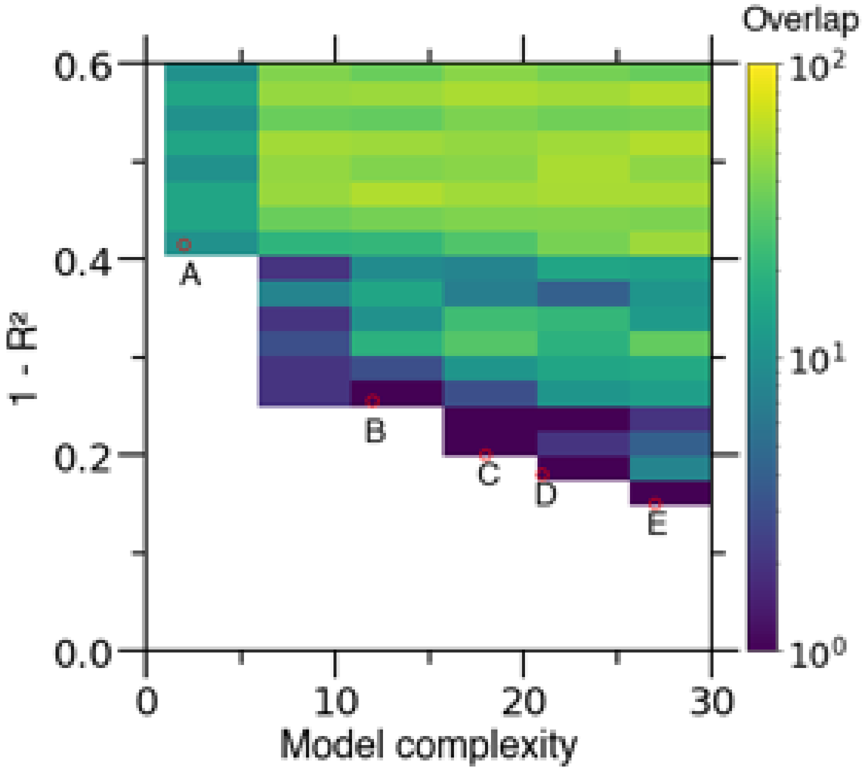

Figure 6.11: Density map of generated symbolic equations with associated accuracy and complexity of validation set model on concrete strength interpolation

Table 6.1: Selected mathematical equations from the Pareto front.

Point	**Formulas**	R^2
A	$0.43T + 22.4$	0.57
B	$5.01T^{1/2} - 94.1\frac{W}{CM} - 0.0762FLA + 60.8$	0.74
C	$11.1T^{1/2} - 0.646T - 151.0\frac{W}{CM} - 0.0876FLA - 73.1(\frac{W}{CM})^3 + 69.2$	0.80
D	$0.01125CE - 0.07686FLA - 0.006911SL + 10.34\log(T) - 102.4(\frac{W}{CM}) + 11.1T^{1/2} + 58.3$	0.81
E	$3.123(\frac{W}{CM})^{-1} \cdot \log(T) + 3.621\log(Ce) + 0.04419CA + 0.06289FA - 0.01359FLA + 0.0348SL - 149.7$	0.85

6.12 Concrete strength Extrapolation

During the evaluation of the symbolic regression model for extrapolating concrete strength, we initially compared its performance using an equal number of training data points. The model was trained and optimized using low strength concrete data and then tested on concrete samples with higher strength. The symbolic regression model exhibited poor performance in terms of the coefficient of determination (R^2) for both the original and synthetic data sets when extrapolating concrete strength. The R^2 value for the original data was only 0.07, indicating a weak correlation, and the mean squared error (MSE) was 146.46 MPa

2

. This difficulty arises because the model needs to extrapolate into the high strength region using knowledge derived from low strength concrete, presenting a significant challenge. However, we made an interesting discovery that synthetic data can enhance the performance of concrete strength extrapolation. By augmenting the original data set, the R^2 value increased from 0.07 to 0.22, and the MSE decreased from 146.46 to 121.72. The symbolic regression model trained on the synthetic data exhibited slightly better performance compared to the model trained on the original data, achieving an MSE of 48.09. This improvement suggests that the data augmentation method effectively enhances the model's ability to accurately extrapolate concrete strength into the unknown data range. Additionally, we aimed to determine the optimal number of synthetic data points by evaluating the model's performance on the test set using varying numbers of synthetic data points. Figure 6.13 illustrates the results, indicating that the coefficient of determination (R^2) of the test set reached a plateau at an R^2 value of 0.4 as the number of synthetic data points increased from 4300 to 11,000. Based on this analysis, we identified that utilizing 11,000 synthetic data points as the training set for concrete strength extrapolation would yield optimal performance.

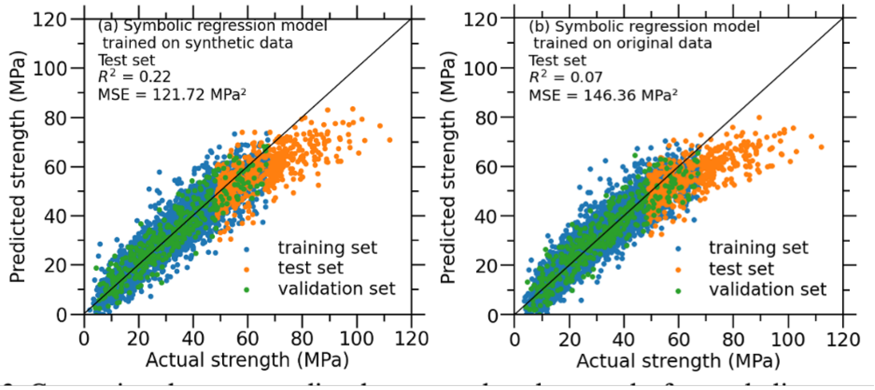

Figure 6.12: Comparison between predicted vs. ground-truth strengths for symbolic regression models trained on 4300 synthetic data (a) original data(b) for concrete strength extrapolation.

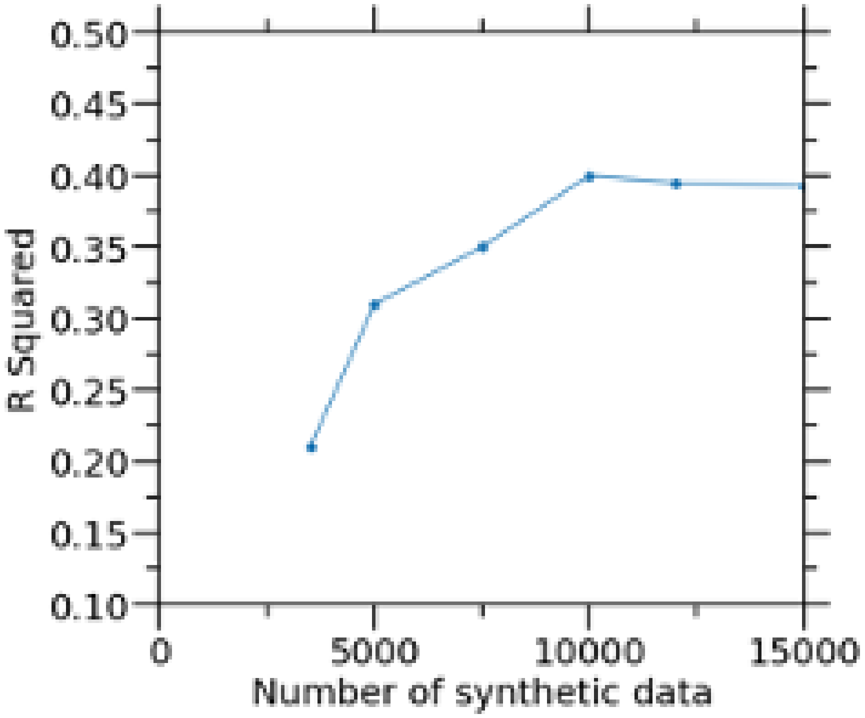

Figure 6.13: R squared of the test set as function of number of synthetic data

Figure 6.14 shows the density map of generated equations with associated accuracy and complexity during symbolic regression model optimization process. We select six equations from Pareto Front and show their formula in table 2. Similar to the optimization process of concrete strength interpolation, the symbolic regression model achieves improved performance initially with increasing complexity, but its performance doesn't get improved when the model complexity reaches twenty-two. The optimal model corresponds to the model on point F with highest R^2 of 0.88 and complexity of twenty-two. Having identified the optimal synthetic data points, we proceeded to evaluate their extrapolation performance using a test set consisting of higher strength concrete. The performance was compared with that of the Gaussian Process Regression (GPR) model. As shown in Figure 6.14, both the GPR model and symbolic regression models exhibited poorer performance compared to the case of concrete strength interpolation. Without the aid of synthetic data, the GPR model achieved an R^2 of -0.13 and a mean squared error (MSE) of 176.64 MPa^2. However, when trained on synthetic data, its performance improved significantly, with the R^2 increasing to 0.24 and the MSE decreasing to 119.70 MPa^2. In contrast, the symbolic regression model consistently outperformed the GPR model, both on synthetic and original data. Notably, when trained on synthetic data, the R^2 of the test set increased to 0.40. Furthermore, the symbolic regression model demonstrated its superiority by achieving these improved results while maintaining simplicity. Its ability to search for the optimal mathematical form to fit the data and simultaneously suppress complexity played a crucial role in its superior performance. This finding supports the notion that a simple model with an appropriate structure can exhibit better extrapolation capabilities in unknown regions compared to an overly complex, black-box model.

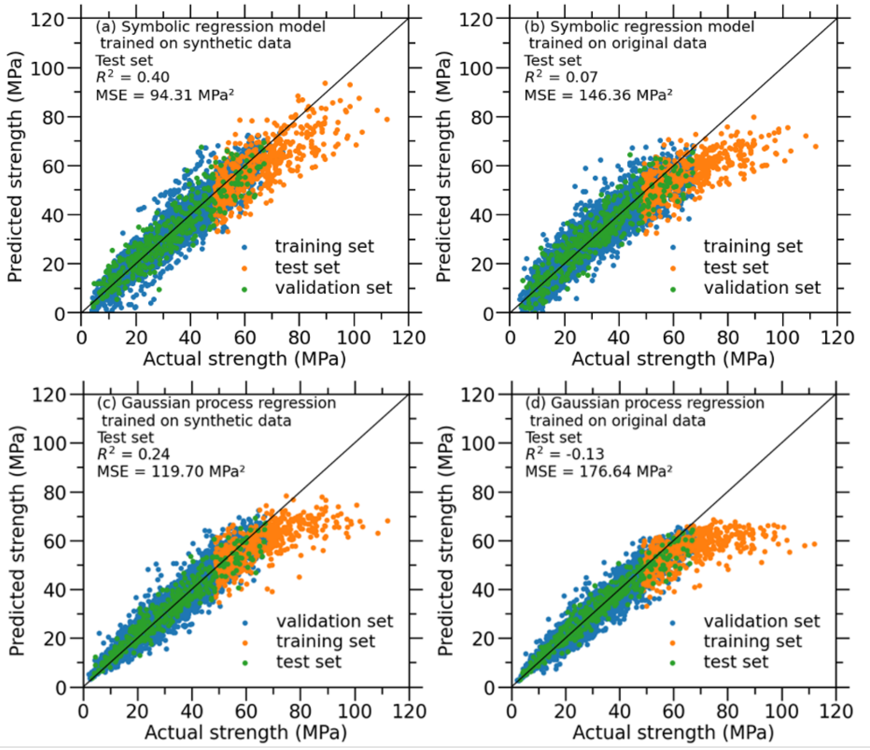

Figure 6.14: Comparison between predicted vs. ground-truth strengths for symbolic regression models and Gaussian process regression models on concrete strength extrapolation with synthetic (11,000 data points) and original Data (4300 data points).

Figure 6.15 shows for each model the predicted vs. measured strengths for the training, validation and test sets. First it can be noted that all the models show worse performance compared to the case of concrete strength interpolation. It is because that the concrete data (training set) used to train the models are not in the strength domain as the data (test set) used to evaluate the model, and it will be challenging for the models to predict data that is not about training data. Compared to empirical model, symbolic regression again outperforms them with minimum RMES, maximum R^2 and minimum confidence interval for the test set. This also illustrates that empirical model is not able to find the pattern between the concrete strength and composition both in the case of concrete strength interpolation and extrapolation.

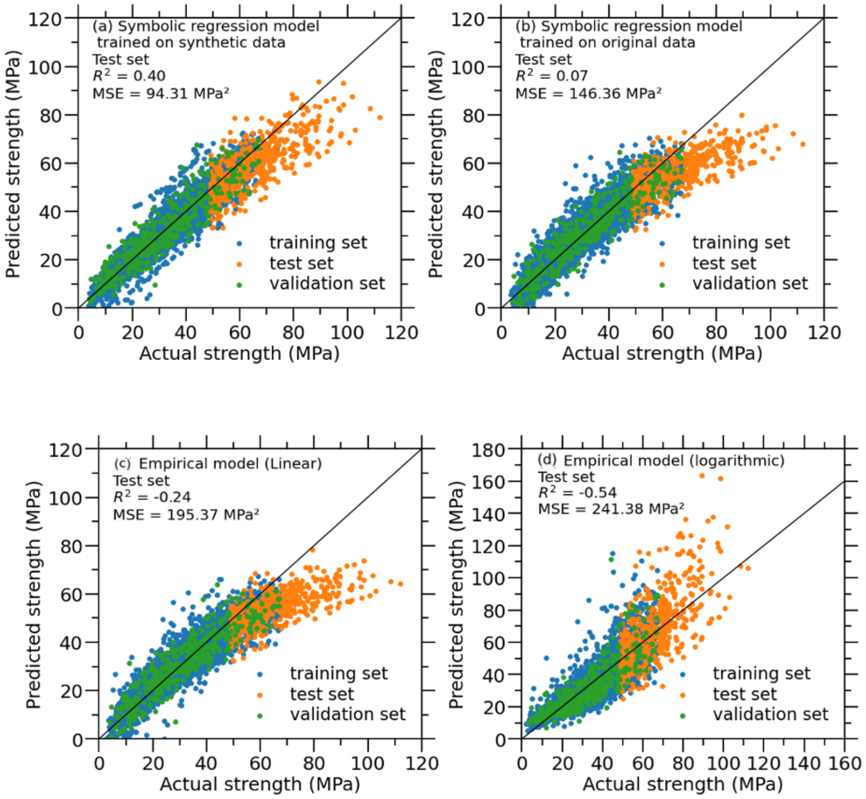

Figure 6.15: Comparison between predicted vs. ground-truth strengths for symbolic regression models and empirical models on concrete strength extrapolation with synthetic (11,000 data points) and original Data (4300 data points.

Figure 6.16 offers a visual representation of the generated symbolic equations, highlighting the interplay between the accuracy and complexity for the task of concrete strength extrapolation. A clear observation from the map is the initial enhancement in model accuracy as its complexity rises. However, this trend seems to taper off at higher complexities, indicating a diminishing return in accuracy gains relative to the increase in complexity.

From the depicted Pareto front in Figure 16, six mathematical models have been chosen, represented by points A through F, as detailed in Table 6.2. Of these, the equation corresponding to point F has been discerned as the most optimal model, given that it registers the highest R^2 value on the validation set.

This analysis serves to emphasize the importance and utility of symbolic regression in mathematical model generation. Through a comparative assessment of the R^2 values and inherent model complexities, we were able to identify a model on the Pareto front that stands out in its predictive accuracy for concrete strength extrapolation. The equation at point F, with its superior R^2 value, best showcases this prowess in forecasting concrete strength.

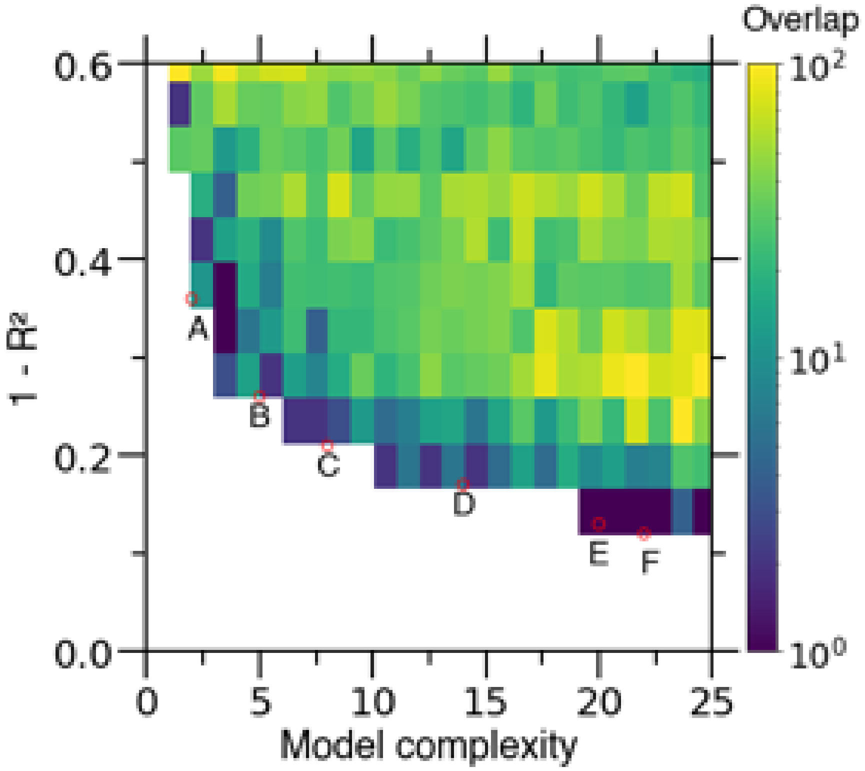

Figure 6.16: Density map of generated symbolic equations with associated accuracy and complexity of validation set model on concrete strength extrapolation.

6.13 Comparative Analysis of Empirical and Symbolic Regression Models

4.1. Abrams' Law and Its Evolution: - Abrams' Law (Eq. 3): This foundational empirical model established a negative exponential relationship between the water-cement ratio and concrete compressive strength. As the water-cement ratio increases, the compressive strength diminishes, which aligns with our understanding from the symbolic regression models like Model B and Model C. Both models have terms that indicate a decline in strength as the water-cement ratio (W/CM) increases.

- Enhanced Water-Binder Ratio (Eq. 4): Abrams' law has been augmented by considering the water-binder ratio, which accounts for other components like fly ash and slag. This expanded formula gives a broader perspective, similar to the multifactorial equations in the symbolic regression, especially Models B, C, and D which incorporate multiple components.

4.2. Other Empirical Models: Linear Law (Eq. 5) and Power Law (Eq. 6): These models offer a linear and power relationship, respectively, between the water-cement ratio and strength. Again, similar trends can be observed in the symbolic regression models. For instance, Model A presents a linear relationship, and while the exact form isn't a power law, other models do show varying rates of strength gain or loss (like the $T^{1/2}$ in Model B.

4.3 Age and Its Role: The postulate that strength is proportional to the logarithm of the concrete age finds echoes in the symbolic regression models. For example, Model D incorporates the term log(time), implying a diminishing rate of strength gain with age. This is in line with the empirical understanding.

4.4 Comparative Physical Insights: While empirical models provide a broad overview and are grounded in experimental observations, symbolic regression models appear more versatile. They're capable of capturing intricate interplays between multiple factors simultaneously. For instance, while empirical models like Abrams' Law focus predominantly on the water-cement or water-binder ratio, symbolic regression models incorporate terms like log(Ce*T) (as in Model E and F), highlighting interactions not just between components but also over time.

Additionally, the symbolic regression models offer equations that provide insights into how specific components like fly ash, slag, cement, and time individually and he artificial neural network modelscollectively impact the strength. Empirical models, in contrast, tend to be more generalized.

Abrams' law and subsequent empirical models laid the foundation for understanding the relationship between concrete composition and strength. They are simple, easy to understand, and have stood the test of time. However, the symbolic regression models present a more comprehensive and nuanced approach. By capturing multifaceted interactions and considering multiple components simultaneously, they offer a deeper, more detailed insight into the science of concrete strength. In practical applications, the choice between empirical and symbolic regression models would hinge on the specific requirements: simplicity and broad generalizations versus detailed, multifactorial analysis.

Table 6.2: The six mathematical equations at the pareto front.

Point	**Formulas**	R^2
A	$0.41T + 24$	0.64
B	$0.00139CE + 0.00984FA - 0.0757FLA - 0.00984SL - 0.626T + 10.8T^{1/2} - 148.0(W/CM)^{1/2} + 98.8$	0.74
C	$162.0\exp(-1.0W/CM) - 0.633\text{time} - 0.0826\text{fly ash} + 10.9\text{time}^{1/2} - 94.7$	0.79
D	$0.01125\text{cement} - 0.07686\text{flyash} - 0.006911\text{slag} + 10.34\log(\text{time}) - 102.4(W/CM) + 58.3$	0.83
E	$3.207(W/CM)^{-1}\cdot\log(Ce\cdot T) + 3.406\log(Ce\cdot T) + 0.04869CA + 0.07005FA + 0.03467SL - 149.7$	0.84
F	$3.123(W/CM)^{-1}\cdot\log(Ce\cdot T) + 3.621\log(Ce\cdot T) + 0.04419CA + 0.06289FA - 0.01359FLA + 0.0348SL - 149.7$	0.88

6.14 Discussion

Training symbolic regression on synthetic data generated by SMOTE for the features and GPR for the targets can offer several advantages over other techniques in certain contexts. One advantage of symbolic regression over other machine learning techniques is its strong constraint of a symbolic nature. By discovering simple, interpretable mathematical relationships between the input features and the target output variable, symbolic regression models can capture the underlying physical knowledge that may exist in the data. Additionally, the use of SMOTE to generate the features and GPR to generate the targets helps ensure that the synthetic data is representative of the real-world data and captures the underlying relationships between the input features and the target variable. This approach can effectively address issues related to data scarcity or class imbalance, allowing for more comprehensive training and evaluation of the symbolic regression models. Moreover, symbolic regression models trained on synthetic data generated by SMOTE for the features and GPR for the targets can be more robust to extrapolation. By limiting the potential for interpolation and capturing the underlying mathematical relationships in the data, these models can make more reliable predictions for input values that lie outside the range of the known data. Symbolic regression models also exhibit high transferability, as they learn simple, interpretable mathematical relationships that can be easily applied to other domains or scenarios. The physical knowledge extracted from the data can provide valuable insights into the underlying relationships between the input features and the target variable, guiding the design of new experiments or applications. By incorporating synthetic data, symbolic regression models can filter out noise and improve their performance. SMOTE, as a non-parametric algorithm, leverages the local structure of the data to generate synthetic samples, helping to reduce the impact of noise and enhance the model's accuracy.

Overall, training symbolic regression models on synthetic data generated by KNN for the features and ANN for the targets can offer several advantages for machine learning applications, including their strong constraint of symbolic nature, robustness to extrapolation, transferability, and ability to address issues related to data scarcity or class imbalance. However, it's important to carefully evaluate the performance of the symbolic regression models on real-world data and test their generalization capabilities in a variety of settings.

6.15 Conclusions

In this study, we utilized the symbolic regression model to search for a mathematical equation that can predict concrete strength based on its composition. We evaluated the performance of the symbolic regression model for both concrete strength interpolation and extrapolation tasks and compared it with empirical and ML models. Our findings demonstrate that the symbolic regression model can construct simple and transparent mathematical equations that establish the relationship between concrete composition and strength. Although the accuracy of the symbolic regression model may not match that of ML models in terms of concrete strength interpolation, it offers the advantage of lower complexity and greater interpretability. The symbolic regression model excels particularly in concrete strength extrapolation, outperforming both empirical and ML models. The significance of this research lies in the ability of the symbolic regression model to provide interpretable mathematical equations that capture the complex relationship between concrete composition and strength. These equations offer valuable insights into the behavior of concrete and can guide concrete mixture design. Additionally, the superior performance of the symbolic regression model in concrete strength extrapolation highlights its potential for applications where accurate predictions beyond the known data range are crucial. Overall, the results of this study demonstrate the advantages of symbolic regression in terms of simplicity, interpretability, and extrapolation performance in predicting concrete strength. By combining the strengths of symbolic regression with the insights gained from the derived equations, we can enhance our understanding of concrete science and facilitate optimized concrete mixture design.

CHAPTER 7

Summary of Contributions and Future Directions

7.1 Summary of Contributions

This thesis has made significant strides in advancing the field of concrete strength prediction through the application and optimization of various machine learning models. Each chapter has contributed distinct insights and innovative methodologies, leading to a comprehensive understanding of the intricate relationship between concrete composition and strength. The major contributions of this research are:

1.Enhanced Outlier Detection with EBOD: Chapter 3 introduced the Ensemble-Based Outlier Detection (EBOD) method, which significantly improves the learning efficiency of Artificial Neural Networks (ANN). EBOD optimizes the cleansing of data, enabling the ANN model to operate with fewer hidden neurons and datapoints while improving accuracy, as evidenced by various performance metrics.

2.Machine Learning Models for Concrete Strength Prediction: Chapter 4 explored different machine learning models, highlighting the trade-offs between model complexity and data requirements. It was found that simpler models like Polynomial Regression (PR) quickly reach their maximum accuracy with smaller datasets, while more complex models like Random Forest (RF) require more data but can achieve higher accuracy.

3. Optimization of Gaussian Process Regression (GPR) Model: In Chapter 5, we optimized the GPR model by integrating physical knowledge of concrete strength development. This model not only showed unprecedented accuracy compared to other ML models but also

effectively captured the uncertainties in the concrete dataset, providing valuable insights for concrete mixture design.

4. Symbolic Regression for Concrete Strength Prediction: Chapter 6 introduced the use of symbolic regression to derive transparent mathematical equations representing the relationship between concrete composition and strength. The symbolic regression model excelled in extrapolation tasks, outperforming empirical and traditional ML models, offering a powerful tool for understanding concrete behavior and guiding mixture design.

7.2 Future Directions

While this research has made substantial contributions, there are several avenues for future work that can further enhance the field of concrete science and machine learning:

1. Natural Language Processing (NLP) for Concrete Data Extraction: The manual collection and organization of concrete data are immensely time-consuming and error-prone due to the diversity in reporting formats and the lack of completeness in many cases. To address this, I plan to employ NLP techniques to automate the extraction of material data from vast quantities of documents. This approach will not only streamline data collection but also ensure the creation of extensive, high-quality datasets that can significantly bolster the capabilities of ML models in concrete science.

2. Mining Process-Structure-Property-Performance Relationships: NLP will also be utilized to mine the vast literature for insights into the relationships between process, structure, property, and performance in concrete materials. This novel approach has the potential to uncover new materials and identify optimal synthesis procedures, thereby pushing the boundaries of concrete science.

3.Integration of NLP and ML for Comprehensive Model Development: The integration of NLP-extracted datasets with advanced ML models promises a new horizon in concrete science. By leveraging the structured data obtained through NLP, ML models can be trained more effectively, leading to more accurate predictions and a deeper understanding of the complex interplay between various factors influencing concrete strength.

In conclusion, the contributions of this thesis lay a solid foundation for future research, and the proposed future work aims to not only extend the frontiers of concrete science but also harness the synergy between different domains of computational science to foster innovation and advancement in material science research.

Bibliography

[1] American National Standards Institute. C618-08a: Standard Specification for Coal Fly Ash and Raw or Calcined Natural Pozzolan for Use in Concrete, 2008.

[2] American Society for Testing and Materials. *Standard test method for compressive strength of cylindrical concrete specimens*, 2003.

[3] American Society for Testing and Materials. C150/C150M-17, Standard Specification for Portland Cement, 2017.

[4] S. A. Ashour and F. F. Wafa. Flexural behavior of high-strength fiber reinforced concrete beams. *Struct. J.*, 90(3):279–287, 1993.

[5] C. Başyigit, I. Akkurt, S. Kilincarslan, and A. Beycioglu. Prediction of compressive strength of heavyweight concrete by ann and fl models. *Neural Computing and Applications*, 19:507–513, 2010.

[6] J. J. Biernacki, J. W. Bullard, G. Sant, K. Brown, F. P. Glasser, S. Jones, T. Ley, R. Livingston, L. Nicoleau, J. Olek, F. Sanchez, R. Shahsavari, P. E. Stutzman, K. Sobolev, and T. Prater. Cements in the 21st century: Challenges, perspectives, and opportunities. *J. Am. Ceram. Soc.*, 100(7):2746–2773, 2017.

[7] L. Billard and E. Diday. Symbolic regression analysis. In *Classification, Clustering, and Data Analysis, Studies in Classification, Data Analysis, and Knowledge Organization*, pages 281–288. Springer, Berlin, Heidelberg, 2002.

[8] S. Bishnoi, S. Singh, R. Ravinder, M. Bauchy, N. N. Gosvami, H. Kodamana, and N. M. A. Krishnan. Predicting young's modulus of oxide glasses with sparse datasets using machine learning. *J. Non-Cryst. Solids.*, 524:119643, 2019.

[9] N. Bouzoubaâ and M. Lachemi. Self-compacting concrete incorporating high volumes of class f fly ash: Preliminary results. *Cement and Concrete Research*, 31:413–420, 2001.

[10] L. E. Burris, P. Alapati, R. D. Moser, M. T. Ley, N. Berke, and K. E. Kurtis. Alternative cementitious materials: Challenges and opportunities. In *International Workshop on Durability and Sustainability of Concrete Structures*, pages '27.1–27.10', Bologna, 2015. ACI Publishers.

[11] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. Smote: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[12] M.-Y. Cheng, C.-C. Huang, and A. Roy. Predicting project success in construction using an evolutionary gaussian process inference model. *Journal of Civil Engineering and Management*, 19:S202–S211, 2013.

[13] P. Chopra, R. Sharma, M. Kumar, and T. Chopra. Comparison of machine learning techniques for the prediction of compressive strength of concrete. *Advances in Civil Engineering*, 2018:5481705, 2018.

[14] M. DeRousseau, J. Kasprzyk, and W. S. III. Computational design optimization of concrete mixtures: A review. *Cement and Concrete Research*, 109:42–53, 2018.

[15] M. A. DeRousseau, J. R. Kasprzyk, and W. V. Srubar III. Computational design optimization of concrete mixtures: A review. *Cem. Concr. Res.*, 109:42–53, 2018.

[16] D. Duvenaud. The kernel cookbook: Advice on covariance functions. Online, 2014. Available at https://www.cs.toronto.edu/ duvenaud/cookbook/.

[17] A. A. Elhakam, A. E. Mohamed, and E. Awad. Influence of self-healing, mixing method and adding silica fume on mechanical properties of recycled aggregates concrete. *Constr. Build. Mater.*, 35:421–427, 2012.

[18] A. A. Elhakam, A. E. Mohamed, and E. Awad. Influence of self-healing, mixing method and adding silica fume on mechanical properties of recycled aggregates concrete. *Construction and Building Materials*, 35:421–427, 2012.

[19] H. Erdal. Two-level and hybrid ensembles of decision trees for high performance concrete compressive strength prediction. *Engineering Applications of Artificial Intelligence*, 26:1689–1697, 2013.

[20] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *arXiv preprint arXiv:1809.11165*, 2018.

[21] M. Hashemi, P. Shafigh, M. R. Karim, and C. D. Atis. The effect of coarse to fine aggregate ratio on the fresh and hardened properties of roller-compacted concrete pavement. *Construction and Building Materials*, 169:553–566, 2018.

[22] T. Hemalatha, K. R. Sundar, A. R. Murthy, and N. R. Iyer. Influence of mixing protocol on fresh and hardened properties of self-compacting concrete. *Constr. Build. Mater.*, 98:119–127, 2015.

[23] T. Hemalatha, K. R. Sundar, A. R. Murthy, and N. R. Iyer. Influence of mixing protocol on fresh and hardened properties of self-compacting concrete. *Construction and Building Materials*, 98:119–127, 2015.

[24] M. Hinchliffe, M. Willis, H. Hiden, M. Tham, B. McKay, and G. Barton. Modelling chemical process systems using a multi-gene genetic programming algorithm. In *Genetic Programming: Proceedings of the First Annual Conference (Late Breaking Papers)*, pages 56–65, 1996.

[25] F. Jäkel, B. Schölkopf, and F. Wichmann. A tutorial on kernel methods for categorization. *Journal of Mathematical Psychology*, 51:343–358, 2007.

[26] D. Karaboga, C. Ozturk, N. Karaboga, and B. Gorkemli. Artificial bee colony programming for symbolic regression. *Information Sciences*, 209:1–15, 2012.

[27] A. Kargari, H. Eskandari-Naddaf, and R. Kazemi. Effect of cement strength class on the generalization of abrams' law. *Structural Concrete*, 20:493–505, 2019.

[28] A. N. M. Krishnan, S. Mangalathu, M. M. Smedskjaer, A. Tandia, H. Burton, and M. Bauchy. Predicting the dissolution kinetics of silicate glasses using machine learning. *J. Non-Cryst. Solids.*, 487:37–45, 2018.

[29] H. Ling, C. Qian, W. Kang, C. Liang, and H. Chen. Combination of support vector machine and k-fold cross validation to predict compressive strength of concrete in marine environment. *Construction and Building Materials*, 206:355–363, 2019.

[30] F. Liu, K. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, Pisa, Italy, 2008. IEEE.

[31] H. Liu, Z. Fu, K. Yang, X. Xu, and M. Bauchy. Machine learning for glass science and engineering: A review. *Journal of Non-Crystalline Solids X*, 4:100036, 2019.

[32] H. Liu, T. Zhang, N. M. A. Krishnan, M. M. Smedskjaer, J. V. Ryan, S. Gin, and M. Bauchy. Predicting the dissolution kinetics of silicate glasses by topology-informed machine learning. *Npj Mater. Degrad.*, 3(1):1–12, 2019.

[33] A. Lübeck, A. Gastaldini, D. Barin, and H. Siqueira. Compressive strength and electrical properties of concrete with white portland cement and blast-furnace slag. *Cement and Concrete Composites*, 34:392–399, 2012.

[34] M. Mannan, H. Basri, M. Zain, and M. Islam. Effect of curing conditions on the properties of ops-concrete. *Building and Environment*, 37:1167–1171, 2002.

[35] J. Meusel and J. Rose. Production of granulated blast furnace slag at sparrows point, and the workability and strength potential of concrete incorporating the slag. *Special Publication*, 79:867–890, 1983.

[36] B. Micenková, B. McWilliams, and I. Assent. Learning representations for outlier detection on a budget. *ArXiv Preprints*, arXiv:1507.08104, 2015.

[37] H. Motulsky and R. Brown. Detecting outliers when fitting data with nonlinear regression–a new method based on robust nonlinear regression and the false discovery rate. *BMC Bioinformatics*, 7:123, 2006.

[38] J. Mourão-Miranda, D. Hardoon, T. Hahn, A. Marquand, S. Williams, J. Shawe-Taylor, and M. Brammer. Patient classification as an outlier detection problem: An application of the one-class support vector machine. *NeuroImage*, 58:793–804, 2011.

[39] F. Moutassem and S. Chidiac. Assessment of concrete compressive strength prediction models. *KSCE Journal of Civil Engineering*, 20:343–358, 2016.

[40] F. Moutassem and S. E. Chidiac. Assessment of concrete compressive strength prediction models. *KSCE J. Civ. Eng.*, 20(1):343–358, 2016.

[41] T. Naik. Sustainability of concrete construction. *Practice Periodical on Structural Design and Construction*, 13:98–103, 2008.

[42] J. Namyong, Y. Sangchun, and C. Hongbum. Prediction of compressive strength of in-situ concrete based on mixture proportions. *Journal of Asian Architecture and Building Engineering*, 3:9–16, 2004.

[43] W. Nash, T. Sellers, S. Talbot, A. Cawthorn, and W. Ford. The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. Technical Report 48, Sea Fisheries Division, 1994.

[44] R. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *Proceedings of VLDB*, pages 144–155, 1994.

[45] T. Oey, S. Jones, J. W. Bullard, and G. Sant. Machine learning can predict setting behavior and strength evolution of hydrating cement systems. *J. Am. Ceram. Soc.*, 103(1):480–490, 2020.

[46] B. Omran, Q. Chen, and R. Jin. Comparison of data mining techniques for predicting compressive strength of environmentally friendly concrete. *Journal of Computing in Civil Engineering*, 30:04016029, 2016.

[47] A. Oner and S. Akyuz. An experimental study on optimum usage of ggbs for the compressive strength of concrete. *Cement and Concrete Composites*, 29:505–514, 2007.

[48] B. Ouyang, Y. Li, F. Wu, H. Yu, Y. Wang, G. Sant, and M. Bauchy. Computational modeling – predicting concrete's strength by machine learning: Balance between accuracy and complexity of algorithms. *ACI Materials Journal*, 2020.

[49] B. Ouyang, Y. Li, F. Wu, H. Yu, Y. Wang, G. Sant, and M. Bauchy. Computational modeling – predicting concrete's strength by machine learning: Balance between accuracy and complexity of algorithms. *ACI Mater. J.*, 2020.

[50] B. Ouyang, Y. Song, Y. Li, G. Sant, and M. Bauchy. Ebod: An ensemble-based outlier detection algorithm for noisy datasets. *Knowledge-Based Systems*, 2021.

[51] A. K. Pani, K. G. Amin, and H. K. Mohanta. Data driven soft sensor of a cement mill using generalized regression neural network. In *International Conference on Data Science Engineering (ICDSE)*, pages 98–102, Cochin, 2012. IEEE Publishers.

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[53] T. Pevný. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102:275–304, 2016.

[54] R. Philleo. Increasing the usefulness of aci 214: Use of standard deviation and a technique for small sample sizes. *Concrete International*, 3:71–74, 1981.

[55] S. Popovics. Analysis of concrete strength versus water-cement ratio relationship. *Materials Journal*, 87:517–529, 1990.

[56] S. Popovics. History of a mathematical model for strength development of portland cement concrete. *Mater. J.*, 95(5):593–600, 1998.

[57] S. Popovics and J. Ujhelyi. Contribution to the concrete strength versus water-cement ratio relationship. *Journal of Materials in Civil Engineering*, 20:459–463, 2008.

[58] T. Powers and T. Brownyard. Studies of the physical properties of hardened portland cement paste. pages 101–132, 1946.

[59] T. C. Powers. Physical properties of cement paste. Skokie: Portland Cement Assoc R & D Lab Bull Publishers, 1960.

[60] A. Pradhan. Support vector machine-a survey. *International Journal of Emerging Technology and Advanced Engineering*, 2:82–85, 2012.

[61] J. L. Provis. Grand challenges in structural materials. *Front. Mater.*, 2:31, 2015.

[62] P. Purnell and L. Black. Embodied carbon dioxide in concrete: Variation with common mix design parameters. *Cem. Concr. Res.*, 42(6):874–877, 2012.

[63] P. Purnell and L. Black. Embodied carbon dioxide in concrete: Variation with common mix design parameters. *Cement and Concrete Research*, 42:874–877, 2012.

[64] X. Qiu, L. Zhang, Y. Ren, P. Suganthan, and G. Amaratunga. Ensemble deep learning for regression and time series forecasting. In *2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*, page 1–6. IEEE, 2014.

[65] M. H. Rafiei, W. H. Khushefati, R. Demirboga, and H. Adeli. Neural network, machine learning, and evolutionary approaches for concrete material characterization. *ACI Mater. J.*, 113(6):781–789, 2016.

[66] M. H. Rafiei, W. H. Khushefati, R. Demirboga, and H. Adeli. Supervised deep restricted boltzmann machine for estimation of concrete. *ACI Materials Journal*, 114, 2017.

[67] C. Rasmussen and C. Williams. Gaussian processes for machine learning. 2006.

[68] C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010.

[69] S. Rayana, W. Zhong, and L. Akoglu. Sequential ensemble learning for outlier detection: A bias-variance perspective. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, page 1167–1172. IEEE, 2016.

[70] S. K. Rejeb. Improving compressive strength of concrete by a two-step mixing method. *Cem. Concr. Res.*, 26(4):585–592, 1996.

[71] S. K. Rejeb. Improving compressive strength of concrete by a two-step mixing method. *Cement and Concrete Research*, 26:585–592, 1996.

[72] G. Rodríguez de Sensale. Strength development of concrete with rice-husk ash. *Cem. Concr. Compos.*, 28(2):158–160, 2006.

[73] D. Searson. Gptips: Genetic programming and symbolic regression for matlab, 2009.

[74] R. Siddique. Performance characteristics of high-volume class f fly ash concrete. *Cement and Concrete Research*, 34:487–493, 2004.

[75] R. Siddique. Properties of self-compacting concrete containing class f fly ash. *Materials and Design*, 32:1501–1507, 2011.

[76] Y. Song, B. Ouyang, J. Chen, X. Wang, K. Wang, S. Zhang, Y. Chen, G. Sant, and M. Bauchy. Decarbonizing concrete with artificial intelligence. In *Computational Modelling of Concrete and Concrete Structures*, pages 168–176. CRC Press, 2022.

[77] W. H. Taylor. *Concrete Technology and Practice*. 4th edition, 1967.

[78] G. E. Troxell, H. E. Davis, and J. W. Kelly. *Composition and Properties of Concrete*. 1968.

[79] K. Vance, G. Falzone, I. Pignatelli, M. Bauchy, M. Balonis, and G. Sant. Direct carbonation of ca(oh)2 using liquid and supercritical co2: Implications for carbon-neutral cementation. *Ind. Eng. Chem. Res.*, 54(36):8908–8918, 2015.

[80] K. Vance, G. Falzone, I. Pignatelli, M. Bauchy, M. Balonis, and G. Sant. Direct carbonation of ca(oh)2 using liquid and supercritical co2: Implications for carbon-neutral cementation. *Industrial Engineering Chemistry Research*, 54:8908–8918, 2015.

[81] A. Vellido, J. Martín-Guerrero, and P. Lisboa. Making machine learning models interpretable. In *ESANN*, pages 163–172. Citeseer, 2012.

[82] Y. Wang, N. Wagner, and J. Rondinelli. Symbolic regression in materials science. *MRS Communications*, 9:793–805, 2019.

[83] S. Wild, B. B. Sabir, and J. M. Khatib. Factors influencing strength development of concrete containing silica fume. *Cem. Concr. Res.*, 25(7):1567–1580, 1995.

[84] S. Wild, B. B. Sabir, and J. M. Khatib. Factors influencing strength development of concrete containing silica fume. *Cement and Concrete Research*, 25:1567–1580, 1995.

[85] I.-C. Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cem. Concr. Res.*, 28(12):1797–1808, 1998.

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

[87] B. A. Young, A. Hall, L. Pilon, P. Gupta, and G. Sant. Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: New insights from statistical analysis and machine learning methods. *Cem. Concr. Res.*, 115:379–388, 2019.

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

[89] M. Zain, H. Mahmud, A. Ilham, and M. Faizal. Prediction of splitting tensile strength of high-performance concrete. *Cement and Concrete Research*, 32:1251–1258, 2002.

[90] M. F. M. Zain and S. M. Abd. Multiple regression model for compressive strength prediction of high performance concrete. *J. Appl. Sci.*, 9(1):155–160, 2009.

[91] D. Zhenchao. Discussion on problem of standard deviation of concrete strength. *ACI Materials Journal*, 117, 2020.

[92] A. Öztaş, M. Pala, E. Özbay, E. Kanca, N. Çag˘lar, and M. Bhatti. Predicting the compressive strength and slump of high strength concrete using neural network. *Construction and Building Materials*, 20:769–775, 2006.