MULTI-OBJECTIVE DENSITY DIAGRAMS DEVELOPED WITH MACHINE LEARNING MODELS TO OPTIMIZE SUSTAINABILITY AND COST-EFFICIENCY OF UHPC MIX DESIGN

A Dissertation

by

CESARIO SARMENTO GONCALVES MARTINS E TAVARES

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Zachary Grasley
Committee Members,	Dan Zollinger
	Robert Lytton
	Thomas Lacy
Head of Department,	Zachary Grasley

May 2022

Major Subject: Civil Engineering

ABSTRACT

The emergence of ultra-high-performance concrete (UHPC) as an attractive solution for precast and prestressed applications has coincided with global efforts towards sustainable construction. The increasing need for tools capable of intuitively demonstrating the effect of concrete mixture composition on mechanical performance, cost and eco-efficiency concurrently has motivated this work in an effort to promote design of more sustainable solutions to help meet environmental goals. Such tools are needed to effectively evaluate the environmental impact of UHPC given the outstanding mechanical properties of the material coupled with high volumetric embodied CO_2. Meanwhile, artificial intelligence (AI) techniques have emerged as a great opportunity for game-changing tools capable of effectively modeling the synergistic relationships between mix proportions and material performance. This work couples machine learning models with orthogonal arrays to generate machine-learning-based tools to evaluate the tradeoffs between emissions, cost and mechanical performance concurrently. Random forest and k-nearest neighbors' models are ensembled to predict the compressive strength of UHPC mixtures and generate Performance Density Diagrams (PDDs). These predicted strengths are then coupled with volumetric environmental factors and unit costs to generate eco- and cost-efficiency density diagrams. The makeup of these tools facilitates the evaluation of rather complicated trends associated with mix proportions and multi-objective outcomes, allowing AI-based tools to be of easy use by industry personnel on a daily basis, while serving as decision-making aids during mix design stages and provide proof of mixture optimization that could be introduced in Environmental Product Declarations. The PDD developed herein enabled the design of a mix with compressive strength of 155 MPa, while keeping the aggregate-to-cementitious ratio above unit. Other mixtures were developed from these models and compared to several different concretes from the literature. Results show that high paste content, high strength (and ultra-high strength) concrete technologies are not necessarily detrimental to cost or eco efficiencies. For the different indices evaluated, optimum solutions were mostly obtained with these types of concrete, which means that industry trends toward requiring minimization of embodied CO_2 in concrete on a per volume basis are misguided and do not minimize the embodied CO_2 in concrete structures.

Dedicated to my parents, Angelica and Agnelo

ACKNOWLEDGEMENTS

There are several individuals that I would like to acknowledge for their role and contributions throughout my graduate studies and up to the writing of this dissertation.

First, I would like to thank my advisor and mentor, Dr. Zachary Grasley. For believing in my potential and capabilities, encouraging me to explore scientific fields that I never thought I was capable of navigating in. For challenging me academically, pushing me to sharpen my thinking and extend my skillset, ultimately impacting my academic and professional growth. For contributing with your invaluable expertise in formulating the methodology implemented in this work. For the constant support and advice related to my future career. Thank you!

I would like to thank the committee members in Dr. Dan Zollinger, Dr. Thomas Lacy and Dr. Robert Lytton for your insightful comments and advice on my preliminary work. Your feedback greatly contributed in shaping this research, making it a much better work.

I would like to acknowledge my former research group colleagues, whose advice, discussions and guidance greatly contributed to my academic growth and success: Dr. Josh Hogancamp, Dr. Aishwarya Baranikumar, Dr. Christa Torrence, Dr. Xijun Shi and Dr. Sajib Saha. I also want to thank all the graduate and undergraduate students who performed research with me throughout these years. In particular, I want to thank Juan Ayala and Xincheng Wang, whose interest, effort and dedication were of vital importance in generating the data and methodology implemented in this work.

I also would like to thank my teaching mentors, Dr. Bjorn Birgisson, Dr. Dallas Little and Dr. Yong-Rak Kim. Working and learning from you during my Teaching Assistant' experience was an honor and definitely shaped my future career goals. If today I want to be a professor, you had a lot to do with it.

Finally, I want to thank my family for all the love, support and dedication towards my personal and professional growth. To my parents, Angelica and Agnelo Tavares, for turning my dream into yours and dedicating your lives to your children's success, and my sister, Liliana Tavares, for being there for me since day one.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Zachary Grasley [advisor], Professor Dan Zollinger and Professor Robert Lytton of the Zachry Department of Civil and Environmental Engineering [Home Department] and Professor Thomas Lacy of the J. Mike Walker '66 Department of Mechanical Engineering [Outside Department]. All work conducted for the dissertation was completed by the student.

Funding Sources

Graduate study was partially supported by a graduate research assistantship under the National Science Foundation (NSF) Federal Award ID No. 1562123, as well as a graduate teaching assistantship from Texas A&M University and a graduate teaching fellowship from Texas A&M University.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOLS AND ABREVIATIONS

Symbol	*Definition*
# obs	number of observations
AI	artificial intelligence
ANN	artificial neural network
A_s	cross-sectional area of steel
Avg	average
b	width
CEDD	cost-efficiency density diagram
CO_2	carbon dioxide
CO_2-eq	co_2 equivalent
CPM	compressible packing models
DOE	design of experiment
DSP	densified with small particles concretes
EEDD	eco-efficiency density diagram
EPD	environmental product declaration
f_c	compressive strength
f_r	modulus of rupture
FRC	fiber reinforced concrete
GHG	greenhouse gas
GWP	global warming potential
h	depth
HPC	high performance concrete
HRWR	high-range water reducer
HSC	high strength concrete
i_c	volumetric environmental impact of concrete
IPCC	intergovernmental panel on climate change

ISO	international organization for standardization
i_s	volumetric environmental impact of steel
ITZ	interfacial transition zone
kNN	k-nearest neighbors
l	length
M_c	moment resisted by the concrete
MDF	macro-defect-free concrete
$M_{\max}$	maximum moment in the outer fiber
$MM_{emissions}$	multi-member eco-efficiency index
MM_{cost}	multi-member cost-efficiency index
MO_{column}	multi-objective comparison index for short columns
$MO_{cracking}$	multi-objective comparison index for initial cracking due to bending
MODD	multi-objective density diagram
N	number of observations
NSC	normal strength concrete
PDD	performance density diagram
RF	random forest
RMSE	root mean squared error
S/N	signal-to-noise ratio
SCC	self-consolidating concrete
SCM	supplemental cementitious material
Sd	standard deviation
SMD	statistical mixture design
sp	specimen
tp	tuning parameter
u_c	volumetric unit cost
UHPC	ultra-high performance concrete
w	distributed load

WBCSD	world business council for sustainable development
y_i	predicted outcome
y_i	observed outcome
$\tilde{I}$	total environmental impact of a member
$\chi_{cracking}$	eco-efficiency index associated with initial cracking due to bending
χ_{column}	eco-efficiency index associated with axial loading of short columns
$\overline{\chi}_{cracking}$	$\chi_{cracking}$ normalized by $\chi_{cracking,\max}$
$\overline{\chi}_{column}$	χ_{column} normalized by $\chi_{column,\max}$
ρ_{column}	cost-efficiency index associated with axial loading of short columns
$\rho_{cracking}$	cost-efficiency index associated with initial cracking due to bending
$\overline{\rho}_{column}$	ρ_{column} normalized by $\rho_{column,\max}$
$\overline{\rho}_{cracking}$	$\rho_{cracking}$ normalized by $\rho_{cracking,\max}$
α	weighting coefficient
β	weighting coefficient equivalent to (1-α)
λ_{fr}	coefficient associated with the modulus of rupture

CHAPTER I.

INTRODUCTION

1.1. Problem Statement

In the last three decades, cement production has grown faster than any other construction material, leading concrete to become the second highest consumed material in the world after water (Andrew 2019; Monteiro, Miller, and Horvath 2017). Carbon dioxide (CO2) emissions associated with Portland cement manufacturing is a major sustainability issue facing the concrete industry, accounting for approximately 8-9% of the global total anthropogenic greenhouse gas (GHG) emissions (Brinkman and Miller 2021). In the same timeframe, advanced concrete materials such as ultra-high performance concrete (UHPC) have emerged as an attractive option for precast and prestressed applications due to its outstanding mechanical properties and superior durability. The advanced properties of UHPC stimulate the development of innovative superstructural elements by promoting design with more efficient shapes and cross-sectional dimensions. This can be achieved due to the high ultimate compressive strength of UHPC materials, which provides new opportunities for bridges by allowing increased prestressing levels that enable the design of longer spans and thinner depth elements (ACI-Committee-239 2018). Yet, most UHPC compositions (ACI-Committee-239 2018; Ibrahim et al. 2017; Alsalman, Dang, and Micah Hale 2017; JJ Park 2008; Talebinejad 2004) involve high dosages of cement and increased costs per unit volume. This has contributed to an overall skepticism within the construction industry, often mis-associating the application of this material with high GHG emissions and raising questions regarding the sustainability of this material. Similarly, the higher

costs per unit volume of UHPC compared to traditional concrete is still a source of hesitancy between designers and project owners, inhibiting them from fully embracing this technology, despite structural elements being geometrically more efficient when cast with UHPC, requiring less volume of material. Thus, it is imperative to promote smart design strategies and proper tools to effectively compare performance, durability, cost and sustainability of UHPC against other traditional concrete technologies.

Several studies have been done throughout the years to evaluate the global warming potential (GWP) associated with the production of raw constituents of concrete (Celik et al. 2015; Purnell 2012; Liu et al. 2012; Flower and Sanjayan 2007). Methods for reducing CO_2 emissions have been gradually implemented (Miller, Horvath, and Monteiro 2016). While most methods did not take into account the influence of material properties and environmental impacts concurrently, comparison indices have been proposed by Miller et al. (Miller et al. 2016), for unreinforced concrete, and later by Kourehpaz and Miller (Kourehpaz and Miller 2019) for reinforced concrete members, allowing environmental impacts and mechanical properties to be evaluated concurrently. While these new metrics represents an important step towards a better assessment of sustainability in construction, new tools for improved visualization and evaluation of the contribution of each mixture constituent towards GWP are still a major need in the concrete industry.

Considering that quality and proportioning of UHPC ingredients is one of the main factors dictating mechanical performance, cost and sustainability, it is vital to promote smart mixture design optimization strategies to make an efficient use of this material. The main challenge related to mixture optimization is that it not only involves satisfying multi-objective performance

levels (materials properties, carbon footprint and cost), but it also entails a synergistic relationship between the various constituents and the various desired outcomes, making linear regression modeling ineffective. Recently, artificial intelligence (AI) branches such as machine learning and deep learning techniques have gained momentum in optimization and prediction studies (Ghafari et al. 2012; Solhmirzaei et al. 2020; Ren et al. 2021). Yet, machine learning models are often viewed as "black-boxes" that receive inputs and returns predictions, with different models characterized by unique parameters, which typically are not as interpretable as, for instance, linear regression equations. Furthermore, the efficiency of AI models depend on the size, distribution, and quality of the data, with most models requiring very large sets of experimental data to attain acceptable prediction performance. This has led researchers to collect data from various sources in the literature to develop their models (Abuodeh, Abdalla, and Hawileh 2020; Solhmirzaei et al. 2020; Sadrossadat et al. 2021). Considering that cost, availability and intrinsic properties of raw materials can vary tremendously with source and region, most of the available models are limited to the specific pre-existing conditions in the datasets. That is, the accuracy of these models is greatly influenced by whether the input variables are within the domain of the applicable boundary values used to train these models, making these models impractical to explore new materials in a constantly growing field such as the UHPC industry. Considering the amount of time, effort and resources required to produce and test laboratory specimens of UHPC following current standardized methods, tied with the costly investment on specialty equipment for end grinding, it is imperative to develop strategies and protocols that facilitate experimental data collection for advanced modeling, allowing

designers and producers to efficiently generate enough data capable of characterizing material behavior within the experimental domain of interest.

Implementing strategic experimental frameworks for efficient data collection tied with rapid testing protocols and advances in AI techniques to address the needs aforementioned, represents an incredible opportunity for innovative design protocols and tools that facilitate the evaluation of the tradeoffs between cost, sustainability and mechanical performance for a range of mix proportions, while also offering enough flexibility for exploring new constituents and accounting for differences in source, availability and cost of material per region. Although sharing data within the research community has a great value to advance collective knowledge, it is important to promote a mix design process that does not rely exclusively on models that are applicable to a limited set of pre-conditions. Furthermore, facilitating the visualization of the internal predictive structure of machine learning models can help address their "black-box" nature and facilitate the communication between different parts involved in projects (owner, regulating entities and designers). Addressing these needs can contribute to a more efficient and fair comparison between different concrete solutions, lifting the existing mis-conceptional barriers regarding UHPC and promote its use where this material is clearly a better option compared to conventional concrete.

1.2. Project Objectives and Primary Contributions

This study presents an innovative approach to design and optimize concrete materials with strengths up to UHPC levels, using a framework that allows one to efficiently design UHPC mixtures considering multi-objective performance levels. In particular, this work focuses on the

development of machine-learning-based tools to maximize compressive strength, cost and eco-efficiency of UHPC mixtures, through cement replacement with waste by-product powders and aggregates.

A strategic framework of experimental data collection is coupled with machine learning models to generate performance density diagrams (PDDs). These diagrams intend to provide a flexible tool to, intuitively, evaluate and communicate the influence of mixture proportioning on mechanical and durability properties directly affecting cost and environmental impact.

Orthogonal arrays used in the Taguchi method are applied in this study to establish the experimental data collection framework, allowing reduced experimental runs to capture relevant datapoints capable of describing the entire experimental domain. A model that combines multiple machine learning techniques to predict and optimize the compressive strength of UHPC mixtures will be developed. The experimental data are modeled with two machine learning algorithms, k-nearest neighbors (kNN) and random forest. The two models are then combined into a single ensemble model for improved prediction of compressive strengths, benefiting from the best features from each model. The generated matrix of predictors and outcomes is used to develop the aforementioned PDDs, intended to not only serve as a decision-making aid tool during mix design stages, but also to help overcome the lack of visual interpretability of machine learning algorithms. PDDs allow performance, durability and environmental impact of different mixture compositions to be evaluated simultaneously, while facilitating proper considerations involving material availability and trade-offs between cost-performance.

Additional models will be created to evaluate cost and environmental impact of different mixture compositions. First, a global warming potential (GWP) parameter, representing the volumetric environmental impact, is combined with compressive strengths to attain eco-efficiency indices to allow accurate assessments of the total environmental impact associated with a produced concrete element. Similarly, volumetric costs associated with each mixture composition and the corresponding unit costs from each ingredient is coupled with the predicted compressive strengths from the developed machine learning models to calculate cost-efficiency indices. The calculated cost and eco-efficiency indices are introduced into the generated matrix of predictors, aforementioned, to generate diagrams resembling PDDs. Finally, these indices are combined into multi-objective functions to allow an overall evaluation of UHPC mixtures that accounts for mechanical performance, cost and environmental impacts concurrently.

To showcase the value and flexibility of these tools, UHPC mixtures with particular characteristics will be identified and tested in laboratory for validation. For this study, a new protocol was developed to characterize the compressive strength of very-high strength binders. The results obtained with this protocol will be compared to results obtained following protocols defined in current standards ASTM C39 and ASTM C1856 to validate the method. The existence of a single end-specimen condition in the current standard limits research engagement from universities that do not possess a fixed-end grinder equipment, which is currently costly (approx. $30,000) when compared to other methods that exist for testing conventional concrete. The testing protocols developed in this study will facilitate the assessment of compressive strengths of UHPC, promoting research development in the field of UHPC materials. Fig. 1 illustrates the flowchart describing the approach and workflow followed in this study.

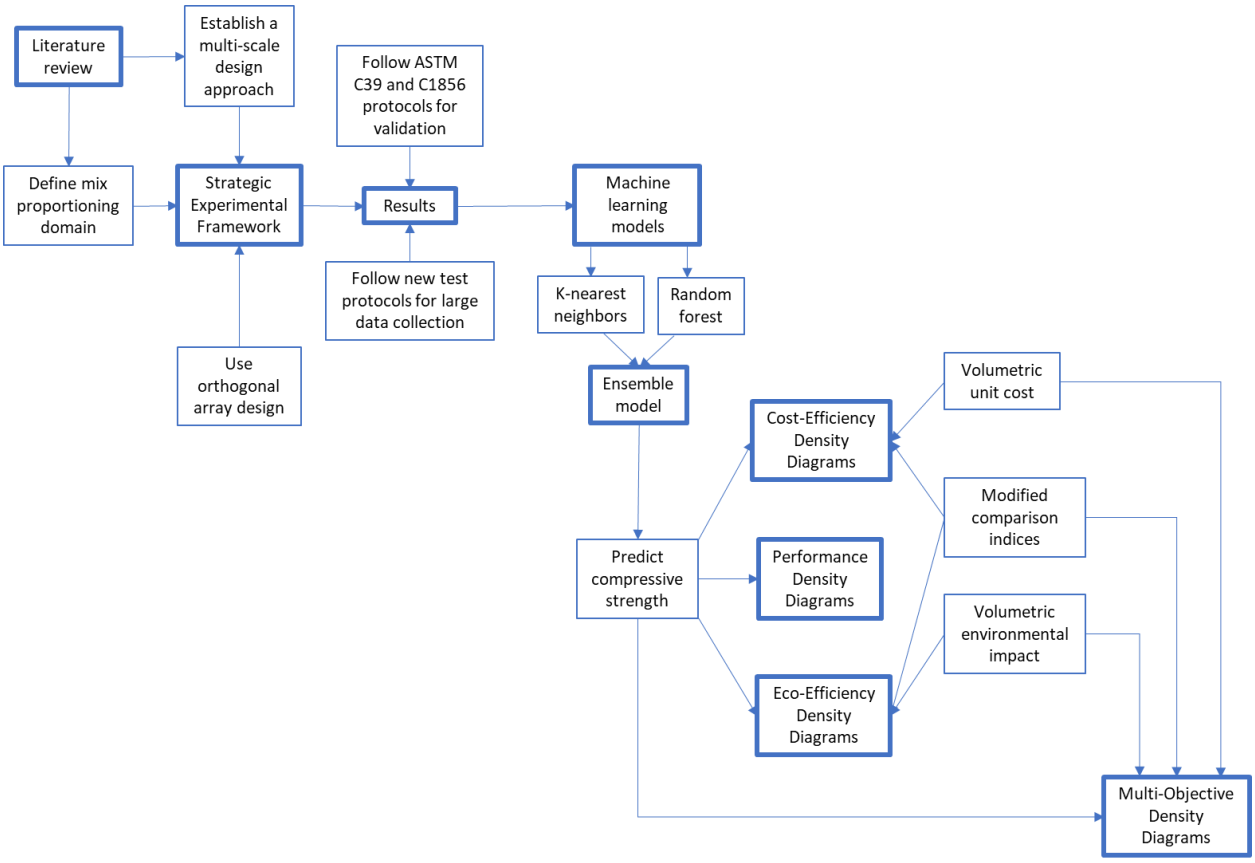


Fig. 1. Flowchart describing the approach and workflow followed in this study

Contributions will be made to the field on the overall understanding regarding the influence of cement replacement with supplemental cementitious materials (SCMs), natural and manufactured sands, and ground quartz on compressive strength of UHPCs cured without special thermal and/or moisture treatments.

As a contribution to practicing engineers, designers and regulating entities, the approach used in this study will enable performance, durability, cost and environmental impacts to be evaluated concurrently without adding exhausting experimental campaigns or limiting models to a set of pre-existing conditions from the literature data. This method will facilitate the decision making

process surrounding material availability, cost-performance trade-off and sustainability requirements.

The primary objectives of the proposed research are summarized below:

- Provide new test protocols that promote expeditious and efficient assessment of compressive strengths of UHPC materials to enhance and facilitate advanced, data-science based modeling of new formulations;
- Enable an innovative mix design methodology supported by performance, cost and eco-efficiency density diagrams as ideal tools for product characterization, offering a game-changing approach to the design and decision making process with respect to assessing performance, cost, availability and environmental impact of concrete materials;
- For global optimization of mixture designs, provide guidance on developing multi-objective and multi-member comparison indices and density diagrams, capable of demonstrating the influence of performance, cost and eco-efficiencies concurrently, while filtering out mixtures that fail to meet performance requirements imposed by design.

1.3. Thesis Outline

This dissertation is organized in the following manner: Chapter II provides a review on UHPC and mix proportioning implications on environment impact, along with mix design optimization, orthogonal arrays for reduced experimental runs and concrete performance predictions using machine learning models. Meanwhile, Chapter III describes the strategic experimental framework established with orthogonal arrays and new test protocols for efficient data

collection, along with materials used and specimen preparation protocols followed. The experimental results obtained and presented in Chapter III are used as inputs for the models developed in the subsequent chapters. In Chapter IV, machine learning models are developed using data collected in Chapter III to predict the compressive strength of mix compositions within the defined experimental domain. Prediction performance of various models are evaluated through common metrics and complemented with three-dimensional performance density plots to evaluate model behavior. The compressive strengths predicted by the optimum models in Chapter IV are used in Chapters V, VI and VII to develop new machine-learning-based graphical tools created to intuitively demonstrate the effect of mix proportioning on mechanical performance, cost and environmental impact. Finally, Chapter VIII summarizes the primary conclusions of this dissertation and suggests future work to possibly extend the methodology performed in this study into practical applications.

1.4. References

Abuodeh, Omar R., Jamal A. Abdalla, and Rami A. Hawileh. 2020. 'Assessment of compressive strength of Ultra-high Performance Concrete using deep machine learning techniques', *Applied Soft Computing*, 95: 106552.

ACI-Committee-239. 2018. "Ultra-High Performance Concrete: An Emerging Technology Report." In, 21.

Alsalman, Ali, Canh N. Dang, and W. Micah Hale. 2017. 'Development of ultra-high performance concrete with locally available materials', *Construction and Building Materials*, 133: 135-45.

Andrew, R. M. 2019. 'Global CO2 emissions from cement production, 1928–2018', *Earth Syst. Sci. Data*, 11: 1675-710.

Brinkman, Leah, and Sabbie A Miller. 2021. 'Environmental impacts and environmental justice implications of supplementary cementitious materials for use in concrete', *Environmental Research: Infrastructure and Sustainability*, 1: 025003.

Celik, Kemal, Cagla Meral, A. Petek Gursel, P. Kumar Mehta, Arpad Horvath, and Paulo J. M. Monteiro. 2015. 'Mechanical properties, durability, and life-cycle assessment of self-consolidating concrete mixtures made with blended portland cements containing fly ash and limestone powder', *Cement and Concrete Composites*, 56: 59-72.

Flower, David J. M., and Jay G. Sanjayan. 2007. 'Green house gas emissions due to concrete manufacture', *The International Journal of Life Cycle Assessment*, 12: 282.

Ghafari, E., M. Bandarabadi, H. Costa, and E. Júlio. 2012. 'Design of UHPC using artificial neural networks.' in A. M. Brandt, J. Olek, M. A. Glinicki and C. K. Y. Leung (eds.), *Brittle Matrix Composites 10* (Woodhead Publishing).

Ibrahim, Mustapha A., Maen Farhat, Mohsen A. Issa, and Jessica Amanda Hasse. 2017. 'Effect of Material Constituents on Mechanical and Fracture Mechanics Properties of Ultra-High-Performance Concrete', *ACI Materials Journal*, 114: 453-65.

JJ Park, ST Kang, KT Koh, SW Kim. 2008. "Influence of the ingredients on the compressive strength of UHPC as a fundamental study to optimize the mixing proportion." In *International Symposium Ultra High Performance Concrete (UHPC)*. Kassel, Germany.

Kourehpaz, Pouria, and Sabbie A. Miller. 2019. 'Eco-efficient design indices for reinforced concrete members', *Materials and Structures*, 52: 96.

Liu, Rui, A. Durham Stephan, L. Rens Kevin, and Anu Ramaswami. 2012. 'Optimization of Cementitious Material Content for Sustainable Concrete Mixtures', *Journal of Materials in Civil Engineering*, 24: 745-53.

Miller, Sabbie A, Arpad Horvath, and Paulo JM Monteiro. 2016. 'Readily implementable techniques can cut annual CO2 emissions from the production of concrete by over 20%', *Environmental Research Letters*, 11: 074029.

Miller, Sabbie A., Paulo J. M. Monteiro, Claudia P. Ostertag, and Arpad Horvath. 2016. 'Comparison indices for design and proportioning of concrete mixtures taking environmental impacts into account', *Cement and Concrete Composites*, 68: 131-43.

Monteiro, Paulo J. M., Sabbie A. Miller, and Arpad Horvath. 2017. 'Towards sustainable concrete', *Nature Materials*, 16: 698-99.

Purnell, P. 2012. 'Material Nature versus Structural Nurture: The Embodied Carbon of Fundamental Structural Elements', *Environmental Science & Technology*, 46: 454-61.

Ren, Qiang, Luchuan Ding, Xiaodi Dai, Zhengwu Jiang, and Geert De Schutter. 2021. 'Prediction of Compressive Strength of Concrete with Manufactured Sand by Ensemble Classification and Regression Tree Method', *Journal of Materials in Civil Engineering*, 33: 04021135.

Sadrossadat, Ehsan, Hakan Basarir, Ali Karrech, and Mohamed Elchalakani. 2021. 'Multi-objective mixture design and optimisation of steel fiber reinforced UHPC using machine learning algorithms and metaheuristics', *Engineering with Computers*.

Solhmirzaei, Roya, Hadi Salehi, Venkatesh Kodur, and M. Z. Naser. 2020. 'Machine learning framework for predicting failure mode and shear capacity of ultra high performance concrete beams', *Engineering Structures*, 224: 111221.

Talebinejad, I. ; Bassam, S.; Iranmanesh, A.; Shekarchizadeh, M. 2004. 'Optimizing Mix Proportions of Normal Weight Reactive Powder Concrete with Strengths of 200-350 MPa', *Proceedings of the International Symposium on UHPC, Kassel, Germany*: 133-41.

CHAPTER II.

LITERATURE REVIEW

2.1. Ultra-High Performance Concrete (UHPC)

The UHPC definition is currently limited to concretes with compressive strengths over 150 MPa (22,000 psi), with specified durability, tensile ductility and toughness requirements (ACI-Committee-212). UHPC typically includes fibers and displays strain-hardening characteristics under uniaxial tension.

The development of this class of materials started in the 1970s with investigations on high strength cement pastes with low water-to-cementitious materials ratio (w/cm), achieving compressive strengths up to 29 ksi (200MPa) (Yudenfreund, Odler, and Brunauer 1972; Yudenfreund, Skalny, et al. 1972; Yudenfreund, Hanna, et al. 1972; Odler, Yudenfreund, et al. 1972; Odler, Hagymassy, et al. 1972; Brunauer, Yudenfreund, et al. 1973; Brunauer, Skalny, et al. 1973), followed by investigations of hot pressing techniques that resulted in pastes with strengths up to 98ksi (680MPa) (Roy, Gouda, and Bobrowsky 1972; Roy and Gouda 1973). The emergence of high range water reducers (HRWR) in the 1980s, along with pozzolanic admixtures such as silica fume, led to the development of Macro-Defect-Free concretes (MDF) (Kendall et al. 1983; Alford and Birchall 1984), followed by Densified with Small Particles concretes (DSP) which ultimately became the basis for the development of modern UHPC (Bache 1987). Since then, the density of UHPC matrixes has been theoretically investigated and optimized (de Larrard and Sedran 1994), while the brittleness of this material has been identified and addressed with the use of several combinations of fibers (Bache 1987; Richard and Cheyrezy 1995).

Commercial application of UHPC started in Europe in the 1990s and it became available in North America for the first time around 2000. UHPC has been used in many applications such as bridges and infrastructure, security and blast resistance, buildings, facades and elements exposed to aggressive environments. These applications involve new construction and rehabilitation, using both precast components and cast-in-place processes.

2.1.1. Mixture Design Principles, Materials and Common Mix Proportioning

The following mixture design principles are typically considered to attain the outstanding properties of UHPC (Richard and Cheyrezy 1995; ACI-Committee-239 2018):

- Increase in homogeneity obtained by eliminating coarse fraction of aggregates from the matrix
- Improved microstructure obtained by optimizing the packing density of the matrix through dense gradation of solid particles and maximization of pozzolanic reactions triggered by the presence of reactive silica existing in mineral admixtures such as silica fume, slag and fly ash.
- Enhanced ductility obtained from using discontinuous fibers to promote strain-hardening behavior in tension.

Most of the materials used to produce UHPC are identical as for traditional concrete, although the water-to-binder ratio usually ranges between 0.15 and 0.25 (ACI-Committee-239 2018). UHPC typically consists of cement, silica fume, fine quartz sand, HRWR, and fibers (ACI-Committee-239 2018; Ibrahim et al. 2017; Alsalman, Dang, and Micah Hale 2017; JJ Park 2008; Talebinejad 2004). However, several mixture compositions have been developed containing

other SCMs such as slag and fly ash (Ibrahim et al. 2017; Alsalman, Dang, and Micah Hale 2017; Yazıcı 2007), as well as coarse fractions of aggregates (Ma et al. 2004; Li, Yu, and Brouwers 2018). High-strength steel fibers are often added to the matrix to improve the ductility and ability to eliminate some of the mild steel reinforcement often present in conventional reinforced concrete members (Graybeal 2006b). In UHPC compositions, the typical ranges of cement mass replacement with mineral admixtures are 10-25% of silica fume, 25-30% of ground quartz, 10-40% of fly ash and 55-59% of slag. The method of curing also impacts the resulting material properties of UHPC (Cheyrezy, Maret, and Frouin 1995; Graybeal 2006a; Graybeal and Stone 2012).

Certain commercially available UHPC mixtures are proprietary, in which their exact composition is not reported. Meanwhile, other mixture designs are readily published and available. Different formulations often make trade-offs to achieve improvement of one property that may adversely impact others.

2.1.2. Mechanical Properties, Durability and Applications

Just like with any class of concrete, the mechanical properties of UHPC vary with mixture formulation, curing conditions and testing age. The strength and elastic modulus measured in UHPC are significantly higher than conventional concrete. Table *1* (ACI-Committee-239 2018) summarizes typical ranges observed for selected mechanical properties of UHPC versus conventional concrete.

Table 1. Comparison of conventional concrete and UHPC (adapted from (ACI-Committee-239 2018))

Material characteristic	Conventional concrete	UHPC
Compressive strength	20 to 40 MPa (3000 to 6000 psi)	150 to 250 MPa (22,000 to 36,000 psi)
Direct tensile strength	1 to 3 MPa (150 to 440 psi)	6 to 12 MPa (900 to 1700 psi)
Elastic modulus (ASTM C469/C469M)	25 to 30 GPa (3,600,000 to 4,400,000 psi)	40 to 50 GPa (6,000,000 to 7,200,000 psi)

UHPC typically has a considerably higher tensile strength and sustained tensile capacity compared to fiber reinforced concrete (FRC) and conventional concrete. In UHPC, the tensile strength is often included in structural design calculations as opposed to conventional concrete, where this property is usually neglected. Additionally, the more important feature of the tensile behavior of UHPC is its post-cracking ductility, obtained with a particular combination of fibers and matrix microstructure, typically characterized by a strain-hardening behavior (continuous increase in resistance after reaching the cracking stress) or strain-softening behavior (gradual and steady decrease of resistance after reaching the cracking stress). The method of curing also impacts the resulting material properties of UHPC (Cheyrezy, Maret, and Frouin 1995; Graybeal 2006a; Graybeal and Stone 2012).

The durability performance of UHPC is far superior of that from conventional concrete due to the discontinuous pore structure, dense matrix and multi-cracking behavior at the microscale. The greater density of the interstitial transition zone between aggregates and matrix and higher density of hydration products results in improved resistance to several harmful gases and liquids, freeze-thaw cycles and chloride attack (Schmidt and Fehling). The porosity of UHPC is typically around 9 percent as opposed to the 15 percent commonly measured in conventional concrete

(Roux, Andrade, and Sanjuan 1996), with little to no capillary pores (Cheyrezy, Maret, and Frouin 1995). Durability properties are typically measured by permeability, freezing-and-thawing, scaling, abrasion, resistance to chloride ingress, alkali-silica reaction, and carbonation tests (Russell and Graybeal 2013).

The outstanding properties of UHPC makes this material suitable for a variety of applications such as highway bridge girders, pedestrian bridges, seismic retrofit of columns, rehabilitation of structures, bridge piles/foundations, field-cast connections, safety and security infrastructure, spent nuclear fuel storage, facades, canopies and shells, impact resistant infrastructure and elements subjected to aggressive environments (ACI-Committee-239 2018).

2.2. Mixture Proportioning Effect on Environmental Impact of UHPC

The last couple of decades have been characterized by a global awareness on rising GHG emissions, with the Intergovernmental Panel on Climate Change (IPCC) calling for 50-85% reductions in these emissions by 2050 to prevent threatening climate changes (Allwood et al. 2013). The building and construction industries have been among the leading consumers of material by mass for almost 100 years (Horvath 2004), with cement manufacturing representing approximately 8-9% of the global total anthropogenic GHG emissions (Brinkman and Miller 2021). With global population growth and the rapid development of third world countries, it is unlikely that advancements in the process efficiency of material manufacture will occur fast enough to meet emission reduction needs considering that demand for materials is expected to double by 2050 (Allwood et al. 2013).

New metrics that account for material performance and environmental impact concurrently during design and material selection stages is of great importance and represents a great opportunity for impactful methods to reduce emissions (W.B.C.S.D. 2009). Considering this challenge, numerous international entities such as the World Business Council for Sustainable Development (WBCSD) and the International Energy Agency have recommended further evaluation of material property changes associated with constituent alteration as a timely-convenient strategy to minimize GHG emissions (W.B.C.S.D. 2009). Mehta et al. (Mehta 2009) recommended three approaches to the concrete industry for achieving sustainability: 1) develop innovative architectural concepts and structural designs that minimizes concrete consumption in new construction and rehabilitation of existing United States structures; 2) specify 56-or-90-day compressive strengths when conceivable to reduce the amount of cementitious material required in a mixture and; 3) follow smart concrete mixture proportioning approaches that consider blended cements containing high proportions of pozzolans. Although the latest is a promising approach to significantly reduce environmental impact of concrete materials, it is still limited due to lack of understanding of the compositional complexity of SCMs.

Recently, emphasis has been placed on Environmental Product Declarations (EPDs) as a product "label" to promote transparency with pertains to the efforts made by concrete producers towards improving the sustainability of their products. The International Organization for Standardization (ISO) 14025 and European Standards (EN) 15804 define EPDs as independently verified documents that report environmental data of products based on life cycle assessment (LCA). Usually, EPDs contains information on a wide range of environmental impacts (e.g., GHG, ozone depletion, water impacts, habitat destruction, toxic substances, etc.) throughout the

product's life cycle (cradle to grave) to measure the sustainability of a product while enabling comparisons with other products fulfilling the same function (CARBONCURE 2021; rediscover_concrete 2021; The_Concrete_Center 2021). While EPDs are currently reported using environmental impacts in units of kg CO_2-eq. per cubic meter of concrete produced, this metric does not promote a consistent and accurate quantification of environmental impacts for the totality of a project. For instance, minimization of CO_2-eq. per volume of concrete might not lead to a minimization of CO_2-eq. produced in the building of a given structure due to the trade-off between mechanical properties and embodied emissions that might require larger structural elements when using a low CO_2 concrete.[1]There is still no consistent analysis or reporting method used at a global scale to quantify and compare emissions generated in construction based on constituent proportioning in mixtures and the resulting performance of the final product.

This omission is particularly important for advanced concrete materials such as UHPC, which typically involves high dosages of cement in its composition. High cement dosages and increased costs are the two main sources of skepticism surrounding the implementation of UHPC in the construction industry. In the last few years, researchers have developed sustainable and economical approaches to design UHPC by reducing the amount of cement and silica fume, while compensating these reductions with fly ash and sand (Soliman and Tagnit-Hamou 2017; Abbas, Nehdi, and Saleem 2016; Wille, Naaman, and Parra-Montesinos 2011). However, these studies did not focus on developing tools to quantify the actual environmental impact of the various

[1] An additional limitation of concrete material EPDs is the neglecting of emissions associated with transportation of the material to a specific jobsite, which can vary significantly from one producer. Neglecting this contribution penalizes those companies that move materials in a more efficient manner (e.g., water-based transport versus rail or road).

mixtures to justify the trade-off provided by UHPC in terms of performance-dimensionality of structural members. This aspect clearly influences the amount of GHG emissions for the totality of a project considering the dimensional reductions obtained with structural members made out of UHPC in comparison to conventional concrete. In several applications, using UHPC may well represent a better option from a performance, durability, cost and sustainability standpoint. However, tools to easily demonstrate this reality are still not available or well developed. Therefore, new metrics that account for material performance, cost and environmental impact concurrently, along with visualization tools that display the influence of mix proportioning in these outcomes are of vital importance to the UHPC industry.

While several researchers have conducted evaluations of both material properties and environmental impacts to compare different concrete mixtures (Celik et al. 2015; Purnell 2012; Liu et al. 2012; Flower and Sanjayan 2007), the results obtained in these studies lack a broader application with respect to elements or mixture compositions differing from the ones investigated. To address these limitations, new indices relating environmental impacts and mechanical properties have been recently developed (Damineli et al. 2010; Fantilli and Chiaia 2013; Gursel, Maryman, and Ostertag 2016; Miller et al. 2016b) to provide more comprehensive methods for comparison of mixture compositions with varying ingredients and processing methods. Although these studies denoted significant progress in eco-mechanical impact assessment, they did not consider the influence of mechanical properties on the volume of material needed for a given structural member, which is the determining factor on the environmental impact produced in the totality of a project. Considering that the volume of material needed for an application is a function of mechanical properties and loading conditions,

a ratio of volumetric environmental impact to material property is not a proper comparison measure for different mixture compositions.

Recent studies by Miller et al. (Miller et al. 2016a) and Kourehpaz and Miller (Kourehpaz and Miller 2019) provide novel approaches for environmental impact assessment, setting the path for the future of sustainable construction. New indices that consider interdependencies between mechanical properties and environmental impacts for different structural elements were developed. These indices were established from environmental impacts associated with cradle-to-gate production of one cubic meter of concrete, in addition to mixture parameters and material properties. Despite the significant progress that these studies represent to the field, the graphical representation of the data still needs significant improvement for a thorough characterization and proper assessment of the contribution from each mixture constituent on the environmental indices evaluated. A machine-learning-based tool is developed in this study to complement the contribution made with these comparison indices.

2.3. Mix Design Optimization and Performance Prediction using Machine Learning Models

Mixture proportioning optimization is driven by an ever-increasing need for decision-makers and designers to produce concrete mixtures capable of fulfilling multiple, and often times, competing performance requirements, including fresh and hardened material properties, durability, cost and environmental impact. Traditional methods for concrete mixture design are classified into two main approaches: prescriptive and performance-based. Prescriptive-based methods consist of frameworks that guide the designer in proportioning an acceptable mixture by following a

step-by-step process. The main limitation of this method resides in the fact that, while useful for general construction applications, it lacks flexibility, given that a designer cannot tailor and adjust individual mixture proportions. In contrast, performance-based approaches impose no strict guidelines on constituent proportions. That is, this approach provides substantial freedom to the designer in establishing unique compositions to meet design specifications. This process was mainly motivated by the emergence of several advanced concrete materials such as self-consolidating concrete (SCC), HPC, and more recently UHPC. The exceptional properties of UHPC are a function of the type, quality and proportions of its constituents. While these outstanding properties encourage broader applications for this material, the production cost remains considerably high due to large cement contents often used and also the elevated cost of other ingredients such as silica fume, fibers, ground quartz and high quality HRWR. Therefore, optimization of constituent dosages in UHPC compositions is crucial to not only ensure satisfactory performance levels, but also to increase the cost-effectiveness of this product to justify application.

Mixture proportioning optimization in UHPC materials started with Richard and Cheyrezy (Richard and Cheyrezy 1995) when compressible packing models (CPM) were used to optimize the granular mixture. Later on, Sonja et al. (Sonja AAM 2009) explained how interaction between very small particles caused by surface forces influences packing density, suggesting that CPM results can deviate considerably from experiments when very small particles are evaluated. According to Sonja et al., CPM could not accurately predict performance of mixtures with nano or micro sized particles. Since then, several authors have developed different strategies towards mixture optimization (Talebinejad 2004; Teichmann T 2004; JJ Park 2008; Kay Wille and Gustavo

2011). However, the design of experiment (DOE) strategies implemented in most of these studies does not allow the results to be generalized. Mathematical optimization and statistical mixture design (SMD) methods have been used for years, especially on multiple-objective problems. Comprehensive experimental optimization of concrete mixtures following standardized testing protocols can be costly, time-consuming and resource intensive. While SMD methods (Ghafari et al. 2015; Ghafari, Costa, and Júlio 2015) have progressed the way material properties are modeled while providing explicit equations relating decision variables to objectives, the performance of these equations are oftentimes insufficient to describe complex relationships. Moreover, these methods are not effective in modeling the nonlinear response in compressive strength with changes to the mixture proportions. To overcome this limitation, significant research has been done on non-linear regressions and computational methods that leverage the multitude of experimental data concerning advanced concrete materials, advanced mathematical techniques, and the power of high-performance computing (Deshpande, Londhe, and Kulkarni 2014; Mohammed et al. 2020; Saadat and Bayat 2019; Yagiz, Sezer, and Gokceoglu 2012).

Significant advances in AI has propelled the extensive use of machine learning models in numerous fields to estimate outcomes that closely resemble real world experiments. Machine learning models have consistently presented higher predictive performance compared to traditional mathematical and statistical methods (Sadrossadat et al. 2021; Solhmirzaei et al. 2020; DeRousseau et al. 2019; Ghafari et al. 2015). A variety of machine learning algorithms can be used to accomplish the same goal: detect patterns in datasets and improve predictive performance. However, the structure and functionality of each type of algorithm can differ

significantly. When the objective function of a model consists of continuous outcomes, the predictive performance is typically judged based on a loss function such as the root mean squared error (RMSE). In practice, the RMSE represents the standard deviation of the residuals (prediction errors), and is defined by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{2.1}$$

where $\hat{y}_i$ represents the predictions, y_i represents the actual outcomes and N represents the number of observations (Irizarry 2019). In machine learning, the RMSE is evaluated across different sets: training, validation and testing sets. The reasoning behind the data split across these sets is later explained in Chapter IV.

Machine learning algorithms such as k-Nearest Neighbor (kNN), random forest and artificial neural network (ANN) have gained popularity over time, as researchers have been exploring these techniques to solve a variety of problems in civil engineering (J.A. Abdalla 2012; Das 2013; Abdalla, Attom, and Hawileh 2015; Deng et al. 2018; Naderpour, Rafiean, and Fakharian 2018; DeRousseau et al. 2019). The emergence of UHPC compelled further development of machine learning modeling towards behavioral predictions (Ghafari et al. 2012; Solhmirzaei et al. 2020; Ren et al. 2021). Despite the growing popularity of machine learning in civil engineering, the black-box nature of these algorithms has inhibited researchers from describing the content of the models mathematically. These often contain complex internal structures, described by parameters that offer little insight on what takes place inside the model during computations.

Recognizing this limitation, various authors have employed several strategies intending to address it. Abuodeh et al. (Abuodeh, Abdalla, and Hawileh 2020) used sequential feature

selection and a neural interpretation diagram to identify the critical features affecting material response prediction when applying an ANN model. This approach allowed reducing the dimensionality of the database from eight to four features, simplifying the model while simultaneously increasing the accuracy. However, the black-box nature of these models was not resolved. In addition, the plots produced to interpret the data allow only two features and one response to be evaluated concurrently while fixing the other variables, which inhibits an overall look at the data and the ability to accurately assess the trends. Meanwhile, Kim et al. (Kim et al. 2020) characterized fly ash on a particle-by-particle basis using automated scanning electron microscopy (ASEM) and analyzed 20 different fly ashes following a principal component analysis (PCA) to assess interrelationships among particle chemical composition. This study represents an important step on mixture design optimization strategies, in which a more general classification of mixture constituents based on the individual particle make-up is encouraged to address the inconsistency of bulk properties in constituents such as fly ash. Furthermore, the graphical tools used to describe the results greatly facilitate the interpretation of material response, particularly with the use of ternary density plots. Yet, these plots are limited to applications where responses are a function of a maximum of three variables. This inhibits a full performance characterization in materials such as UHPC, which has a high number of variables affecting material response.

A key aspect to consider when defining suitability of machine learning models is robustness. This is especially important for highly complex algorithms with several parameters. The most challenging tasks in modeling large datasets with intelligent algorithms consist of preventing overfitting and data leakage. Overfitting occurs when the model performs significantly well on the training dataset but poorly on the testing dataset, implying poor generalized performance on

unseen data. While researchers are usually well aware of basic overfitting, data leakage is regarded as one of the top ten mistakes practiced in machine learning (Nisbet 2009). Data leakage occurs when information available in test sets is leaked into train sets. While it may lead to overfitting (overly optimistic accuracy on training set), data leakage is mostly characterized by causing overly optimistic accuracy on test sets. Thus, data leakage often results in unrealistically high accuracies on both training and testing sets, but very poor predictive performance on new/blind data. Overfitting and data leakage verifications are crucial in ensuring that models do not memorize how to interpret a specific range of inputs such that this range can be later implemented to predict results in a newly collected database.

Ghafari et al. (Ghafari et al. 2015) built two analytical models using ANN and SMD to predict fresh and hardened material properties. A total set of 53 different mixtures was used to build the models, in which 80% was used for training, while 20% was used for validation and testing. Considering that, neither a feature selection process nor a variable importance assessment was performed to evaluate the influence of each feature on the material response, the methodology used in this study could have induced data leakage. Several mixtures with identical contents on the main features but differing quantities on features with little influence on the model response (i.e., no feature was filtered) could have fallen into different modeling sets (training and testing data). This allows the model to see the same data in the training and testing sets, leading to over optimistic estimations ($R^2 > 95\%$). Further evidence of data leakage in this study comes from the fact that the optimized mixtures returned by the model were experimentally tested and results indicate that these were outperformed by several mixtures from the initial experiments.

Meanwhile, Abuodeh et al. (Abuodeh, Abdalla, and Hawileh 2020) collected 110 experimental results from the literature and performed a 70%-15%-15% split for training, validation and testing sets, respectively. In this study, numerical solver verification and overfitting verification processes were combined to resemble a k-fold cross validation approach in attempting to prevent overfitting. However, similarly to the previous study, a basic random splitting process does not ensure that mixture designs with identical input ranges (at least four pairs were identified in this case) are kept within the same splitting group prior to fitting the model to the data. As mentioned earlier in this section, a feature selection process was implemented to reduce the number of features in the model. Considering that only cement, fly ash, silica fume and water were selected as inputs for the final model, mixtures differing in other features (fiber content, sand, steel fiber, quartz powder and admixture) could have fallen into different sets of data as well. These two cases suggest that, when modeling large sets of data collected from the literature, a detailed data mining process is required to avoid overfitting and data leakage.

2.4. Orthogonal Array Design for Reduced Experimental Runs

Product manufacturing is a field that generally involves a multi-objective, multi-factor design approach to optimize the final properties of a given material in order to achieve desired performance levels. This challenging process involves defining an optimal setting of control factors and tailoring their values, where often the improvement of one property/response may result in performance loss in another property/response below acceptable levels. The Taguchi method is one of the most popular optimization methods within the field of production engineering (Rowlands, Antony, and Knowles 2000; Ranjit 2010). This method consists of performing a limited number of experimental runs based on orthogonal array design, providing

a well-balanced framework of data collection, and a signal-to-noise ratio (S/N ratio) analysis that serves the objective function to be optimized within the experimental domain.

The main feature of orthogonal array experiments is that it allows considerable optimization of the experimental data collection stage by reducing the experimental runs by several orders of magnitude, when compared to testing all possible combinations of the control factors. Unfortunately, the Taguchi method does not offer enough flexibility to readily identify and consider multiple optimum options that satisfy a required level of performance. In addition, this method does not provide an estimation of the magnitude for the predicted optimum response. Also, this method alone is not sufficient for multi-objective optimization problems. To overcome this limitation, authors have coupled the Taguchi method with Grey relational analysis for multi-objective optimization studies (Tarng, Juang, and Chang 2002; Lim S-H 2006; Datta, Bandyopadhyay, and Pal 2008). Yet, this combined method still does not predict the magnitude of the optimum responses nor it enables multiple optimum alternatives to be easily identified without exhaustive analysis of S/N ratios.

2.5. References

Abbas, S., M. L. Nehdi, and M. A. Saleem. 2016. 'Ultra-High Performance Concrete: Mechanical Performance, Durability, Sustainability and Implementation Challenges', *International Journal of Concrete Structures and Materials*, 10: 271-95.

Abdalla, Jamal A., Mousa F. Attom, and Rami Hawileh. 2015. 'Prediction of minimum factor of safety against slope failure in clayey soils using artificial neural network', *Environmental Earth Sciences*, 73: 5463-77.

Abuodeh, Omar R., Jamal A. Abdalla, and Rami A. Hawileh. 2020. 'Assessment of compressive strength of Ultra-high Performance Concrete using deep machine learning techniques', *Applied Soft Computing*, 95: 106552.

ACI-Committee-212. 1963. "Admixtures for Concrete." In *ACI Journal Proceedings*.

ACI-Committee-239. 2018. "Ultra-High Performance Concrete: An Emerging Technology Report." In, 21.

Alford, N. McN, and J. D. Birchall. 1984. 'The Properties and Potential Applications of Macro-Defect-Free Cement', *MRS Proceedings*, 42: 265.

Allwood, Julian M., Michael F. Ashby, Timothy G. Gutowski, and Ernst Worrell. 2013. 'Material efficiency: providing material services with less material production', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371: 20120496.

Alsalman, Ali, Canh N. Dang, and W. Micah Hale. 2017. 'Development of ultra-high performance concrete with locally available materials', *Construction and Building Materials*, 133: 135-45.

Bache, H H. 1987. 'Introduction to compact reinforced composite', *Nordic concrete research*: 19-33.

Brinkman, Leah, and Sabbie A Miller. 2021. 'Environmental impacts and environmental justice implications of supplementary cementitious materials for use in concrete', *Environmental Research: Infrastructure and Sustainability*, 1: 025003.

Brunauer, Stephen, Jan Skalny, Ivan Odler, and Marvin Yudenfreund. 1973. 'Hardened portland cement pastes of low porosity: VII. Further remarks about early hydration. Composition and surface area of tobermorite gel. Summary', *Cement and Concrete Research*, 3: 279-93.

Brunauer, Stephen, Marvin Yudenfreund, Ivan Odler, and Jan Skalny. 1973. 'Hardened portland cement pastes of low porosity VI. Mechanism of the hydration process', *Cement and Concrete Research*, 3: 129-47.

CARBONCURE. 2021. https://www.carboncure.com/concrete-corner/concrete-epds-the-4-things-you-should-know/.

Celik, Kemal, Cagla Meral, A. Petek Gursel, P. Kumar Mehta, Arpad Horvath, and Paulo J. M. Monteiro. 2015. 'Mechanical properties, durability, and life-cycle assessment of self-consolidating concrete mixtures made with blended portland cements containing fly ash and limestone powder', *Cement and Concrete Composites*, 56: 59-72.

Cheyrezy, Marcel, Vincent Maret, and Laurent Frouin. 1995. 'Microstructural analysis of RPC (Reactive Powder Concrete)', *Cement and Concrete Research*, 25: 1491-500.

Damineli, Bruno L., Fernanda M. Kemeid, Patricia S. Aguiar, and Vanderley M. John. 2010. 'Measuring the eco-efficiency of cement use', *Cement and Concrete Composites*, 32: 555-62.

Das, S.K. 2013. 'Artificial neural networks in geotechnical engineering: Modeling and application issues. metaheuristics in water.' in (Geotech. Transp. Eng.).

Datta, Saurav, Asish Bandyopadhyay, and Pradip Kumar Pal. 2008. 'Grey-based taguchi method for optimization of bead geometry in submerged arc bead-on-plate welding', *The International Journal of Advanced Manufacturing Technology*, 39: 1136-43.

de Larrard, F., and T. Sedran. 1994. 'Optimization of ultra-high-performance concrete by the use of a packing model', *Cement and Concrete Research*, 24: 997-1009.

Deng, Fangming, Yigang He, Shuangxi Zhou, Yun Yu, Haigen Cheng, and Xiang Wu. 2018. 'Compressive strength prediction of recycled concrete based on deep learning', *Construction and Building Materials*, 175: 562-69.

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

Deshpande, Neela, Shreenivas Londhe, and Sushma Kulkarni. 2014. 'Modeling compressive strength of recycled aggregate concrete by Artificial Neural Network, Model Tree and Non-linear Regression', *International Journal of Sustainable Built Environment*, 3: 187-98.

Fantilli, Alessandro P., and Bernardino Chiaia. 2013. 'Eco-mechanical performances of cement-based materials: An application to self-consolidating concrete', *Construction and Building Materials*, 40: 189-96.

Flower, David J. M., and Jay G. Sanjayan. 2007. 'Green house gas emissions due to concrete manufacture', *The International Journal of Life Cycle Assessment*, 12: 282.

Ghafari, E., M. Bandarabadi, H. Costa, and E. Júlio. 2012. 'Design of UHPC using artificial neural networks.' in A. M. Brandt, J. Olek, M. A. Glinicki and C. K. Y. Leung (eds.), *Brittle Matrix Composites 10* (Woodhead Publishing).

Ghafari, Ehsan, Mojtaba Bandarabadi, Hugo Costa, and Eduardo Júlio. 2015. 'Prediction of Fresh and Hardened State Properties of UHPC: Comparative Study of Statistical Mixture Design and an Artificial Neural Network Model', *Journal of Materials in Civil Engineering*, 27: 04015017.

Ghafari, Ehsan, Hugo Costa, and Eduardo Júlio. 2015. 'Statistical mixture design approach for eco-efficient UHPC', *Cement and Concrete Composites*, 55: 17-25.

Graybeal, Benjamin A. 2006a. 'Material Property Characterization of Ultra-High Performance Concrete'.

———. 2006b. 'Structural Behavior of Ultra-High Performance Concrete Prestressed I-Girders'.

Graybeal, Benjamin A., and Brenton Stone. 2012. 'Compression Response of a Rapid-Strengthening Ultra-High Performance Concrete Formulation'.

Gursel, Aysegul Petek, Helena Maryman, and Claudia Ostertag. 2016. 'A life-cycle approach to environmental, mechanical, and durability properties of “green” concrete mixes with rice husk ash', *Journal of Cleaner Production*, 112: 823-36.

Horvath, Arpad. 2004. 'CONSTRUCTION MATERIALS AND THE ENVIRONMENT', *Annual Review of Environment and Resources*, 29: 181-204.

Ibrahim, Mustapha A., Maen Farhat, Mohsen A. Issa, and Jessica Amanda Hasse. 2017. 'Effect of Material Constituents on Mechanical and Fracture Mechanics Properties of Ultra-High-Performance Concrete', *ACI Materials Journal*, 114: 453-65.

Irizarry, R.A. 2019. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R* (Chapman and Hall/CRC).

J.A. Abdalla, M.F. Attom, R. Hawileh. 2012. "Artificial neural network prediction of factor of safety of slope stability of soils." In *The 14th International Conference on Computing in Civil and Building Engineering*. Moscow, Russia.

JJ Park, ST Kang, KT Koh, SW Kim. 2008. "Influence of the ingredients on the compressive strength of UHPC as a fundamental study to optimize the mixing proportion." In *International Symposium Ultra High Performance Concrete (UHPC)*. Kassel, Germany.

Kay Wille, Antoine E. Naaman, and J. Parra-Montesinos Gustavo. 2011. 'Ultra-High Performance Concrete with Compressive Strength Exceeding 150 MPa (22 ksi): A Simpler Way', *ACI Materials Journal*, 108.

Kendall, K., A. J. Howard, James Derek Birchall, P. L. Pratt, B. A. Proctor, S. A. Jefferis, Peter Bernhard Hirsch, James Derek Birchall, D. D. Double, Anthony Kelly, G. K. Moir, and C. D. Pomeroy. 1983. 'The relation between porosity, microstructure and strength, and the approach to advanced cement-based materials', *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 310: 139-53.

Kim, Taehwan, M. Tyler Ley, Shinhyu Kang, Jeffrey M. Davis, Seokhyeon Kim, and Pouya Amrollahi. 2020. 'Using particle composition of fly ash to predict concrete strength and electrical resistivity', *Cement and Concrete Composites*, 107: 103493.

Kourehpaz, Pouria, and Sabbie A. Miller. 2019. 'Eco-efficient design indices for reinforced concrete members', *Materials and Structures*, 52: 96.

Li, P. P., Q. L. Yu, and H. J. H. Brouwers. 2018. 'Effect of coarse basalt aggregates on the properties of Ultra-high Performance Concrete (UHPC)', *Construction and Building Materials*, 170: 649-59.

Lim S-H, Lee C-M, Chung WJ 2006. 'A study on the optimal cutting condition of a high speed feeding type laser cutting machine by using Taguchi method', *International Journal of Precision Engineering and Manufacturing*, 7: 18-23.

Liu, Rui, A. Durham Stephan, L. Rens Kevin, and Anu Ramaswami. 2012. 'Optimization of Cementitious Material Content for Sustainable Concrete Mixtures', *Journal of Materials in Civil Engineering*, 24: 745-53.

Ma, J., M. Orgass, F. Dehn, D. Schmidt, and N.V. Tue. 2004. "Comparative investigations on ultra-high performance concrete with and without coarse aggregates." In *International Symposium on Ultra High Performance Concrete*, 205-12. Kassel, Germany.

Mehta, P. Kumar. 2009. 'Global Concrete Industry Sustainability', *Concrete International*, 31.

Miller, Sabbie A., Paulo J. M. Monteiro, Claudia P. Ostertag, and Arpad Horvath. 2016a. 'Comparison indices for design and proportioning of concrete mixtures taking environmental impacts into account', *Cement and Concrete Composites*, 68: 131-43.

———. 2016b. 'Concrete mixture proportioning for desired strength and reduced global warming potential', *Construction and Building Materials*, 128: 410-21.

Mohammed, Ahmed, Serwan Rafiq, Parveen Sihag, Rawaz Kurda, Wael Mahmood, Kawan Ghafor, and Warzer Sarwar. 2020. 'ANN, M5P-tree and nonlinear regression approaches with statistical evaluations to predict the compressive strength of cement-based mortar modified with fly ash', *Journal of Materials Research and Technology*, 9: 12416-27.

Naderpour, Hosein, Amir Hossein Rafiean, and Pouyan Fakharian. 2018. 'Compressive strength prediction of environmentally friendly concrete using artificial neural networks', *Journal of Building Engineering*, 16: 213-19.

Nisbet, R., Elder, J., Miner, G. 2009. *Handbook of statistical analysis and data mining applications* (Academic Press).

Odler, Ivan, Julius Hagymassy, Edward E. Bodor, Marvin Yudenfreund, and Stephen Brunauer. 1972. 'Hardened portland cement pastes of low porosity IV. Surface area and pore structure', *Cement and Concrete Research*, 2: 577-89.

Odler, Ivan, Marvin Yudenfreund, Jan Skalny, and Stephen Brunauer. 1972. 'Hardened portland cement pastes of low porosity III. Degree of hydration. Expansion of paste. Total porosity', *Cement and Concrete Research*, 2: 463-80.

Purnell, P. 2012. 'Material Nature versus Structural Nurture: The Embodied Carbon of Fundamental Structural Elements', *Environmental Science & Technology*, 46: 454-61.

Ranjit, K.R. 2010. *A primer on the Taguchi method* (Society of Manufacturing Engineers).

rediscover_concrete. 2021. http://rediscoverconcrete.com/en/sustainability/reducing-our-footprint/environmental-product-declarations.html.

Ren, Qiang, Luchuan Ding, Xiaodi Dai, Zhengwu Jiang, and Geert De Schutter. 2021. 'Prediction of Compressive Strength of Concrete with Manufactured Sand by Ensemble Classification and Regression Tree Method', *Journal of Materials in Civil Engineering*, 33: 04021135.

Richard, Pierre, and Marcel Cheyrezy. 1995. 'Composition of reactive powder concretes', *Cement and Concrete Research*, 25: 1501-11.

Roux, N., C. Andrade, and M. A. Sanjuan. 1996. 'Experimental Study of Durability of Reactive Powder Concretes', *Journal of Materials in Civil Engineering*, 8: 1-6.

Rowlands, Hefin, Jiju Antony, and Graeme Knowles. 2000. 'An application of experimental design for process optimisation', *The TQM Magazine*, 12: 78-84.

Roy, D. M., G. R. Gouda, and A. Bobrowsky. 1972. 'Very high strength cement pastes prepared by hot pressing and other high pressure techniques', *Cement and Concrete Research*, 2: 349-66.

Roy, Della M., and G. R. Gouda. 1973. 'High strength generation in cement pastes', *Cement and Concrete Research*, 3: 807-20.

Russell, Henry G., and Benjamin A. Graybeal. 2013. 'Ultra-high performance concrete : a state-of-the-art report for the bridge community'.

Saadat, Mohsen, and Meysam Bayat. 2019. 'Prediction of the unconfined compressive strength of stabilised soil by Adaptive Neuro Fuzzy Inference System (ANFIS) and Non-Linear Regression (NLR)', *Geomechanics and Geoengineering*: 1-12.

Sadrossadat, Ehsan, Hakan Basarir, Ali Karrech, and Mohamed Elchalakani. 2021. 'Multi-objective mixture design and optimisation of steel fiber reinforced UHPC using machine learning algorithms and metaheuristics', *Engineering with Computers*.

Schmidt, M., and E. Fehling. 2005. 'Ultra-High-Performance Concrete: Research, Development and Application in Europe', *ACI Symposium Publication*, 228.

Solhmirzaei, Roya, Hadi Salehi, Venkatesh Kodur, and M. Z. Naser. 2020. 'Machine learning framework for predicting failure mode and shear capacity of ultra high performance concrete beams', *Engineering Structures*, 224: 111221.

Soliman, Nancy A., and Arezki Tagnit-Hamou. 2017. 'Using glass sand as an alternative for quartz sand in UHPC', *Construction and Building Materials*, 145: 243-52.

Sonja AAM, Fennis JCW, den Uijl Joop A. 2009. "The use of particle packing models to design ecological concrete." In. Netherlands: Delft University of Technology.

Talebinejad, I. ; Bassam, S.; Iranmanesh, A.; Shekarchizadeh, M. 2004. 'Optimizing Mix Proportions of Normal Weight Reactive Powder Concrete with Strengths of 200-350 MPa', *Proceedings of the International Symposium on UHPC, Kassel, Germany*: 133-41.

Tarng, Y. S., S. C. Juang, and C. H. Chang. 2002. 'The use of grey-based Taguchi methods to determine submerged arc welding process parameters in hardfacing', *Journal of Materials Processing Technology*, 128: 1-6.

Teichmann T, Schmidt M. . 2004. "Influence of the packing density of fine particles on structure, strength and durability of UHPC." In *International symposium on ultra high performance concrete*, edited by Fehling E In: Schmidt M, Geisenhanslüke C, editors. Kassel, Germany.

The_Concrete_Center. 2021. https://www.concretecentre.com/Codes/Environmental-Assessment/EPD.aspx.

W.B.C.S.D., I.E.A. 2009. 'Carbon Emission Reductions Up to 2050', *Cement Technology Roadmap*.

Wille, Kay, Antoine E. Naaman, and Gustavo J. Parra-Montesinos. 2011. 'Ultra-High Performance Concrete with Compressive Strength Exceeding 150 MPa (22 ksi): A Simpler Way', *ACI Materials Journal*, 108: 46-54.

Yagiz, S., E. A. Sezer, and C. Gokceoglu. 2012. 'Artificial neural networks and nonlinear regression techniques to assess the influence of slake durability cycles on the prediction of uniaxial compressive strength and modulus of elasticity for carbonate rocks', *International Journal for Numerical and Analytical Methods in Geomechanics*, 36: 1636-50.

Yazıcı, Halit. 2007. 'The effect of curing conditions on compressive strength of ultra high strength concrete with high volume mineral admixtures', *Building and Environment*, 42: 2083-89.

Yudenfreund, Marvin, Kaissar M. Hanna, Jan Skalny, Ivan Older, and Stephen Brunauer. 1972. 'Hardened Portland cement pastes of low porosity V. Compressive strength', *Cement and Concrete Research*, 2: 731-43.

Yudenfreund, Marvin, Ivan Odler, and Stephen Brunauer. 1972. 'Hardened portland cement pastes of low porosity I. Materials and experimental methods', *Cement and Concrete Research*, 2: 313-30.

Yudenfreund, Marvin, Jan Skalny, Raouf Sh Mikhail, and Stephen Brunauer. 1972. 'Hardened portland cement pastes of low porosity II. Exploratory studies. Dimensional changes', *Cement and Concrete Research*, 2: 331-48.

CHAPTER III.

EXPERIMENTAL PROGRAM: OPTIMIZED FRAMEWORK & PROTOCOLS

This chapter describes the strategic experimental framework established with orthogonal arrays (from the Taguchi method) and new test protocols developed for expeditious and efficient data collection. Materials, methods and specimen preparation are described herein. Furthermore, the experimental results obtained are presented and will be used as inputs to feed the machine learning models developed in the subsequent chapters.

3.1. Multi-Phase Experimental Framework Designed with Orthogonal Arrays

This study entails a multi-scale, multi-phase mix design approach to optimize the compressive strength of UHPC. It consists of two main phases, A and B, for which the objectives are, respectively: A) Maximize the compressive strength at the binder scale; B) Optimize the compressive strength at the mortar scale considering the inclusion of fine aggregates. In UHPC mix design, binder selection typically involves microstructural packing and rheology analysis (Arora et al. 2018; Arora et al. 2019; Shi et al. 2021). Additionally, the optimization of compressive strengths in cementitious binders should involve porosity measurements to assess the normalized strength by porosity, which represents the actual strength of the solid structure in the matrix. This is relevant considering that the volume of capillary pores in cementitious pastes can be reduced by adding air detrainers during the mixing process or reducing the w/cm in design stages. Nevertheless, to maintain focus on the tools developed in this study and facilitate comparisons between phases, the compressive strengths obtained in Phase A are not further normalized by porosity nor involve microstructural packing or rheology analysis. Thus, at this

phase, the optimum binders are selected solely based on the maximum compressive strength predicted by the models developed herein.

The initial range of proportions defined for each constituent was chosen based on numerous studies done in the field (summarized by Carey et al. (Carey et al. 2020)). The content of SCMs was defined on a % by weight (wt.) of cement replaced basis. The max content of slag, microsilica and fly ash were fixed at 60%, 20% and 15%, respectively. Although higher contents of fly ash can greatly improve mechanical performance in later ages, the max content used in this study is kept low to differentiate these mixtures from high-volume fly ash concretes (HVFA). On the other hand, literature (Carey et al. 2020) indicates that optimum contents of silica fume in UHPC can be obtained with up to 25% cement replacement. However, densified silica fume (or microsilica) has a much lesser role as a filler due to the increased size of the particles. Phase A included an additional iteration [A(II)] to increase the number of data points in a particular region of interest in the experimental domain. Considering that this study entails maximizing the compressive strength, preliminary models obtained in Phase A(I) were used to define this region by estimating the range of SCM contents that produces mixtures with compressive strengths in the top 20 percentile. Consequently, the maximum contents of slag, microsilica and fly ash for Phase A(II) were fixed in 52%, 10.5% and 15%, respectively. For Phase B, the maximum contents of ground quartz, concrete sand (silica sand) and crushed sand were fixed in 25% by wt. of cementitious binder replaced in the system. Phase B included an additional step [B(II)] to evaluate the optimum mixtures predicted by some of the models. In addition, this sub-phase serves to compare the results obtained following the test protocol developed in this study with the results obtained following the standard testing protocol for UHPC in ASTM C1856/1856M (ASTM 2017).

Fig. 2 shows the mix design phases and strategy defined in the study herein to maximize the compressive strength of UHPC mixtures.

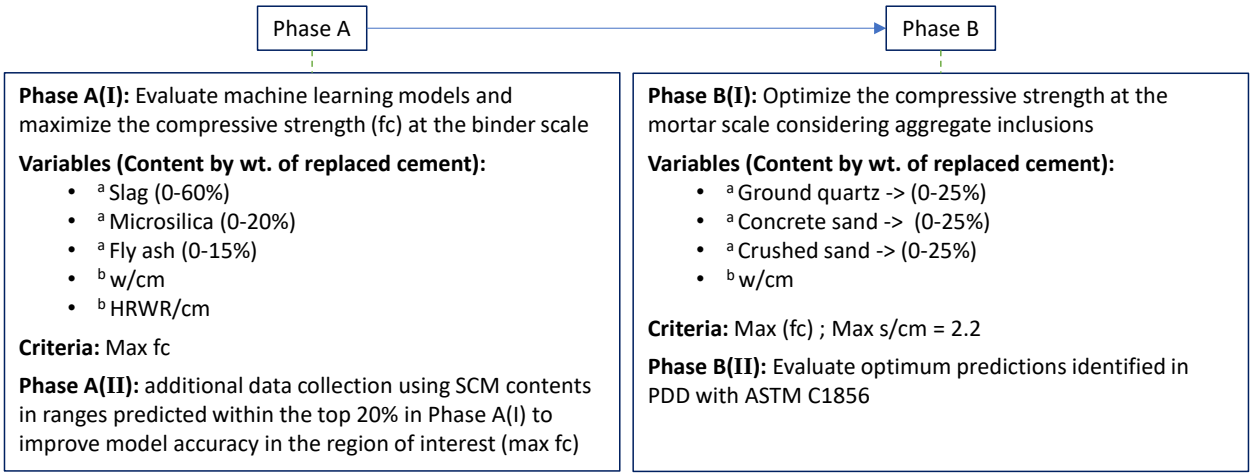
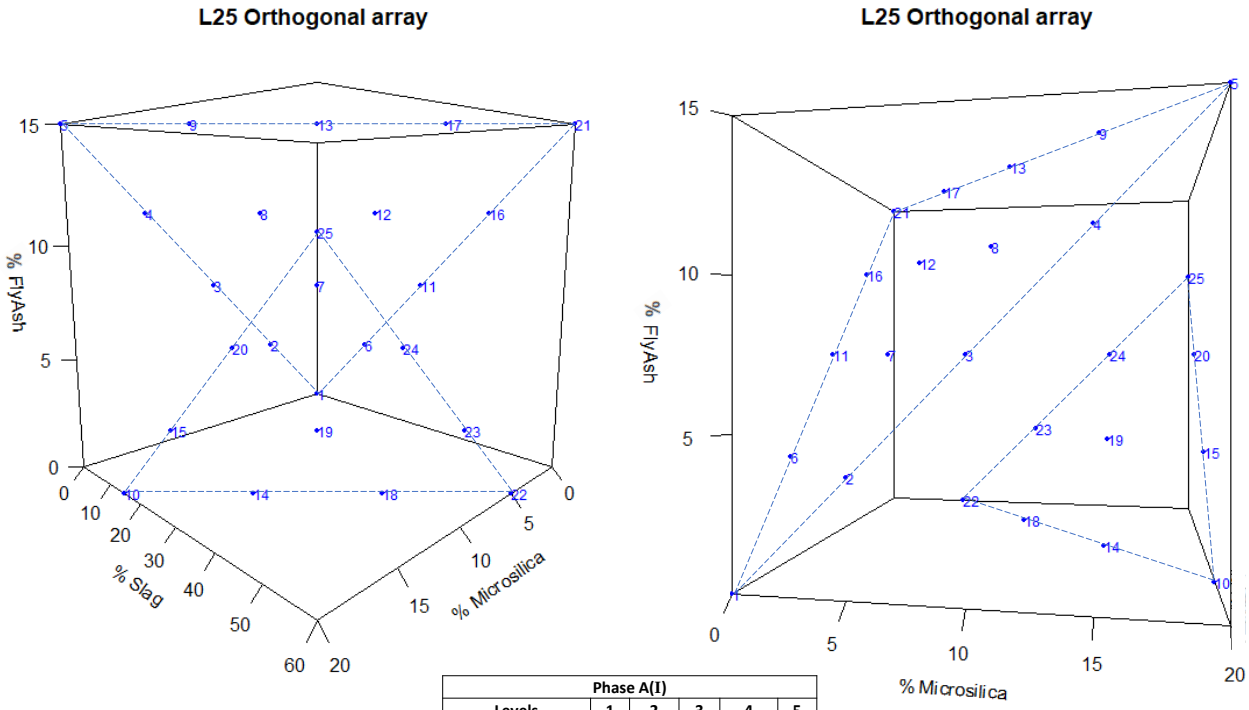


Fig. 2. Design and experimental phases of the optimization study

The experimental frameworks used for data collection in Phases A(I) and B(I) were designed using a L25 orthogonal array (5^2 levels) with 3 variables. Meanwhile, Phase A(II) is a complementary experimental phase, designed with a smaller array in L16 (4^2 levels) while maintaining the design variables. Table A1 through Table A5 in APPENDIX A contain detailed information on the orthogonal arrays used and the corresponding design levels for each phase of this study. Fig. 3 shows the design matrix used in Phase A(I), where the mixture compositions tested experimentally were defined based on the content of Portland cement replaced with SCMs on a % by weight basis.

The arrangement of orthogonal arrays is ideal for machine learning modeling. It reduces the chances of overtraining without the need to resort to techniques such as feature selection or assessing variable importance parameters. While these methods tend to improve predictive

performance by selecting the absolute necessary features to maximize model accuracy, they are not vital to prevent overtraining in orthogonal array datasets given that each row differs from the others regardless of the number of variables/columns used.



Phase A(I)						
Levels		1	2	3	4	5
Features	Slag	0	15	30	45	60
	Microsilica	0	5	10	15	20
	Fly Ash	0	3.75	7.5	11.25	15

Fig. 3. L25 orthogonal array used to design the experimental framework for Phase A(I) (Table A1 - APPENDIX A). The coordinates for each mix design number indicate the content % by wt. of SCMs replacing cement, considering only the dry ingredients content.

3.2. Surrogate Samples Suitable for Collecting Large Data on Compressive Strength of Binders and Mortars with Strengths up to UHPC Levels

Motivated by the resource and time-consuming nature of available standard methods, this study was performed using a novel test protocol that facilitates extensive data collection to feed machine learning models. A test protocol using cylindrical specimens of reduced dimensions was

evaluated for testing cement pastes and mortars with strengths up to the level of UHPC materials. This protocol involved a fast production process of 23 x 46 mm cylinders, cast in silicone molds as shown in Fig. 4. Later, both top ends of the cylinders were cut to obtain smooth surfaces and the aforementioned dimensions.

Fig. 4. Silicone molds used to cast binders and mortars in this study

Capping systems evaluated for this protocol consisted of commercially available felt cushion pads (often designated as heavy duty pads) and neoprene pads, cut out of sheet rolls. Fig. 5 shows these capping systems, along with specimen dimensions.

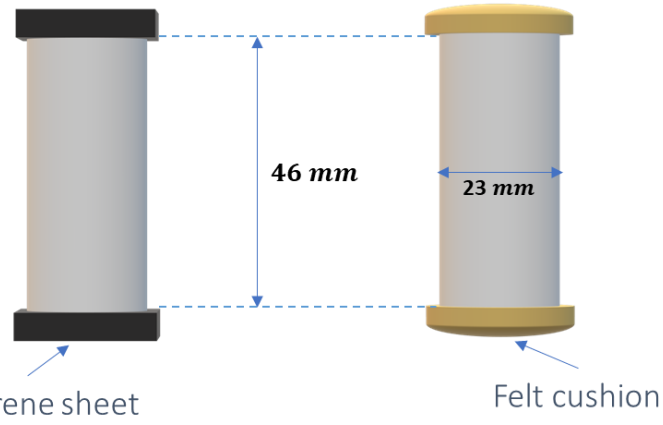


Fig. 5. Capping pads used with new test protocol: neoprene square pads (left) vs felt cushion circular pads (right)

This method permits large datasets to be generated in short periods of time. The entire mixing and casting time is approximately seven times faster than the standardized method in ASTM C1856/1856M involving 76 x 152 mm cylinders. Additionally, the specimens used in this study consume approximately forty times less volume of materials when compared to the cylinders used in the aforementioned standard. Fig. 6 illustrates specimens cast during Phase A(I), which were easily produced in approximately one week.

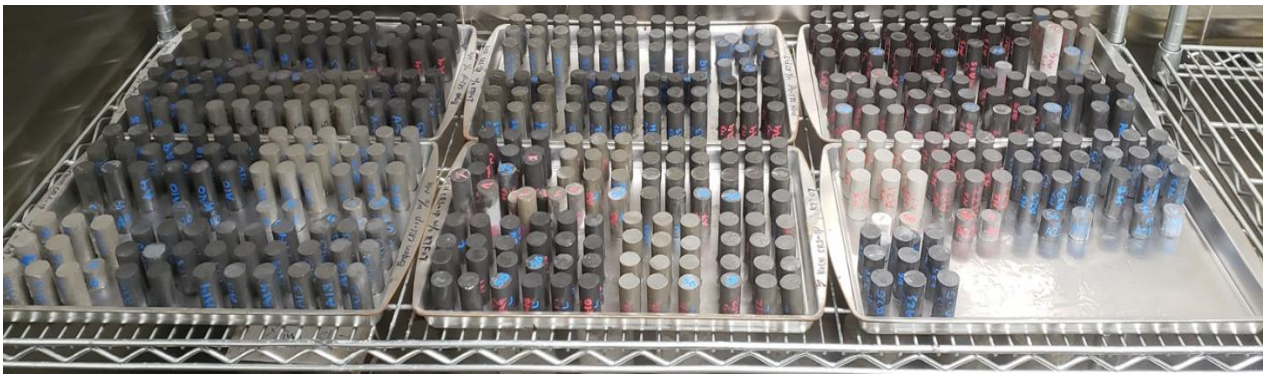


Fig. 6. Specimens cast during Phase A(I)

Casting large numbers of mixtures involving 76 x 152 mm cylinders can become cumbersome, exhausting, and time- and resource-intensive. Fig. 7 illustrates preliminary results suggesting that the proposed test protocol can be a viable option to test UHPC pastes and mortars when the number of experiments is considerably high.

As shown in Fig. 7, when specimens were capped using felt cushion pads, great agreement was observed between the new test protocol and the reference standard ASTM C39/C39M (ASTM 2021), for normal strength, and ASTM C1856/1856M, for strength levels up to 125 MPa (18,100 psi). On the other hand, neoprene pads were typically destroyed when testing mixtures with strengths exceeding 50 MPa. For this reason, felt cushion pads were used for all the remaining

experiments in this study to generate accurate data to feed the models developed in the subsequent chapters.

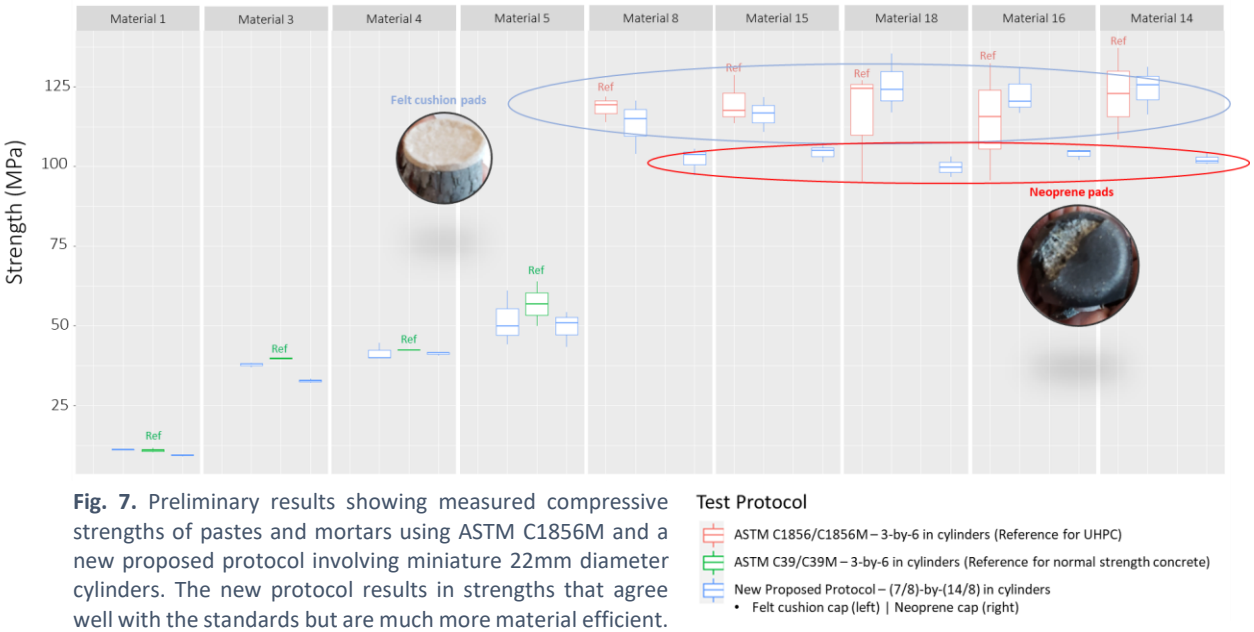


Fig. 7. Preliminary results showing measured compressive strengths of pastes and mortars using ASTM C1856M and a new proposed protocol involving miniature 22mm diameter cylinders. The new protocol results in strengths that agree well with the standards but are much more material efficient.

3.3. Materials and Specimen Preparation

UHPC pastes and mortars were prepared in a 20-Quart high shear stand mixer with a stainless-steel container. The w/cm was initially targeted at 0.20, however, adjustments in the HRWR content were necessary to achieve spread values in the range of a self-consolidating material. This resulted in w/cm ranging between 0.19 and 0.22. The HRWR content was specified as the minimum needed to achieve spread values of at least 20 cm. For cementitious pastes produced in Phase A, the maximum HRWR content was controlled by visually inspecting the fresh binder for evidence of bleeding. On the other hand, for mortars produced during Phase B, the maximum HRWR content was controlled by visually inspecting the fresh mortar for evidence of bleeding or segregation. Mixtures in Phase B that exhibited evidence of segregation were discarded and re-

batched with reduced HRWR contents. Overall, the HRWR content appears to not have a significant impact on the compressive strength for the range of contents used in this work (Fig. A1 - APPENDIX A), where little to no correlation is observed between compressive strength and measured flow values. The cementitious materials used consisted of Portland cement type I/II (composition: portland cement clinker 90%–96%, gypsum 2%–5%, calcium carbonate 0%–5%, blast furnace slag 0%–5%), densified silica fume described as microsilica (composition: microsilica content 100%, amorphous silicon dioxide 92%-98%), blast furnace slag (composition: slag 100%, calcium oxide 30%-50%, magnesium oxide 0%-20%, crystalline silica <1%) and class F fly ash (silicon dioxide 55%, aluminum oxide 26%, ferric oxide 7%, calcium oxide 9%, magnesium oxide 2%, sulfite ion 1%). The liquids involved in the mix designs consisted of tap water (City-of-College-Station 2018) and a polycarboxylate based HRWR. For mortar mixtures, fine aggregates were added; namely, a manufactured crushed sand, a natural silica sand and a ground quartz powder. The total mixing time for each batch was 25 min, consisting of 12 min pre-mixing the dry ingredients to ensure a uniform distribution of the fine particles, 1 min of adding water and HRWR to ensure uniform dispersion of the liquids in the matrix, 8 min mixing in medium speed, 2 min rest, and 2 min of high speed mixing (maximum) to improve homogeneity while simultaneously avoiding adding entrapped air to the system. After mixing, the material is poured into a flow cone to measure the spread value, following ASTM C1856/1856M (ASTM 2017). The cone is filled to the top and lifted, allowing the paste to flow. After 2 min, the spread value is obtained by measuring and averaging two diameters at a right angle. Finally, three specimens per mixture (plus a few spare samples) were cast in 23 x 46 mm cylinders to assess the compressive strength. The exposed ends were covered in plastic sheet to prevent moisture loss.

All samples were left to cure in their molds for 24 h at room temperature, set to 73°F±2°F (23°C) by default. After 24 h, all samples were demolded and moved to a controlled chamber with 98% relative humidity at a temperature of 23°C for a controlled curing process, until tested.

3.4. Experimental Results

Table 2 illustrates the experimental results obtained for the binders produced and tested during Phase A. For each mixture design, the average compressive strength (in MPa), standard deviation (in MPa) and measured spread flow (in cm) are presented. Furthermore, the fracture type observed for each specimen was determined according to ASTM C39/39M and the results are displayed in the APPENDIX A (Fig. A1).

Meanwhile, Table 3 illustrates the experimental results obtained for the mortars produced and tested during Phase B. These results are used to train the models discussed in the subsequent chapter.

Table 2. Experimental Results for Phase A

Mixture #	SCM replacing cement (% by wt.)			w/cm	HRWR /cm	flow (cm)	testing at age 1 day		testing at age 56 days	
	Slag	Microsilica	Fly Ash				avg fc (MPa)	sd (MPa)	avg fc (MPa)	sd (MPa)
A1	0	0	0	0.19	1.2%	26	63.3	2.4	86.4	2.8
A2	0	5	3.75	0.20	1.7%	31	79.6	3.6	110.5	6.2
A3	0	10	7.5	0.20	2.3%	31.5	71.2	1.7	97.3	2.6
A4	0	15	11.25	0.20	2.9%	28.5	36.9	3.7	94.2	2.1
A5	0	20	15	0.21	3.4%	25	25.7	3.0	68.9	13.7
A6	15	0	3.75	0.19	1.4%	36.5	56.9	4.8	113.7	2.8
A7	15	5	7.5	0.20	1.7%	36	45.1	3.3	113.8	10.8
A8	15	10	11.25	0.20	2.3%	34	32.1	3.4	96.1	7.9
A9	15	15	15	0.20	2.9%	31	22.8	2.4	91.7	4.6
A10	15	20	0	0.21	3.5%	25	35.1	1.7	93.4	4.8
A11	30	0	7.5	0.19	1.3%	38	43.8	11.8	104.0	11.0
A12	30	5	11.25	0.19	1.4%	37	36.7	6.9	115.6	5.2
A13	30	10	15	0.20	2.0%	35	29.3	2.6	96.4	4.2
A14	30	15	0	0.21	3.3%	30.5	24.4	1.4	100.4	7.4
A15	30	20	3.75	0.21	3.7%	26	12.6	0.3	90.4	7.6
A16	45	0	11.25	0.19	1.0%	35	28.2	3.5	76.9	19.5
A17	45	5	15	0.19	1.3%	36.5	30.1	1.0	111.5	12.0
A18	45	10	0	0.20	2.0%	33	33.7	1.7	109.7	2.3
A19	45	15	3.75	0.20	2.3%	31	20.6	0.8	88.1	12.8
A20	45	20	7.5	0.22	5.1%	31	2.5	0.3	69.4	6.6
A21	60	0	15	0.20	1.6%	38	8.7	0.1	106.5	7.2
A22	60	5	0	0.19	1.4%	32.5	32.0	1.8	96.7	8.5
A23	60	10	3.75	0.19	1.5%	29.5	21.5	0.5	90.5	11.3
A24	60	15	7.5	0.21	4.0%	33	4.3	0.4	75.1	9.2
A25	60	20	11.25	0.21	3.3%	29	4.7	0.9	72.9	7.1
A26	0	3.5	3.75	0.20	1.1%	40	93.6	2.5	119.4	9.2
A27	0	7	7.5	0.20	1.6%	40	84.5	4.0	122.4	3.6
A28	0	10.5	15	0.20	1.6%	40	68.6	2.5	120.7	10.2
A29	17.33	0	3.75	0.20	0.8%	27	76.6	3.0	121.5	5.2
A30	17.33	3.5	7.5	0.20	1.0%	38	77.8	2.9	121.4	5.1
A31	17.33	7	15	0.20	1.1%	40	67.2	2.1	123.1	2.3
A32	17.33	10.5	0	0.20	1.6%	38	72.0	3.9	120.8	7.8
A33	34.67	0	7.5	0.20	0.8%	30	63.7	4.7	126.6	7.4
A34	34.67	3.5	15	0.20	1.2%	40	54.7	2.0	133.7	8.2
A35	34.67	7	0	0.20	1.5%	40	63.0	1.7	123.5	9.3
A36	34.67	10.5	3.75	0.20	1.1%	26	53.8	2.1	115.9	5.4
A37	52	0	15	0.20	1.1%	40	41.4	0.8	127.9	5.1
A38	52	3.5	0	0.20	1.1%	40	52.3	3.2	123.3	14.2
A39	52	7	3.75	0.20	1.1%	35	46.0	2.0	113.2	9.9
A40	52	10.5	7.5	0.20	1.1%	40	35.4	1.9	121.6	8.5

Mixture # - mix design number
avg fc – average compressive strength (MPa)
sd – standard deviation (MPa)

Table 3. Experimental results from Phase B

Mixture #	Aggregates replacing cementitious (% by wt.)			w/cm	HRWR/ cm	flow (cm)	testing at age 56 days	
	Ground quartz	Concrete sand	Crushed sand				avg fc (MPa)	sd (MPa)
Mix B1	0	0	0	0.20	0.97%	38	114.4	5.4
Mix B2	0	6.25	6.25	0.20	0.93%	29	99.7	3.8
Mix B3	0	12.5	12.5	0.20	0.83%	20	109.5	3.3
Mix B4	0	18.75	18.75	0.20	0.90%	26	94.3	3.3
Mix B5	0	25	25	0.20	0.97%	26	74.7	5.2
Mix B6	6.25	0	6.25	0.20	0.98%	23	112.8	2.5
Mix B7	6.25	6.25	12.5	0.20	1.01%	23	87.9	2.9
Mix B8	6.25	12.5	18.75	0.20	1.11%	31	75.1	3.4
Mix B9	6.25	18.75	25	0.20	1.24%	21	74.4	0.8
Mix B10	6.25	25	0	0.20	1.07%	20	106.4	6.9
Mix B11	12.5	0	12.5	0.20	1.22%	27	106.2	4.8
Mix B12	12.5	6.25	18.75	0.21	2.16%	26	86.1	2.7
Mix B13	12.5	12.5	25	0.21	1.94%	32	103.3	3.7
Mix B14	12.5	18.75	0	0.20	1.37%	36	121.1	0.7
Mix B15	12.5	25	6.25	0.20	0.15%	34	107.6	3.4
Mix B16	18.75	0	18.75	0.20	1.55%	34	114.4	2.8
Mix B17	18.75	6.25	25	0.21	2.39%	30	103.5	2.7
Mix B18	18.75	12.5	0	0.21	1.94%	36	119.3	1.8
Mix B19	18.75	18.75	6.25	0.21	2.19%	33	110.1	4.1
Mix B20	18.75	25	12.5	0.21	3.01%	26.5	61.9	4.8
Mix B21	25	0	25	0.23	2.71%	26.5	76.9	9.5
Mix B22	25	6.25	0	0.24	1.61%	36	125.8	9.1
Mix B23	25	12.5	6.25	0.23	2.11%	30	89.2	3.8
Mix B24	25	18.75	12.5	0.25	2.52%	29	69.0	5.8
Mix B25	25	25	18.75	0.30	5.82%	26	55.1	4.9

3.5. Summary of Important Outcomes

This section was motivated by the resource and time-consuming nature of available standard methods. A strategic experimental framework was established with orthogonal arrays (from the Taguchi method) and a new test protocol was developed to facilitate extensive data collection to feed machine learning models in an expeditious and efficient way. The main findings from this part of the work are:

- This method allows large datasets to be generated in short periods of time. Mixing and casting processes are approximately seven times faster compared to the standardized method in ASTM C1856/1856M involving 76 x 152 mm cylinders.
- The specimens used in this study consume approximately forty times less volume of materials when compared to the cylinders used in the aforementioned standard.
- The new test protocol developed for pastes and mortars using reduced size samples and simplified end specimen conditions agrees well with standard methods up to strengths of 125 MPa. Further evaluation of the end-specimen conditions is required to improve this test protocol for materials with strengths over 125 MPa.

This protocol encourages innovation in mixture design with new materials in a constantly growing industry such as UHPC. Considering that nanomaterials such as nanosilica and carbon nanofibers are often costly and somewhat challenging to disperse, this protocol provides an efficient way to explore different concentrations and dispersion methods while quickly assessing the effect of these inclusions on the compressive strength of mixtures. This assuming that compressive strength is used as a quick indicator of dispersion efficiency, considering that poorly dispersed fiber reinforced SCCs tend to experience decrease in strength with respect to control samples due to the presence of fiber clumps.

3.6. References

Arora, Aashay, Matthew Aguayo, Hannah Hansen, Cesar Castro, Erin Federspiel, Barzin Mobasher, and Narayanan Neithalath. 2018. 'Microstructural packing- and rheology-based binder selection and characterization for Ultra-high Performance Concrete (UHPC)', *Cement and Concrete Research*, 103: 179-90.

Arora, Aashay, Asim Almujaddidi, Farrokh Kianmofrad, Barzin Mobasher, and Narayanan Neithalath. 2019. 'Material design of economical ultra-high performance concrete (UHPC) and evaluation of their properties', *Cement and Concrete Composites*, 104: 103346.

ASTM. 2017. "ASTM C1856 / C1856M-17 Standard Practice for Fabricating and Testing Specimens of Ultra-High Performance Concrete." In. West Conshohocken, PA: ASTM International.

———. 2021. "ASTM C39 / C39M-21, Standard Test Method for Compressive Strength of Cylindrical Concrete Specimens." In. West Conshohocken, PA: ASTM International.

Carey, A. S., I. L. Howard, D. A. Scott, R. D. Moser, J. Shannon, and A. Knizley. 2020. 'Impact of Materials, Proportioning, and Curing on Ultra-High-Performance Concrete Properties', *ACI Materials Journal*, 117: 213-22.

City-of-College-Station. 2018. "2018 Drinking Water Quality Report." In.

Shi, Ye, Guangcheng Long, Xiaohui Zen, Youjun Xie, and Taoping Shang. 2021. 'Design of binder system of eco-efficient UHPC based on physical packing and chemical effect optimization', *Construction and Building Materials*, 274: 121382.

CHAPTER IV.

MACHINE LEARNING MODELS

4.1. Theory and Methodology

Machine learning is a subset of AI (the science and engineering of making intelligent machines, especially intelligent computer programs) and computer science. In particular, machine learning focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy (IBM 2020). The computations performed in this study were done in the data analysis software "R" using the machine learning algorithms built in the caret package (Kuhn 2008).

The ability of a model to minimize bias and variance is a tradeoff. Proper understanding of these errors is vital to build accurate models that avoid overfitting and oversmoothing. In machine learning, bias is represented by the inability of a model to capture the true relationship (actual outcomes) in the dataset, whereas variance represents the difference in fits (variability of model prediction) across different datasets (training vs testing/validation sets). Models with high bias tend to oversimplify the existing trends (very little attention to the training data), leading to high error on training and test data (also known as oversmoothing). Models with high variance pay a lot of attention to training data and lack the ability to generalize on unseen datasets. As a result, such models perform very well on training data, while exhibiting very high error on test data (also known as overfitting). Ideal models have low bias and can accurately model the true relationship, while also showing low variance by producing consistent predictions across different datasets. In machine learning, different algorithms use different methods to achieve balanced bias-variance

tradeoffs, aiming for developing models with low bias and low variance, and thus, improved predictive performance. The objective consists in reducing the remaining error to as close as possible to the irreducible error, which is associated with the noise in the experimental dataset (standard deviation).

To optimize the mixture proportions of UHPC binders and mortars, two different machine learning algorithms were considered: kNN and Random Forest. kNN is a supervised-learning machine learning algorithm within the set of instance-based learning methods. This algorithm estimates conditional probabilities by averaging the properties of “k” nearest neighbors (Irizarry 2019). Larger values of “k” lead to smoother estimates, while smaller values of “k” provide more flexible and sinuous estimates. When using this model, the risk of overtraining is often associated with small values of “k”, where the accuracy is significantly higher in the train set than in the test set. For instance, overtraining is at its worst when k is equal to 1. In this case, each estimate is obtained using only the information corresponding to that same point (i.e., each point is its own nearest neighbor). On the other hand, very large numbers of “k” neighbors affect the flexibility of the model, in which several points are averaged to compute each single estimated conditional probability. This often leads to oversmoothing, where the accuracy in both training and testing sets drop to unsatisfactory levels. A commonly used technique to optimize the value of “k” nearest neighbors (and thus achieve the best possible bias-variance tradeoff) is the k-fold cross validation method (Stone 1974; Mosteller and Tukey 1968; Mosteller and Wallace 1963; Larson 1931). In this method, the training data is split into k-non overlapping folds that store data for the validation sets. The model works in each fold individually by fitting the data in the training set, predicting the outcome in the validation set and storing the RMSE for a range of possible “k”

values. After performing an iteration in each fold, the average RMSE for each "k" across all folds is obtained to identify the optimum "k" that minimizes the RMSE across all validation sets. After establishing the model with the optimum "k" value, the test set is used only to estimate the predictive performance of the model when presented to new independent datasets containing data that have not been "seen" by the model during training. Fig. 8a) illustrates 5 initial folds randomly generated during Phase A(I) to split training and testing datasets, prior to cross validation. Mixtures that were "blind" to Fold 1 in this original split were stored in a test set as shown in Fig. 8b). With this split, 16% of the data (mixtures #13, 14, 15 and 25) were stored in the testing set, while 84% of the data (remaining 21 mixtures) were assigned for training, where validation and training subsets were randomly generated to perform the cross-validation process described above.

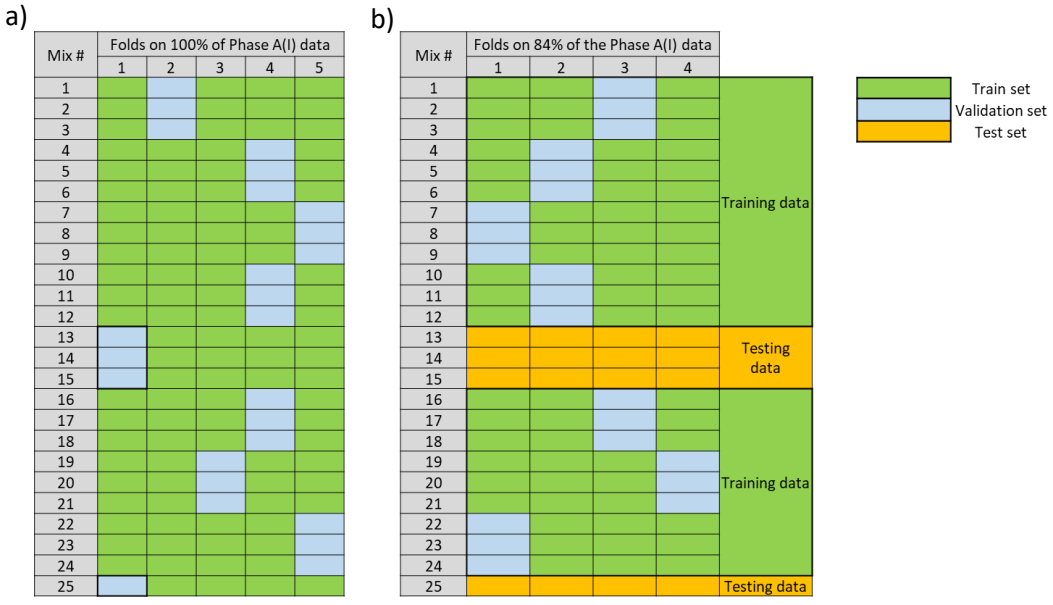


Fig. 8. Folds randomly created in performing a k-fold cross validation method used to train and test the models. Five folds were originally created in the entire dataset of Phase A(I). Mixtures that were "blind" to Fold 1 were stored in the test set. Four new folds were randomly generated with the remaining data to create the required dataset (training and validation sets) for cross validation to optimize the tuning parameters and generate the model

Another strategy established in this study, aiming to prevent data leakage, consisted of grouping the data of individual specimens by mix design during cross-validation. This was performed using the GroupKFold function, which is a variation of the k-fold cross-validation with non-overlapping groups. This method ensures that, when accessing the individual information of each specimen, the same mixture is not represented in both validation/testing and training sets. This permits evaluating additional models that access the individual response of each specimen to model the grouped response of each mix design without compromising the defined experimental unit (which in this case is each mix design), while avoiding data leakage occurrences, and thus preventing bias performance estimations.

Another popular algorithm in machine learning is the random forest, which combines several random regression (continuous) or decision (categorical) trees. Regression trees operate by recursively partitioning the feature/predictor space to estimate an outcome. To avoid overtraining, these models set minimum requirements before adding new partitions, which are based on: 1) level of improvement on residual sum of squares; 2) number of observations to be partitioned; and 3) number of observations in each partition (Irizarry 2019). These models are advantageous in data pre-processing efforts and data preparation considering that they do not require normalization or scaling of the data. On the other hand, these models are highly unstable given that small changes in the data can cause large change in the main structure. To overcome the shortcomings of regression trees, random forest models were developed to reduce instability and improve prediction performance. Random forest models average multiple regression trees (a forest of trees) constructed with randomness, using a bootstrap process to optimize the number of variables randomly sampled in each split. For simplicity purposes, random forest

models will be denoted as RF for the remainder of this work, although it is not a universal acronym such as kNN. Additionally, the test sets defined for the kNN models were maintained for the RF models to allow an unbiased evaluation of predictive performance and comparison between the different models.

A popular approach to improve predictive performance is to build ensemble models, which consists in creating a model that benefits from the best features from multiple models. To avoid overextending the evaluation of machine learning techniques, the simplest ensemble method is followed in this study by averaging the predictions obtained from the best-performing models, as discussed in the following section.

4.2. Regression Models

4.2.1. Developing Models: Variables and Datasets

In this study, various models were developed using different modeling techniques: kNN, RF and linear regression. The designation attributed to each model was established based on: a) whether the data were modeled using kNN, RF or linear models; b) whether the model was created using the average results for each mixture (avg) or accessing each individual specimen (sp) to model the grouped response; c) whether the model was trained using 80-84% of the orthogonal array dataset (train set) and evaluated on the remaining 16-20% (test set), or trained using the entire orthogonal array dataset (100%) and tested on a new testing set defined by the optimum mixtures (optimum test set) predicted by several of the generated models; and d) whether the model involved results from Phase A(I) or from the entire Phase A [A(I) + A(II)].

Models were developed in a similar manner for Phase B (described in APPENDIX B). All models were further evaluated in three stages: 1) using SCM replacement (slag, microsilica and fly ash)

as the only variables; 2) adding a 4th variable in w/cm; 3) adding a 5th variable in HRWR/cm. This intended to provide insight regarding the influence of small variations in w/cm and HRWR/cm on the predictive performance of the models. As mentioned earlier, the experimental w/cm ranged between 0.19 to 0.22 due to the adjustment required in the HRWR content to achieve spread values over 20cm. This is especially important for UHPC to allow proper consolidation considering that vibration is not recommended for self-consolidating concretes.

Phase A involved two iterations: A(I) and A(II). First for A(I), 25 mixtures (A1-A25) were designed and tested with varying combinations of SCM contents, following the L25 orthogonal array described in Fig. 3. Forty eight models were developed based on the data collected at this stage and are indicated in Table 4 with a designation ending in "Phase A(I)". Next, an additional iteration [Phase A(II)] was performed to improve predictive performance in a region of the experimental domain that is more likely to contain mixtures with the highest compressive strengths, resulting in fifty four additional models. For the purpose of defining the experimental domain for Phase A(II), the model described as kNN_avg_84_PhaseA(I), including w/cm as a feature, was selected to estimate the SCM contents that produce the top 20% highest compressive strength mixtures at age 56 days. Consequently, the new range of SCM contents (% replacing cement by wt.) predicted by this model consisted of (0-52%) slag, (0-10.5%) microsilica and (0-15%) fly ash. Considering this new experimental domain and to prevent overlapping with Phase A(I), 16 mixtures were designed for Phase A(II) using a L16 orthogonal array (Table A2 - APPENDIX A).

Table 4. RMSE obtained for each model when predicting the compressive strength of mixtures cast in Phase A and tested at age 56 days. These models were built considering three scenarios: 1) using only SCMs as the features; 2) adding w/cm as a feature; 3) adding w/cm and HRWR/cm as a feature

	#obs			Models with 3 variables (SCMs)					Models with 4 variables (SCMs + w/cm)					Models with 5 variables (SCMs + w/cm)				
					train set		test set	opt test set		train set		test set	opt test set		train set		test set	opt test set
Models	train set	test set	opt test set	tp	RMSE$^{(cv\|ba)}$	RMSE	RMSE	RMSE	tp	RMSE$^{(cv\|ba)}$	RMSE	RMSE	RMSE	tp	RMSE$^{(cv\|ba)}$	RMSE	RMSE	RMSE
kNN_avg_84_Phase A(I)	21	4	11	2	9.4	6.11	19.5	10.3	8	10.4	9.6	14.3	14.2	7	10.5	9.2	14.8	13
kNN_avg_100_Phase A(I)	25	-	11	4	11.5	9.55	-	10.7	8	11.3	10.8	-	11.4	10	11.4	10.8	-	12.8
kNN_avg_80_Phase A(I)+A(II)	32	8	11	4	13.3	11.2	18.1	8.4	5	13.2	11.4	17.1	9.8	3	12	8.3	15.6	10.2
kNN_avg_80_Phase A(I)+A(II)_2.0	32	8	11	3	14.5	11.14	17.8	8	11	14.2	14.1	12.7	7.6	1	12.9	≈0	10.3	11
kNN_avg_100_Phase A(I)+A(II)	40	-	11	4	13.9	12.65	-	8.6	3	12.8	9.5	-	10.3	1	11.3	≈0	-	11.3
kNN_sp_84_Phase A(I)	63	4	11	15	15.5	10.1	13.7	11.7	3	13.2	1.74	≈0	7.9	3	13.9	1.7	≈0	7.9
kNN_sp_100_Phase A(I)	75	-	11	33	13.2	12	-	13.8	27	12.6	10.9	-	12.6	24	12.7	10.6	-	11.8
kNN_sp_80_Phase A(I)+A(II)_2.0	102	8	11	12	16.6	11.4	16	7.9	21	15.8	12.2	14.8	8	24	14.8	11.6	15.1	8.3
kNN_sp_100_Phase A(I)+A(II)	120	-	11	12	15.3	12.7	-	8.6	12	14.1	10.3	-	9.9	12	12.8	9.5	-	10.3
RF_avg_84_Phase A(I)	21	4	11	2	11.3	5.7	15.5	12	1	11.2	6.3	14.5	14	1	11.3	6	14.2	14.7
RF_avg_100_Phase A(I)	25	-	11	1	12.3	7.8	-	13.4	3	11.3	5.1		13.2	2	10.7	5.5	-	16.9
RF_avg_80_Phase A(I)+A(II)	32	8	11	2	13.6	6.2	15.6	8.5	2	13.4	6	12.9	8.2	5	10	4.4	10.5	13.2
RF_avg_80_Phase A(I)+A(II)_2.0	32	8	11	2	14.8	7.3	14.3	7.9	2	13.7	6.3	13.3	7.3	4	12.4	5	7.5	8.9
RF_avg_100_Phase A(I)+A(II)	40	-	11	2	15	7.4	-	8.7	3	12.8	5.2		10.5	5	11.1	4.3	-	14.3
RF_sp_84_Phase A(I)	63	4	11	1	11.5	3.9	4.7	13	1	11	3.4	3.2	12.7	1	11	3.6	2.7	15.2
RF_sp_100_Phase A(I)	75	-	11	2	11.2	1.6	-	12.2	3	10.8	1		12.1	2	11.1	1	-	17.2
RF_sp_80_Phase A(I)+A(II)_2.0	102	8	11	2	12.1	4.5	4.3	9.4	3	11.4	3.3	1.2	10.4	5	11.2	2	1.6	12.6
RF_sp_100_Phase A(I)+A(II)	120	-	11	2	11.6	1.7	-	9.5	3	10.2	1.1		11.3	5	10.4	0.8	-	14.4
loess_avg_84_Phase A(I)	21	4	11	-	-	14.7	18.3	13.2	-	-	-	-	-	-	-	-	-	-
loess_avg_100_Phase A(I)	25	-	11	-	-	14.3	-	13.1	-	-	-	-	-	-	-	-	-	-
loess_avg_84_Phase A(I)+A(II)	32	8	11	-	-	16.3	18.3	9.82	-	-	-	-	-	-	-	-	-	-
loess_avg_100_Phase A(I)+A(II)	40	-	11	-	-	16.8	-	8.9	-	-	-	-	-	-	-	-	-	-
poly1_avg_84_Phase A(I)	21	4	11	-	-	13.2	11.5	16.5	-	-	-	-	-	-	-	-	-	-
poly1_avg_100_Phase A(I)	25	-	11	-	-	12.9	-	15.7	-	-	-	-	-	-	-	-	-	-
poly1_avg_84_Phase A(I)+A(II)	32	8	11	-	-	17	15.7	8.37	-	-	-	-	-	-	-	-	-	-
poly1_avg_100_Phase A(I)+A(II)	40	-	11	-	-	16.7	-	8.3	-	-	-	-	-	-	-	-	-	-
poly2_avg_84_Phase A(I)	21	4	11	-	-	12.4	13.1	15.6	-	-	-	-	-	-	-	-	-	-
poly2_avg_100_Phase A(I)	25	-	11	-	-	12.4	-	15.2	-	-	-	-	-	-	-	-	-	-
poly2_avg_84_Phase A(I)+A(II)	32	8	11	-	-	15.3	18.5	8.78	-	-	-	-	-	-	-	-	-	-
poly2_avg_100_Phase A(I)+A(II)	40	-	11	-	-	15.8	-	8	-	-	-	-	-	-	-	-	-	-
poly3_avg_84_Phase A(I)	21	4	11	-	-	12.2	14	16.9	-	-	-	-	-	-	-	-	-	-
poly3_avg_100_Phase A(I)	25	-	11	-	-	12.4	-	15.5	-	-	-	-	-	-	-	-	-	-
poly3_avg_84_Phase A(I)+A(II)	32	8	11	-	-	17.5	18.4	8.82	-	-	-	-	-	-	-	-	-	-
poly3_avg_100_Phase A(I)+A(II)	40	-	11	-	-	15.8	-	8.1	-	-	-	-	-	-	-	-	-	-

tp - tuning parameter ("k" for kNN; "mtry" for RF)

#obs - number of observations

opt test set - optimum test set

RMSE$^{(cv)}$ - root mean squared error of the model estimated during cross validation (kNN models)

RMSE$^{(ba)}$ - root mean squared error of the model estimated during bootstrap aggregation (RF models)

Furthermore, the first mixture of Phase A(II) was discarded considering that it involves the same design levels of the first mixture in Phase A(I). Table 4 contains detailed information on each model developed. Particularly, the RMSE obtained when fitting these models to the training, testing and optimum testing sets are listed. In addition, more accurate estimations of the RMSE in the entire training datasets were obtained for the ML models through cross-validation and bootstrap aggregation. Although a similar process can be followed to improve the estimation of the RMSE in the testing sets, this level of detail is out of scope for this work.

4.2.2. Evaluation of kNN, random forest and linear models

Average compressive strengths per mix design were used to develop the main models in this study. However, kNN cross validations were also performed while activating a GroupKFold argument that enables the model to access the individual specimen response while still fitting the data by group/experimental unit (in this case, each mix design). With this method, extra care is required when randomly splitting the data to avoid data leakage. For instance, if the results are not properly grouped by mixture design prior to splitting the data into training, validation and testing sets (and maintained that way during the cross validation process), one specimen of a given mixture can fall into the training set while the other two specimens might fall into the testing or validation sets, resulting in data leakage. While using average results typically eliminates the risk of data leakage, the aforementioned approach is used herein to compare predictive performance following both methods. On the other hand, the RF models used herein involve a bootstrap process that, as presently constructed, does not allow the data to be grouped by folds during the process. Yet, models were developed using individual specimen results with RF models exclusively to illustrate data leakage detection using the developed tools in this study.

Another pertinent consideration regarding data split is related to the orthogonal array datapoints excluded from training. As mentioned earlier, orthogonal arrays provide a strategic framework for reduced experimental runs. Essentially, each point provides important information describing that region of the experimental domain, which ultimately allow these reduced experiments to be sufficient to generate relevant ML models. With this in mind, models involving average results and a designation ending in "2.0" were generated by forcing testing sets that use only data corresponding to the additional iteration [Phase A(II)]. In this step, data from Phase A(II) was randomly split into 8 mixtures for the testing set and 7 mixtures for the training set. Thus, 32 mixtures [7 from A(II) + 25 from A(I)] were available for training, maintaining the ratio at 80% for training, similar to the ratio defined for the initial models (84% for training). As can be observed, the models obtained following this approach [for instance kNN-avg_80_Phase A(I)+A(II) 2.0] exhibited considerably lower RMSE in the testing datasets when compared to the models [for instance kNN-avg_80_Phase A(I)+A(II)] obtained by randomly splitting the data across Phases A(I) and A(II). This comparison was not performed for models using individual specimens since the data were initially split by folds through a random process. This allowed the author to pick a randomly generated fold (in this case, fold #3), for which the blind mixtures fell within Phase A(II) and were stored in the testing set, thus avoiding possible removal of important datapoints from the orthogonal array defined in Phase A(I). This iteration is shown in the APPENDIX A (Fig. A2). Further model evaluations can be done through parameters such as the variable importance, which describes how much a given feature is used by a model to accurately make predictions (Table A6 - APPENDIX A).

Typically, models that exhibit more consistency in the RMSE across training and testing sets provide more confidence in the robustness of the model and are more likely to prevent overtraining, oversmoothing and data leakage. Fig. 9 summarizes how the best performing models for each modeling technique (kNN, RF and linear models) fit the different data sets (training, testing and optimum testing) in this study. In this figure, predicted values versus actual outcomes are plotted against each other to evaluate predictive performance. In addition, the RMSE obtained across each dataset (train, test and optimum sets) are listed in each plot to facilitate comparisons between models. As illustrated, the best performing linear model (3rd degree polynomial) does not effectively model the nonlinear response in compressive strength with changes to the mixture proportions.

Not only the obtained RMSE is unbalanced between the training and both testing sets, but also very poor correlation is observed between predictions and actual outcomes. On the other hand, kNN and RF models present great balance in the obtained RMSE across training and testing sets, as well as very strong correlations between predicted versus actual outcomes. While adding w/cm as the 4th variable helped improve predictive performance, adding HRWR/cm had a negligible effect on predictive response (see results in Table 4).

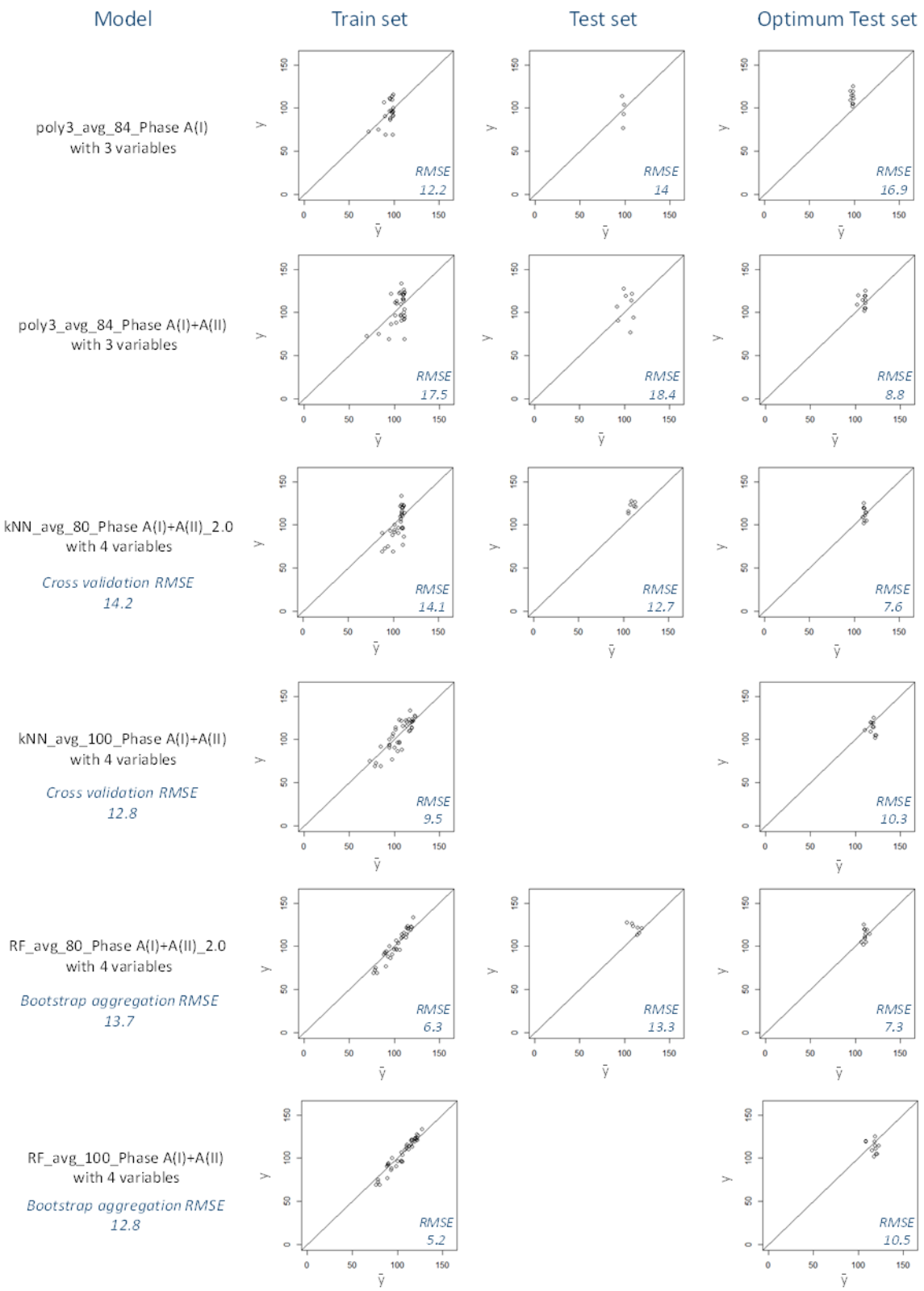


Fig. 9. Predicted compressive strengths (ȳ) versus actual outcomes (y), in MPa, for the best performing models following different modeling techniques (kNN, RF and linear models). Model nomenclature is described in section 4.2

Evaluating Fig. 9 and Table 4, the models selected to later develop the PDDs in this study were the “kNN_avg_100_PhaseA(I)+PhaseA(II)” and the “RF_avg_100_PhaseA(I)+PhaseA(II)”. Fig. A3 through Fig. *A5* (APPENDIX A) display the plots summarizing the predictive performance of the developed models for each modeling technique evaluated in this study. Further evaluation of the models is discussed in subsequent sections using 3D plots and PDDs. Based on the considerations above, Table 5 presents the optimum mixtures predicted by each model. Several of these mixtures were stored in the “optimum set” mentioned earlier.

Table 5. Optimum mixtures predicted by each model

Models	Models without w/cm % weight replacing cement			Models with w/cm % weight replacing cement		
	Slag	Microsilica	Fly Ash	Slag	Microsilica	Fly Ash
kNN_avg_84_Phase A(I)	32.0	0.6	3.6	31.0	0.0	7.0
kNN_avg_100_Phase A(I)	27.9	4.6	0.5	28.0	3.0	3.0
kNN_avg_85_Phase A(I)+A(II)	5.5	1.2	14.8	5.0	1.0	14.0
kNN_avg_85_Phase A(I)+A(II)_2.0	4.2	0.2	14.5	53.0	2.0	8.0
kNN_avg_100_Phase A(I)+A(II)	4.2	0.6	14.5	44.0	0.0	8.0
kNN_sp_84_Phase A(I)	30.9	1.2	3.6	42.4	5.3	6.4
kNN_sp_100_Phase A(I)	37.0	0.0	6.2	14.5	7.3	4.1
kNN_sp_85_Phase A(I)+A(II)_2.0	41.2	5.3	4.6	28.5	5.3	5.2
kNN_sp_100_Phase A(I)+A(II)	4.2	0.6	14.6	30.3	0.8	12.9
RF_avg_84_Phase A(I)	30.3	5.1	9.4	23.0	2.6	9.4
RF_avg_100_Phase A(I)	7.9	2.6	3.8	7.9	2.6	7.6
RF_avg_85_Phase A(I)+A(II)	37.6	1.8	13.2	37.6	2.6	13.2
RF_avg_85_Phase A(I)+A(II)_2.0	35.2	3.6	13.2	32.7	2.6	13.2
RF_avg_100_Phase A(I)+A(II)	32.7	3.6	13.2	37.6	2.6	13.2
RF_sp_84_Phase A(I)	23.0	5.1	9.4	7.9	2.6	9.4
RF_sp_100_Phase A(I)	23.0	2.6	11.4	15.2	0.0	2.0
RF_sp_85_Phase A(I)+A(I)_2.0	52.7	0.0	13.2	52.7	0.0	13.2
RF_sp_100_Phase A(I)+A(II)	37.6	2.6	13.2	32.7	1.8	13.2

4.2.3. Three-Dimensional Performance Density Plots for Model Evaluation

Fig. 10 illustrates the predictive structure of the linear and machine learning models developed in this study in a three-dimensional space, characterized by a slice in the experimental domain for which fly ash content is equal to zero. The results correspond to the 1-day compressive strengths, for which the disparity in strength between mixtures (0-93 MPa) is significantly higher than the one from the 56-day results (70-130 MPa). This allows a better comparison between the models. As shown in Fig. 10, the machine learning models do not follow a linear response across the entire domain, in contrast to the polynomial models, for which the response is characterized by a smooth surface. The flexibility provided by machine learning models allows one to predict responses that are a function of synergistic relationships, which is the case for the compressive strength of UHPC as a function of its material constituents. Considering the lack of predictive flexibility provided by linear models, these are not further evaluated in this work. Comparing the machine learning models, it is visible that the kNN models provide more flexibility in modeling the response than the RF models. This is expected, considering that the RF model is based on regression trees built with splitting nodes (yes/no conditions), leading to fixed outcomes provided at the end of each node in a given tree. However, RF models tend to predict the response magnitude with more accuracy than kNN models given that these latter models tend to oversmooth the response with increasing number of neighbors. Fig. 10 also shows how differently the RF models behave when built using individual specimen results compared to when they are built using the average results for each mixture. As can be observed, the model loses flexibility and shows a tendency for overtraining. This is not observed for the kNN models due to the GroupKFold cross validation method used in training.

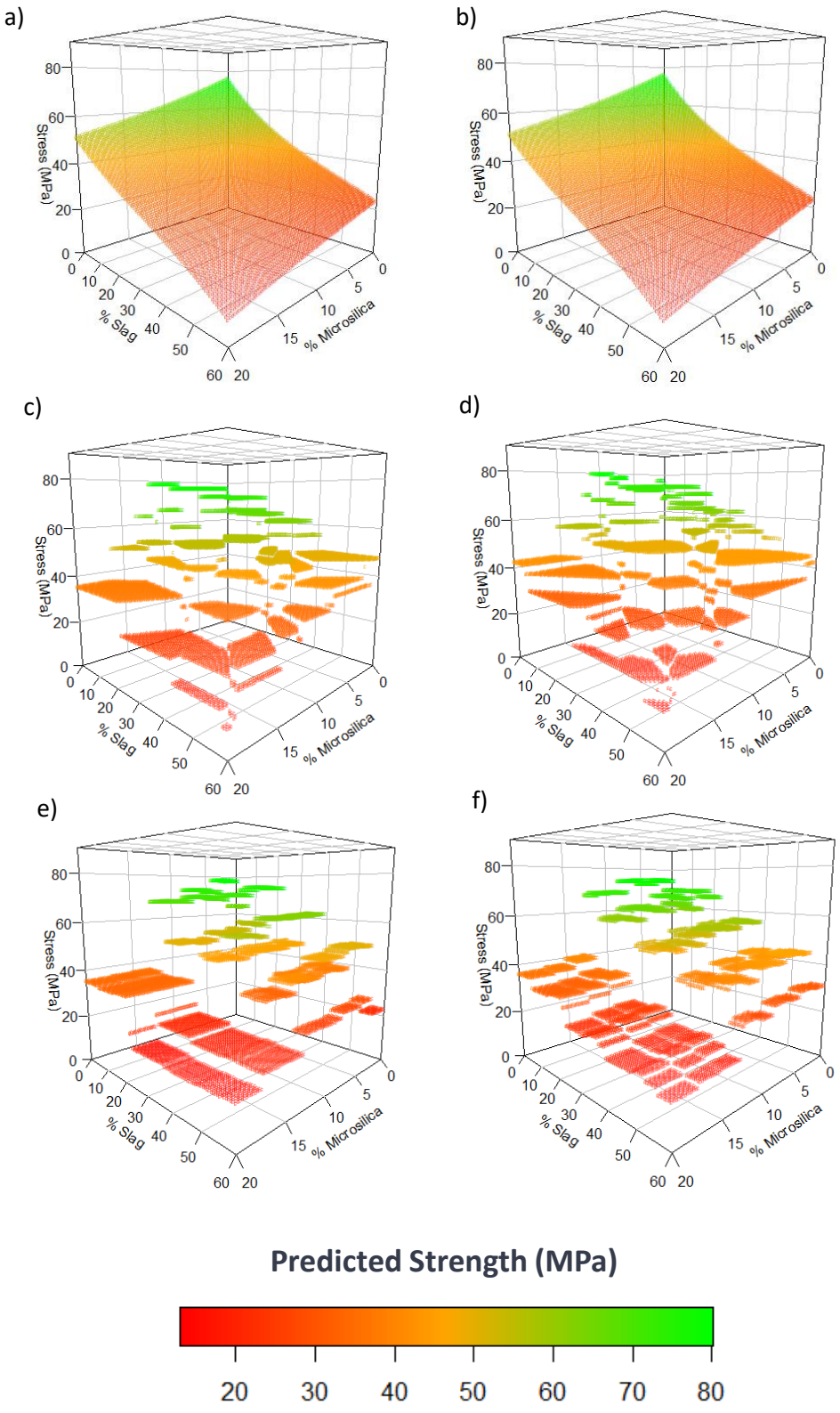


Fig. 10. 3D performance density plots generated to compare machine learning models in kNN (c,d) and RF (e,f) against polynomial models (a,b). Plots in the left (a,c,e) represent models generated with individual specimen data, while plots in the right (b,d,f) consist of models generated using average data for each mix design. The plots illustrate a slice taken in the domain, for which fly ash content = 0%

To improve predictive performance, an ensemble model was built by combining the best kNN and RF models, which is discussed in the next section. Fig. 11 illustrates the effect of using the ensemble method on the final model response. Once again, the 1-day compressive strength results were used, allowing a higher disparity between different mixtures that enables a better visualization of the ensemble effect.

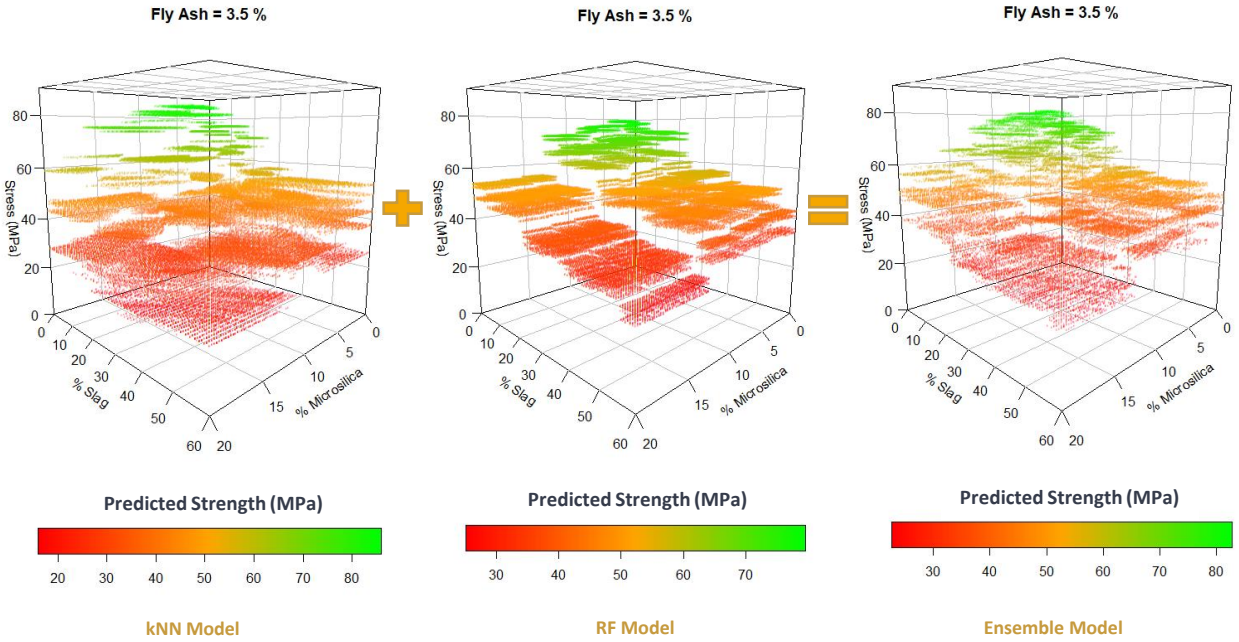


Fig. 11. 3D performance density plots generated to evaluate the effect of the ensemble model

4.3. Classification Models

4.3.1. Categorical Performance Density Diagrams for Failure Predictions

The fracture type observed during compression tests was recorded following ASTM C39/C39M (ASTM 2021) and used in this work to illustrate the development of categorical PDDs. Considering that there was no variation observed for the fracture type of specimens tested at age 56 days (Fig. A1 - APPENDIX A), the data from specimens tested at age 1 day was used to develop this model. In addition, Fig. A1 (APPENDIX A) shows a strong correlation between the fracture type

and the compressive strength. Thus, in addition to the SCM variables, the compressive strength was also used to predict the fracture type. First, different kNN models were evaluated in predicting the compressive strength at age 1 day, similar to the process described in the previous section. The best performing model was used to predict the compressive strength and generate a vector that was later added to the expanded predictors matrix. Next, a kNN model was built using slag, microsilica, fly ash and compressive strength as features, while training the model on 85% of the Phase A data. Fig. 12 illustrates the k-fold cross validation process used to optimize the tuning parameter "k" for this classification model.

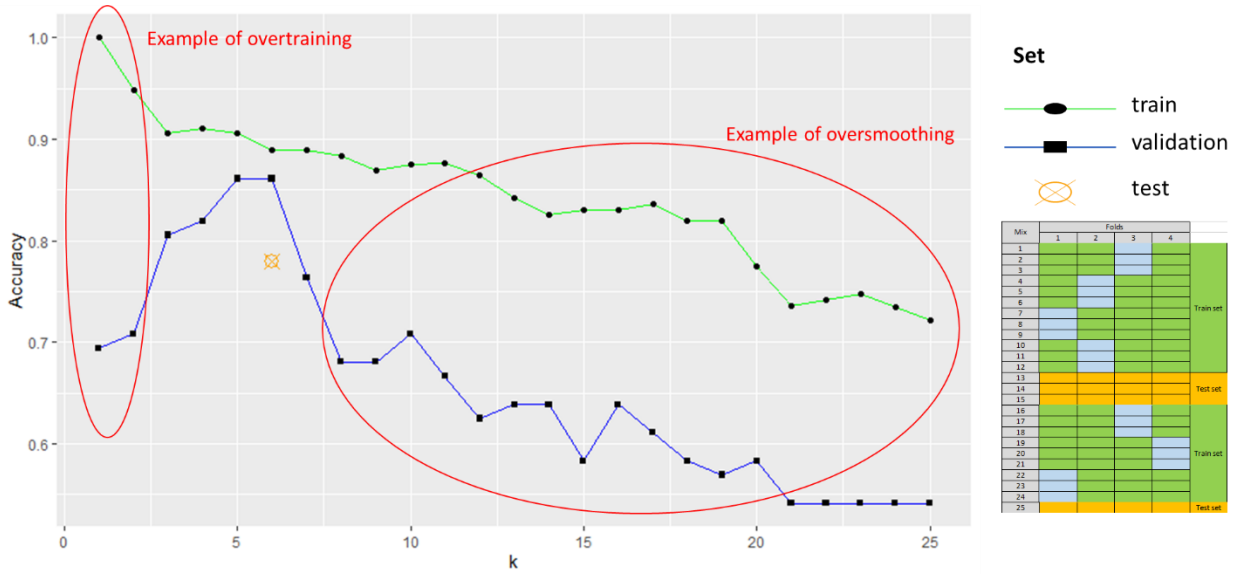


Fig. 12. Tuning parameter optimization performed for a kNN model using k-fold cross validation. The maximum average accuracy in the validation sets was obtained with k=6. After defining the kNN model with k=6, k-fold cross validation was performed in a similar manner, creating different test set folds to estimate the overall model accuracy.

For categorical outcomes, the goal consists of maximizing the accuracy in the validation set to define the optimum "k" value. After defining this "k" value, a new cross-validation process can be used to get a better estimate of the accuracy of the model by testing it in different testing sets. Note that other metrics such as a balanced accuracy or F-1 score are often preferred to

evaluate the performance of classification models. However, such discussion is out of the scope of this study.

4.4. Summary of Important Outcomes

This chapter was motivated by several challenges related to the use of AI models in predicting, describing and displaying concrete material's performance, especially with small experimental datasets. The objective consisted of developing a new tool that permits optimizing the compressive strength of UHPC while maximizing the fa/cm. The main findings from this part of the study are:

- After evaluating predictive performance of the models evaluated, results suggest that the experimental framework used in this study, supported by orthogonal arrays and the use of surrogate samples has shown to be an effective method for rapidly generating data to support AI algorithms for concrete mixture design optimization. This method can be used to overcome the uncertainty of models generated with large, multiple-source datasets by enabling reduced experimental runs capable of effectively describing the experimental domain.
- Results suggest that predictive performance of models improved with increased number of observations in the dataset. RMSEs indicate that the two-step iteration process contributed to increased performance in the region of interest (high compressive strengths) for blind data (optimum test sets).

The machine learning algorithms used in this study were dictated by the author's proficiency in these particular techniques, whereas the number of experimental runs were dictated by the time

constraints associated with the graduate program timeline. Further studies should be conducted using different machine learning techniques that, perhaps, are even more suitable for small datasets. Furthermore, increased design levels of orthogonal arrays (e.g. 7^2) can be used to increase the dataset and improve predictive performance in the entire domain.

4.5. References

ASTM. 2021. "ASTM C39 / C39M-21, Standard Test Method for Compressive Strength of Cylindrical Concrete Specimens." In. West Conshohocken, PA: ASTM International.

IBM. 2020. 'Artificial Intelligence', IBM Cloud Education. https://www.ibm.com/cloud/learn/what-is-artificial-intelligence.

Irizarry, R.A. 2019. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R* (Chapman and Hall/CRC).

Kuhn, Max. 2008. 'Building Predictive Models in R Using the caret Package', *Journal of Statistical Software*, 28: 1 - 26.

Larson, S. C. 1931. 'The shrinkage of the coefficient of multiple correlation', *Journal of Educational Psychology*, 22: 45-55.

Mosteller, F., and J Tukey. 1968. 'Data Analysis, including Statistics.' in G. Lindzey and E. Aronson (eds.), *Revised Handbook of Social Psychology* (Addison Wesley).

Mosteller, Frederick, and David L. Wallace. 1963. 'Inference in an Authorship Problem', *Journal of the American Statistical Association*, 58: 275-309.

Stone, M. 1974. 'Cross-Validatory Choice and Assessment of Statistical Predictions', *Journal of the Royal Statistical Society. Series B (Methodological)*, 36: 111-47.

CHAPTER V.

PERFORMANCE DENSITY DIAGRAMS

5.1. Concept and Application

A matrix containing the ranges of slag, microsilica and fly ash contents defined earlier was generated in increments of 0.5% by wt. replacing cement. A function that takes this matrix and produces all possible combinations for the existing contents was used to expand this matrix, producing a new matrix of predictors/features. Each model developed as described in the previous section was used to predict the compressive strength of each row of the predictor's matrix, which essentially represents a mixture design. For the models developed with w/cm and HRWR/cm as features, additional columns were generated for the predictor's matrix with fixed values of 0.20 and 0.015, respectively.

For each model, the vector containing the predicted outcomes was combined with the matrix of features to develop a new mix design tool described as PDD. PDD is a diagram that resembles a matrix of contour plots, with the outcome displayed as the z-coordinate with varying density levels, built using orthogonal arrays to reduce experimental runs and machine learning algorithms to efficiently model responses that are not characterized by smooth surfaces in space. The efficiency of this tool is a function of the data provided within the experimental domain and shall be generated following proper strategies for reduced experimental runs such as orthogonal arrays or any other suitable factorial design method. The synergistic effect of multiple features on a given outcome can be evaluated concurrently using PDDs with multiple faceting strategies. Fig. 13 illustrates the reasoning for the variable display defined for this study, which consists of

placing the most important variables (Table A6 – APPENDIX A) in the x and y axes, while faceting the least important variables by blocks. The magnitude of the outcome (z axis) is shown in the diagram as the density factor.

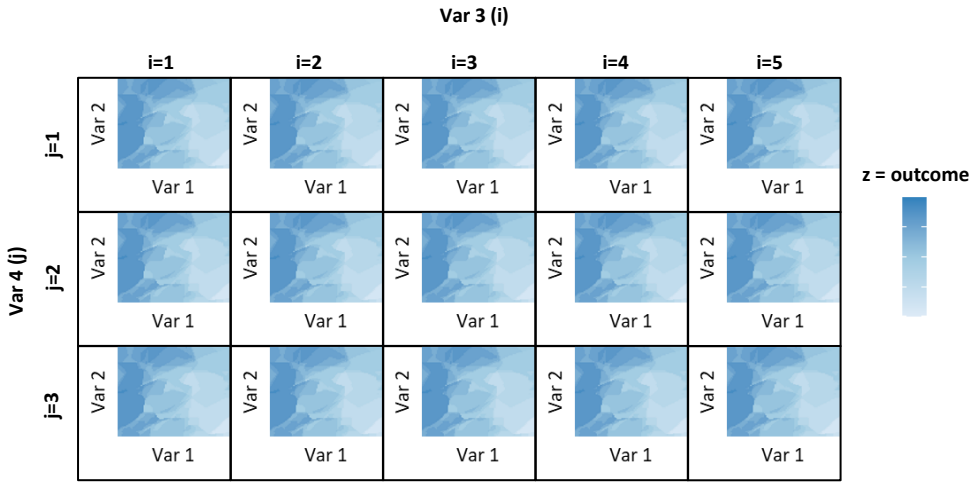


Fig. 13. Example of a faceting strategy to evaluate 4 variables (features/predictors) concurrently using PDDs

5.2. Predicting Compressive Strength with Performance Density Diagrams

The models created in section 4.2 were developed to predict the compressive strength of every possible mixture within the range of the experimental domain. As mentioned in that section, models that exhibit more consistency in the RMSE across training, testing and optimum sets typically provide more confidence in the robustness of the model and are more likely to prevent overtraining, oversmoothing and data leakage. However, PDDs developed in this study indicate that performance evaluation of the models should not rely solely on balanced RMSE results. Fig. 14 illustrates an example where this assumption can lead to poorly developed models. The PDD illustrated corresponds to the model RF_sp_100_Phase_A(I). Judging by the results in Table 4, this model does not give any indication of overtraining or data leakage occurrences considering the balanced RMSE obtained across training and testing (optimum) sets. Yet, the predictive structure of the model displayed by the PDD in Fig. 14 shows inconsistency in the trends

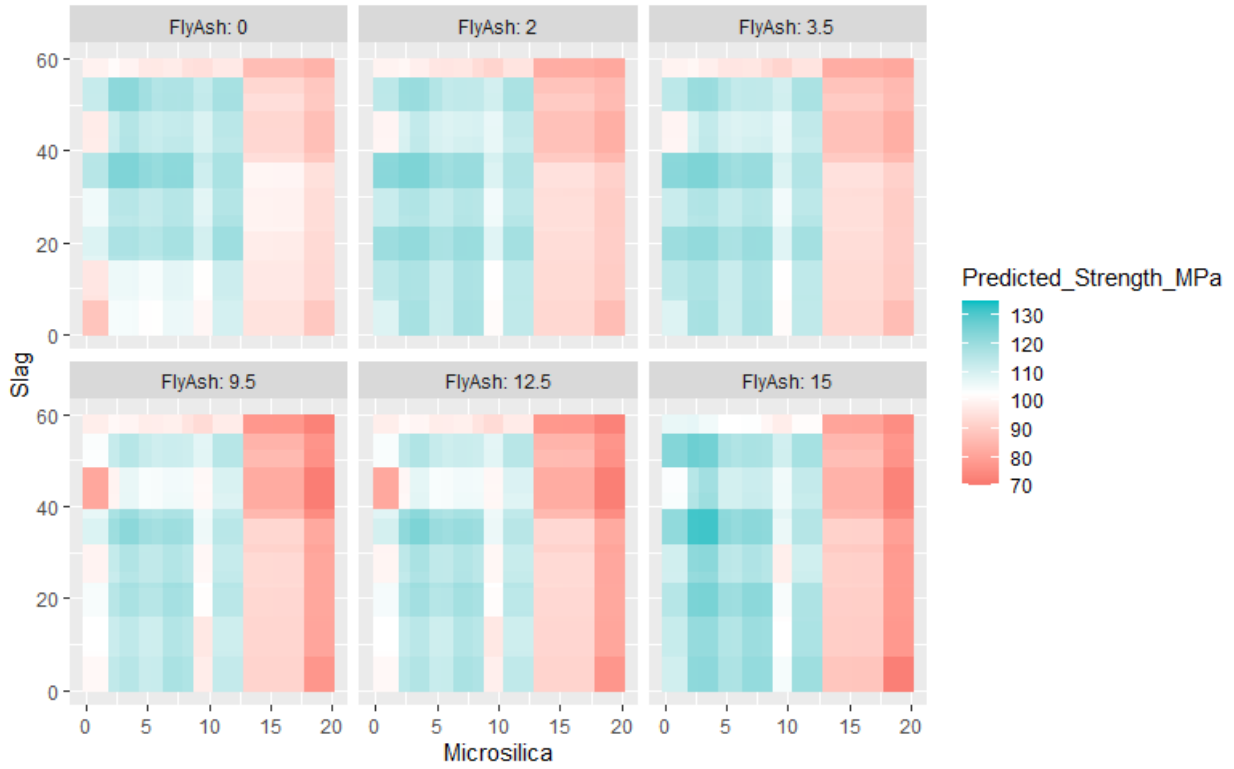


Fig. 14. PDD generated with the *"RF_sp_100_Phase_A(I)"* model (with w/cm as a variable) to predict the 56-day compressive strength of UHPC binders during Phase A. Inconsistency in observed in the trends suggest occurrence of data leakage

associated with changes in mix proportions and mechanical response. For instance, for mixtures with fixed contents of fly ash and microsilica (e.g., 0%) the changes in predicted strength are not consistent with the changes in slag content, which is shown by sequential increases and decreases in strength. The consistent RMSEs obtained across training and testing sets are, most probably, a consequence of the data leakage phenomena, where the same information is present in both training and testing sets. After careful evaluation of the models in Table 4, based on the obtained RMSE, and the tendencies also observed for each model in Fig. 10, the models selected to build the PDDs for compressive strength prediction at age 56-days are the "kNN_avg_100_PhaseA(I)+PhaseA(II)" and the "RF_avg_100_PhaseA(I)+PhaseA(II)", both using w/cm as a feature. Fig. 15 and Fig. 16 show the PDDs generated for the kNN and RF models, respectively.

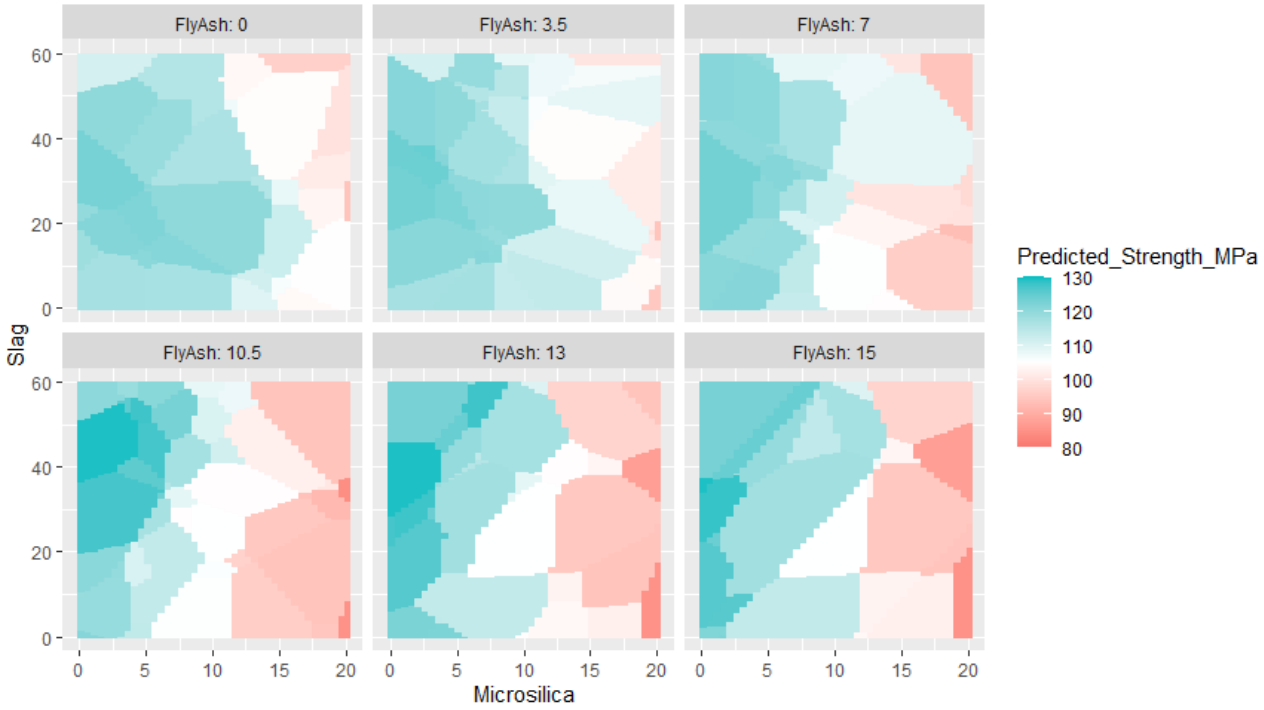


Fig. 15. PDD generated with the kNN model to predict the 56-day compressive strength of UHPC binders during Phase A

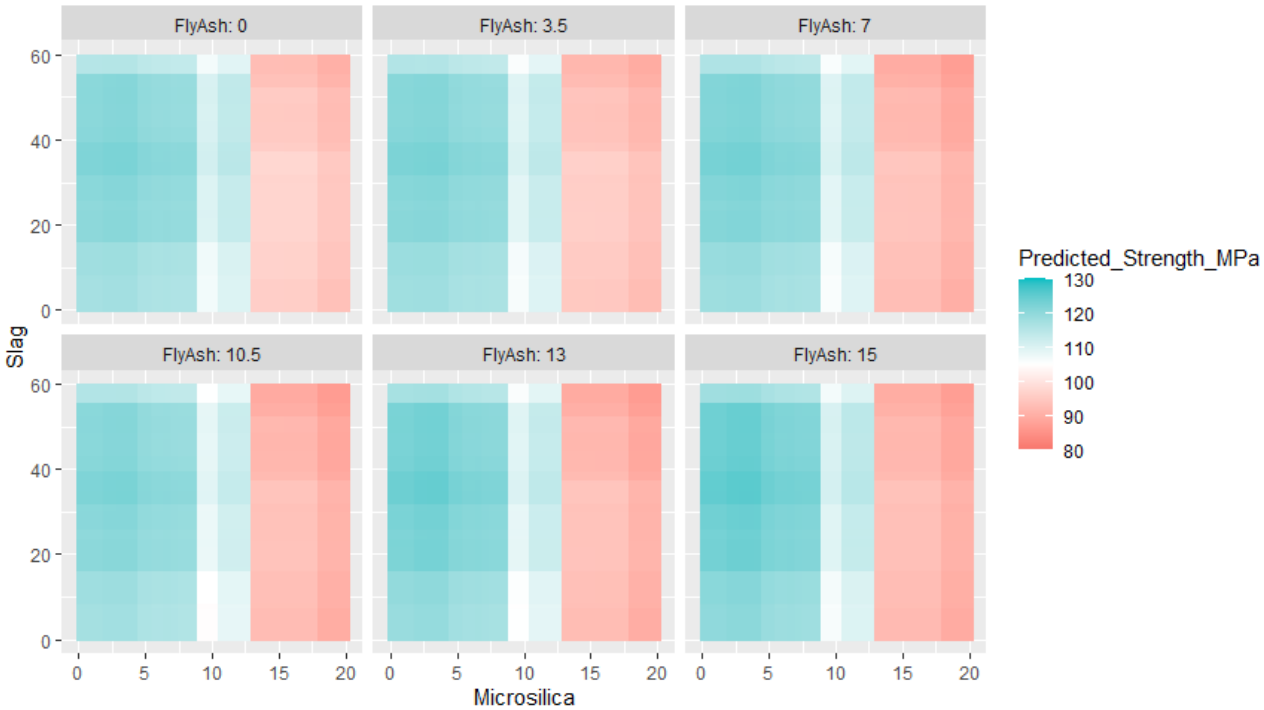
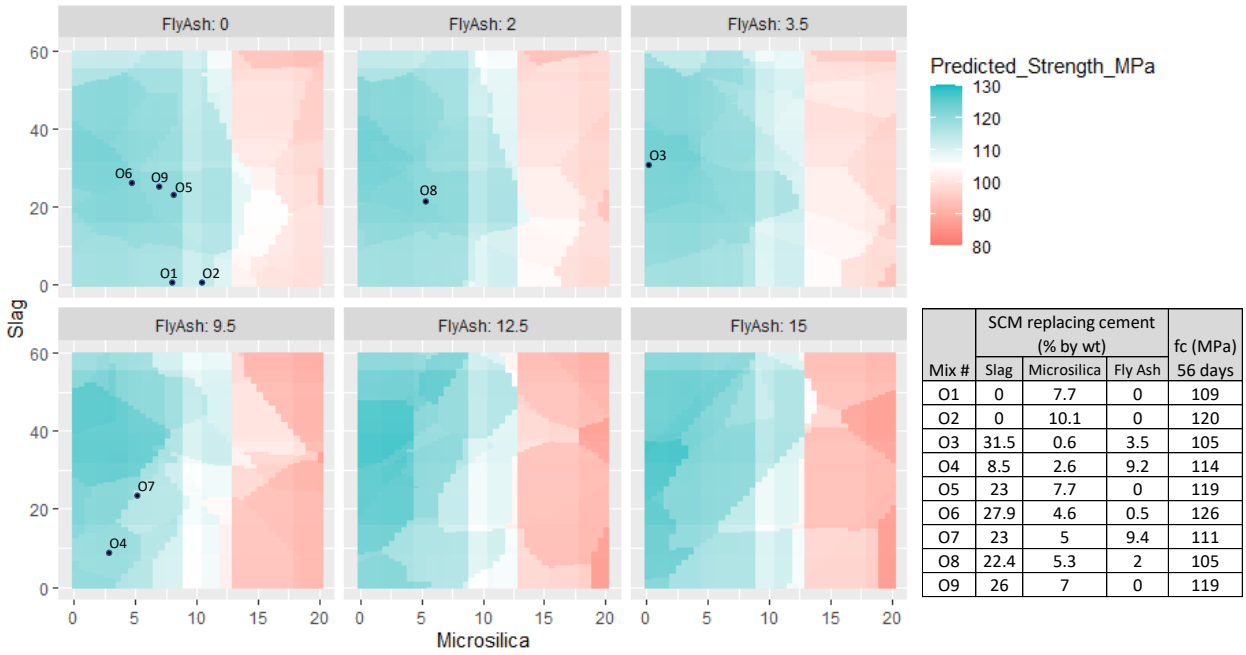


Fig. 16. PDD generated with the RF model to predict the 56-day compressive strength of UHPC binders during Phase A

As mentioned earlier, in order to benefit from the best qualities from each model and enhance the overall predictive performance, an ensemble model was built by averaging the predictions from the kNN and RF models mentioned above. The PDD generated with the ensemble model is shown in Fig. 17.



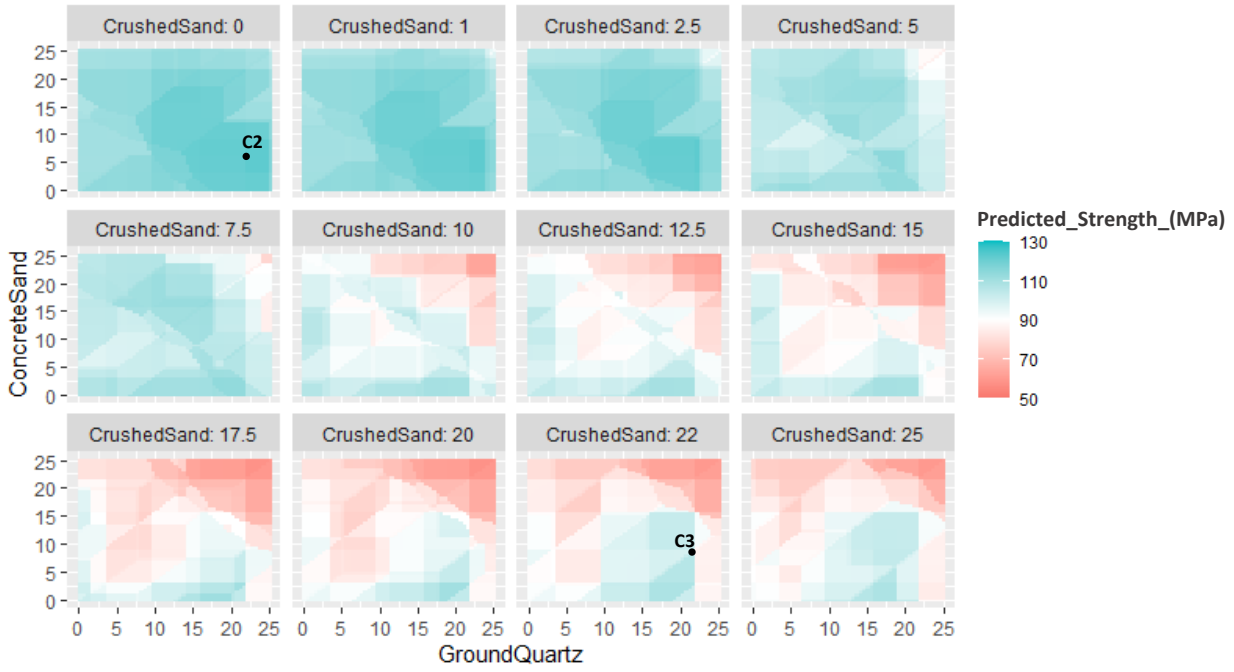
Mix #	SCM replacing cement (% by wt)			fc (MPa)
	Slag	Microsilica	Fly Ash	56 days
O1	0	7.7	0	109
O2	0	10.1	0	120
O3	31.5	0.6	3.5	105
O4	8.5	2.6	9.2	114
O5	23	7.7	0	119
O6	27.9	4.6	0.5	126
O7	23	5	9.4	111
O8	22.4	5.3	2	105
O9	26	7	0	119

Fig. 17. PDD generated with the Ensemble model to predict the 56-day compressive strength of UHPC binders from Phase A. Optimum mixtures denoted with "O" were tested to evaluate the model efficiency

Each point represents a mixture design. SCM contents (as % by wt. replacing cement) are indicated as coordinates, whereas the corresponding compressive strength is shown as the density factor. Several mixtures described as optimum by each model (Table 5) were experimentally tested and stored in the optimum set. As shown in Fig. 17, most of the experimental results from the optimum set are effectively predicted by the Ensemble PDD, both in magnitude and in rank [O6>(O5=O9)>O4>O7]. The exceptions are mixtures O3, O8 and O2. This could be explained by experimental outliers or with natural errors expected from a model created with such a reduced dataset. Most models were developed after initiating Phase B. A

preliminary model was used to estimate the best binder from Phase A, which was then selected as the base binder (constant SCM/cm) to evaluate the inclusion of aggregates during Phase B. This model suggested mixture O8 as the optimum, which in reality does not represent the binder with the highest compressive strength in Phase A, as it can be observed in Fig. 17.

Phase B was designed following a L25 orthogonal array, for which the variables consist of ground quartz, concrete sand and a crushed sand. After developing an ensemble model, following the methodology described for Phase A, a PDD for this model was created and used to identify the final optimum mixtures for the optimization process. Fig. 18 illustrates the ensemble PDD created for Phase B, as well as the experimental results for three particular mixtures of interest.



Mixture #	SCM replacing cement (% by wt)			Aggregates replacing cementitious (% by wt)			56 day results	
							New test method	ASTM C1856
	Slag	Microsilica	Fly Ash	Ground Quartz	Concrete Sand	Crushed Sand	fc (MPa)	fc (MPa)
C1	0	10.1	0	22	6.5	0	128.4	133.8
C2	22.4	5.25	1.97	22	6.5	0	129.2	158.1
C3	22.4	5.25	1.97	21.5	8.5	22	128.5	154.8

Fig. 18. PDD generated with an Ensemble model to predict the 56-days compressive strength of UHPC concrete during Phase B

These mixtures were designated with a "C" to differentiate them from mixtures produced in Phase B using the orthogonal array. Mixture C1 consists of a mortar containing the same binder proportioning of mixture O2, which entails a very common silica fume content in UHPC materials (10% replacing cement) according to the literature (Ibrahim et al. 2017; Alsalman, Dang, and Micah Hale 2017). This mixture is not shown in the PDD for Phase B considering that this PDD was generated using a different base binder (Mixture O8). On the other hand, mixture C2 contains the binder of mixture O8, which was initially presumed as the ideal binder from Phase A. The optimum aggregate content to maximize the compressive strength for both C1 and C2 was chosen by observing the PDD in Fig. 18. To evaluate the possibility of designing a much more sustainable mixture, the PDD in Fig. 18 was used to identify the mixture C3, for which the aggregate content is higher than the cementitious content, while still predicting compressive strengths over 120 MPa. All three mixtures were further tested following the standard test method described in ASTM C1856/1856M, which consists of using 76 x 152 mm cylinders with the top ends ground by a fixed-end grinder[2]. It is noticeable that not only the strength rank was maintained, but more importantly, three mixtures with outstanding performance levels were designed by quickly observing the PDDs. In particular, mixture C3 averaged a compressive strength of 155 MPa using a fine-aggregate-to-cementitious ratio of 1.04. This fine aggregate content is significantly higher than values traditionally reported for UHPC materials (0.5-0.9) (Carey et al. 2020). In addition, several other factors must be considered when comparing the

[2] Despite the fact that the new compressive strength test protocol developed herein showed great agreement with the standard method for the preliminary phase (Fig. 7), it appears that the end capping condition must be further improved for strengths over 125 MPa, considering the discrepancies obtained between this test protocol and the standard method.

obtained mixture design with other proprietary and non-proprietary mixtures published in the literature: 1) special curing conditions such as heat curing or post-thermal treatment (von Werder et al. 2021) were not applied; 2) nanomaterials such as carbon nanofibers, which are often used in new generation UHPC improving packing in the nanostructure (MacLeod et al. 2020; Wu et al. 2018; Meng and Khayat 2018; Muhd Norhasri et al. 2016; Yu, Spiesz, and Brouwers 2014), were not used; 3) high strength fibers, which are known to typically improve the compressive strength in UHPC matrices (Pyo, Kim, and Lee 2017; Ibrahim et al. 2017; Yoo, Lee, and Yoon 2013), were not used; 4) a dense gradation was not pursued through sieving the aggregates; 5) air detrainers, often used to reduce the pores in the matrix (ACI-Committee-212 ; Pham, Toumi, and Turatsinze 2018a; Pham, Toumi, and Turatsinze 2018b), were not used; and 6) the binder used as the base for Phase B was not the optimum measured experimentally. The author hypothesize that developing new PDDs with ingredients and conditions that overcome the shortcomings mentioned above can lead to innovative and sustainable UHPC mix designs, with compressive strengths over 200 MPa while keeping the aggregate-to-cementitious ratio over unity.

5.3. Predicting Fracture Type with Categorical Performance Density Diagrams

PDDs are also useful to predict categorical outcomes. As mentioned earlier, the fracture type/pattern observed during compression tests is used here to illustrate the development of these type of diagrams. Description of different fracture patterns can be found in ASTM C39 (ASTM 2021). Fig. 19 illustrates the fracture types observed experimentally against the predictions provided by the model through a categorical PDD. It is important to note that, for each mixture design, the fracture type displayed corresponds to the one with higher frequency of occurrence between the three tested specimens. For a more conservative estimation, these

PDDs can be built based on the probability of occurrence of a given fracture type. This type of PDD is illustrated in Fig. 20.

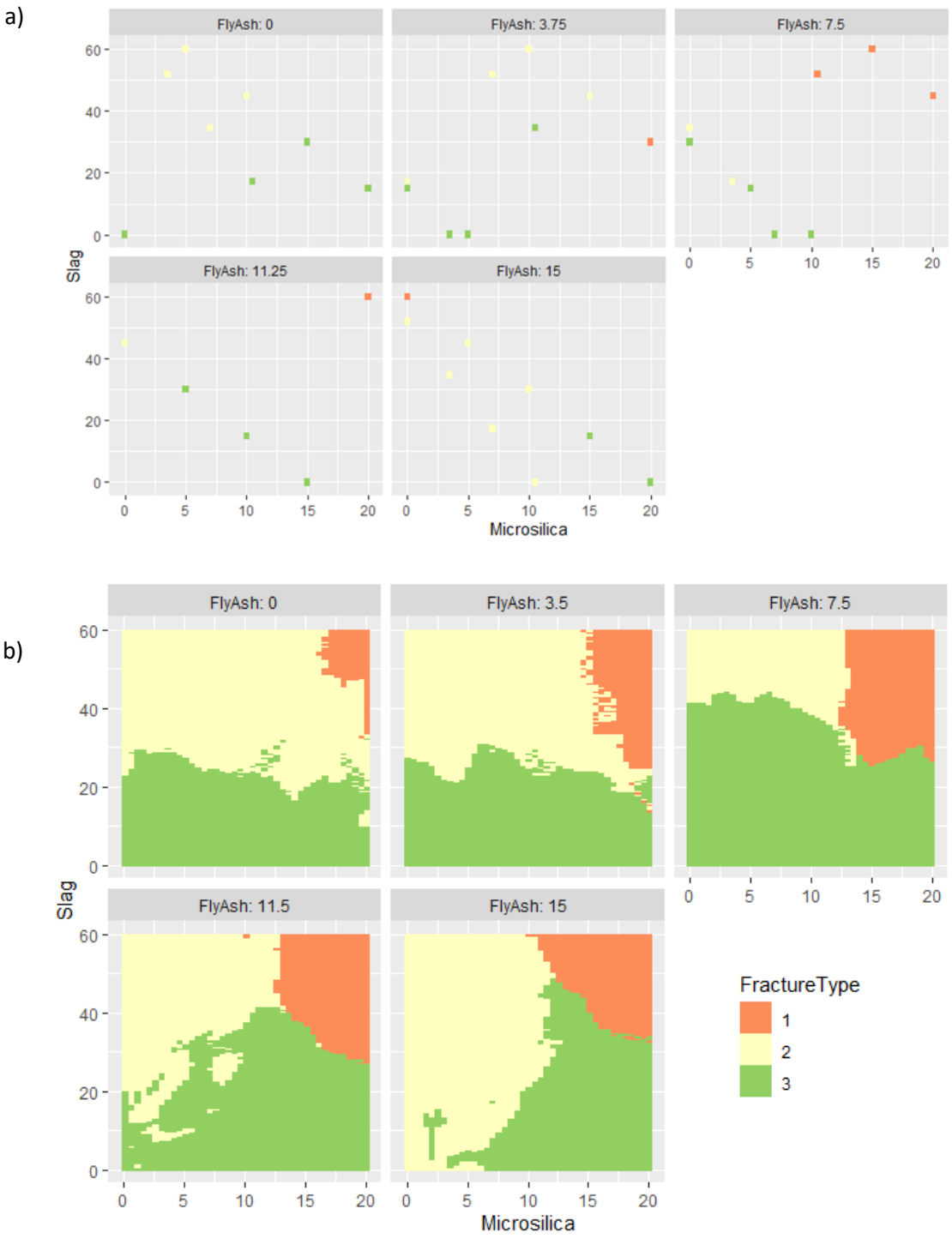


Fig. 19. Comparison between experimental results and predictions from categorical PDDs: a) most frequent fracture type observed for each one of the 25 mixtures tested experimentally in Phase A(I). Each datapoint defines a mix design with the % by wt. of SCM content replacing cement, while the color describes the fracture type; b) Categorical PDD generated to predict fracture type

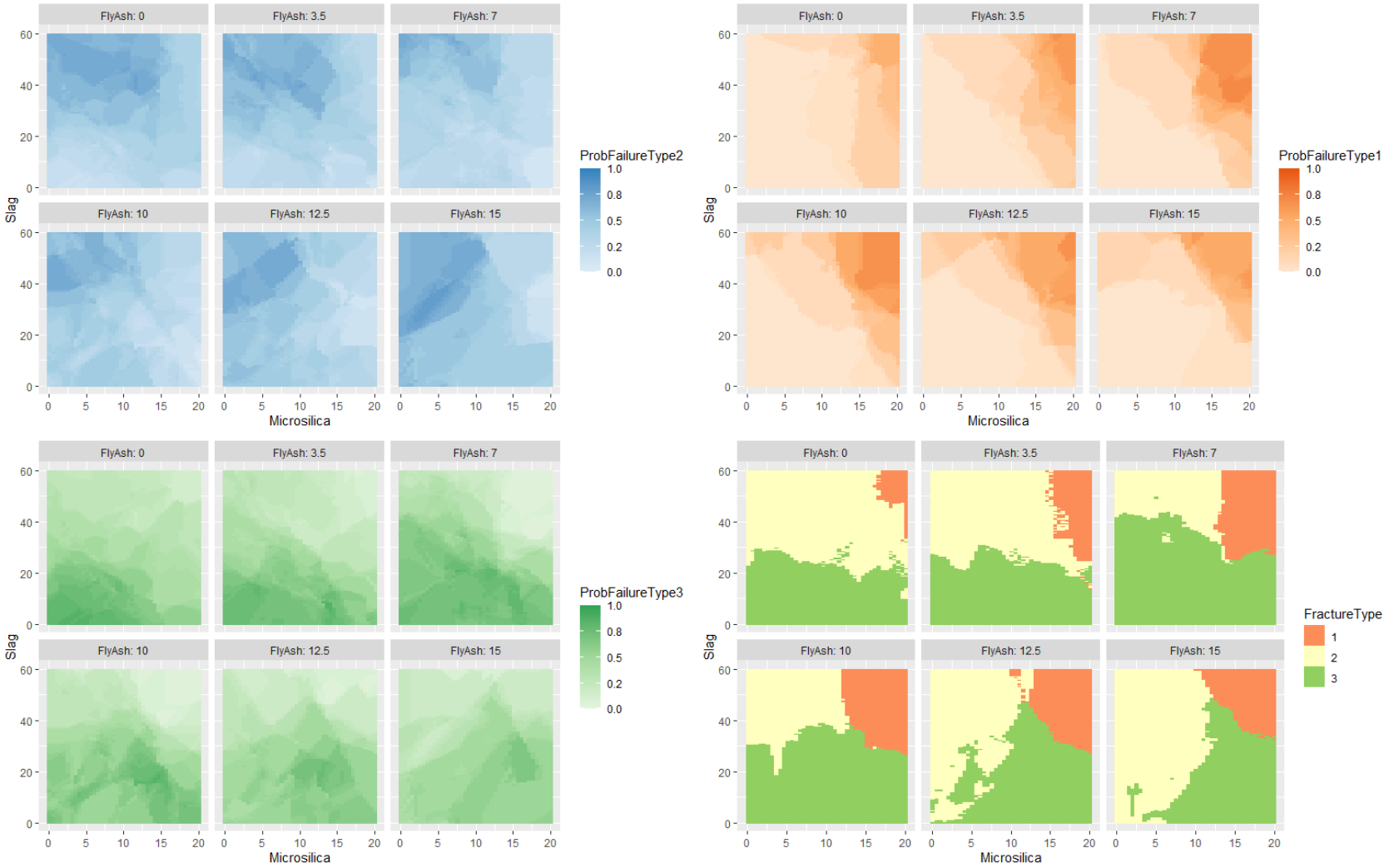


Fig. 20. Categorical PDDs generated with a kNN model to predict the fracture type of specimens under uniaxial load as a function of different cement replacement levels of fly ash, slag, and microsilica.

As it can be observed, PDDs also provide new opportunities for several areas in civil engineering other than mixture proportioning optimization. For instance, a similar approach can be used to develop models that predict distress type in pavements under a given set of conditions or to develop models that predict the failure mode for structural elements given other design factors influencing performance that are not necessarily material proportioning.

All codes used to develop PDDs associated with Phase A are shown in APPENDIX D. Code for Phase B followed the same methodology, with changes in the input variables and objective functions.

5.4. Summary of Important Outcomes

This part of the study was motivated by the increasing need of easily interpretable tools and methods to be used by industry personnel on their day-to-day operations that allow quick assessments of the effect of changes in mixture proportions on mechanical performance. The objective consisted of developing a new tool that permits optimizing the compressive strength of UHPC while maximizing the fa/cm. The main findings from this part of the study are:

- In regression models, consistency in the RMSE across training and testing sets typically provide more confidence in the robustness of the model and is a good indicator of potential overtraining, oversmoothing and data leakage occurrences. However, PDDs developed in this study suggest that model predictive performance evaluation should not rely solely on RMSE. A potential data leakage occurrence was detected using PDDs, where inconsistent trends in the domain where observed. The consistent RMSE obtained across training and testing sets are most probably a consequence of the data leakage phenomena, where the same information was present in the training and testing set.

- A PDD developed with an ensemble model enabled the design of an UHPC mixture averaging a compressive strength of 155 MPa (22,480 psi) at age 56 days, while maintaining the fine-aggregate-to-cementitious ratio above unity (fa/cm = 1.04).

The approach and methodology used in this study represents a great opportunity for innovative and impactful design protocols, capable of offering enough flexibility for exploring new constituents and account for differences in source, availability and cost of material per region. This is particularly important for advanced concrete materials such as UHPC, for which mixture optimization is key to justify its application, considering its high cost per unit volume along with its volumetric environmental impact. PDDs can further be used to evaluate mechanical and durability properties directly affecting cost and environmental impact and, combined with 3D density plots, they not only serve as decision-making aid tools to be used during mix design stages, but also help overcome the challenges mentioned throughout this manuscript. In particular, these tools provided further insight on the inner workings of machine learning models by displaying their predictive structure, helping identify overfitting and data leakage occurrences where the RMSE was not sufficiently informative. The approach used in this chapter will be used to further develop new diagrams that allow designing mixtures that are more eco-friendly and cost-effective. Multi-objective diagrams will also be developed to evaluate mechanical performance, cost and environmental impacts concurrently.

5.5. References

ACI-Committee-212. 1963. "Admixtures for Concrete." In *ACI Journal Proceedings*.

Alsalman, Ali, Canh N. Dang, and W. Micah Hale. 2017. 'Development of ultra-high performance concrete with locally available materials', *Construction and Building Materials*, 133: 135-45.

ASTM. 2021. "ASTM C39 / C39M-21, Standard Test Method for Compressive Strength of Cylindrical Concrete Specimens." In. West Conshohocken, PA: ASTM International.

Carey, A. S., I. L. Howard, D. A. Scott, R. D. Moser, J. Shannon, and A. Knizley. 2020. 'Impact of Materials, Proportioning, and Curing on Ultra-High-Performance Concrete Properties', *ACI Materials Journal*, 117: 213-22.

Ibrahim, Mustapha A., Maen Farhat, Mohsen A. Issa, and Jessica Amanda Hasse. 2017. 'Effect of Material Constituents on Mechanical and Fracture Mechanics Properties of Ultra-High-Performance Concrete', *ACI Materials Journal*, 114: 453-65.

MacLeod, Alastair J. N., Andras Fehervari, Will P. Gates, Estela O. Garcez, Laurie P. Aldridge, and Frank Collins. 2020. 'Enhancing fresh properties and strength of concrete with a pre-dispersed carbon nanotube liquid admixture', *Construction and Building Materials*, 247: 118524.

Meng, Weina, and Kamal H. Khayat. 2018. 'Effect of graphite nanoplatelets and carbon nanofibers on rheology, hydration, shrinkage, mechanical properties, and microstructure of UHPC', *Cement and Concrete Research*, 105: 64-71.

Muhd Norhasri, M. S., M. S. Hamidah, A. Mohd Fadzil, and O. Megawati. 2016. 'Inclusion of nano metakaolin as additive in ultra high performance concrete (UHPC)', *Construction and Building Materials*, 127: 167-75.

Pham, N. P., A. Toumi, and A. Turatsinze. 2018a. 'Rubber aggregate-cement matrix bond enhancement: Microstructural analysis, effect on transfer properties and on mechanical behaviours of the composite', *Cement and Concrete Composites*, 94: 1-12.

Pham, Ngoc Phuong, Ahmed Toumi, and Anaclet Turatsinze. 2018b. "Effect of Styrene-Butadiene Copolymer Coating on Properties of Rubberized Cement-Based Composites." In *Strain-Hardening Cement-Based Composites*, edited by Viktor Mechtcherine, Volker Slowik and Petr Kabele, 342-50. Dordrecht: Springer Netherlands.

Pyo, Sukhoon, Hyeong-Ki Kim, and Bang Yeon Lee. 2017. 'Effects of coarser fine aggregate on tensile properties of ultra high performance concrete', *Cement and Concrete Composites*, 84: 28-35.

von Werder, Julia, Sebastian Simon, André Gardei, Patrick Fontana, and Birgit Meng. 2021. 'Thermal and hydrothermal treatment of UHPC: influence of the process parameters on the phase composition of ultra-high performance concrete', *Materials and Structures*, 54: 44.

Wu, Zemei, Caijun Shi, Kamal Henri Khayat, and Linbin Xie. 2018. 'Effect of SCM and nano-particles on static and dynamic mechanical properties of UHPC', *Construction and Building Materials*, 182: 118-25.

Yoo, Doo-Yeol, Joo-Ha Lee, and Young-Soo Yoon. 2013. 'Effect of fiber content on mechanical and fracture properties of ultra high performance fiber reinforced cementitious composites', *Composite Structures*, 106: 742-53.

Yu, R., P. Spiesz, and H. J. H. Brouwers. 2014. 'Effect of nano-silica on the hydration and microstructure development of Ultra-High Performance Concrete (UHPC) with a low binder amount', *Construction and Building Materials*, 65: 140-50.

CHAPTER VI.

COST AND ECO-EFFICIENCY DENSITY DIAGRAMS

In this chapter, the domain of mix proportions defined in section 3.1 and their respective 56^{th}-day compressive strengths (predicted in section 5.2) will be used to estimate the changes in strength associated with changes in mix proportioning. Unit cost and GWP factors, displayed in Table 6, will be combined with these predicted strengths to calculate the comparison indices, described in section 6.1, for the proposed domain of mix proportions. New machine-learning-based density diagrams will be developed following the methodology described in section 5.1 to evaluate the effect of mix proportioning on cost and eco-efficiency indices.

6.1. Comparison Indices: Theory, Methodology and Proposed Modifications

6.1.1. Comparison Indices for Reinforced Concrete Members (Kourehpaz and Miller 2019)

In recent years, Miller et al. (Miller et al. 2016) proposed metrics that account for the role of material properties on the volume of material required for a given structural member and the associated volumetric environmental impact for the given mixture. These comparison indices were based on simplified cases, where only one material property was defined as the controlling factor for performance. Initially, these indices were developed for non-reinforced concrete applications. Later on, Kourehpaz and Miller (Kourehpaz and Miller 2019) proposed modified indices to account for steel reinforcement in short columns and beams under flexure.

To show how mechanical performance and volumetric environmental impact influence the design of structural members, a comparison index was derived in (Kourehpaz and Miller 2019) for a simply supported reinforced beam subjected to a uniformly distributed load. For this case,

width, load and length were assumed as specified by design, while the depth was modeled as a free variable allowed to vary based on the required moment of inertia as determined by the strength of the concrete material. For this scenario and assuming a rectangular cross-section, the moment resisted by the concrete (M_c) in the outer fiber can be defined as

$$M_c = \frac{f_r b h^2}{6}, \quad \{1\}$$

where h and b are, respectively, the depth and width of the member, and f_r is the modulus of rupture. For a simply supported beam subjected to a uniformly distributed load, the maximum moment in the outer fiber is

$$M_{\max} = \frac{w l^2}{8}, \quad \{2\}$$

in which w is the distributed load and l is the length of the member. If we relate equations {1} and {2} and solve for h we find

$$h = 0.866 \frac{w^{0.5} l}{b^{0.5} f_r{}^{0.5}}. \quad \{3\}$$

Meanwhile, the total environmental impact $\tilde{I}$ associated with a given member can be written as

$$\tilde{I} = l(bh - A_s) i_c + l A_s i_s, \quad \{4\}$$

where A_s is the cross-sectional area of steel, i_c is the volumetric environmental impact of concrete and i_s is the volumetric environmental impact of steel. To incorporate the compressive

strength f_c of different concretes into this assessment, the modulus of rupture in {3} was estimated as

$$f_r = 0.62\sqrt{f_c(MPa)} \quad \{5\}$$

following the relationship defined by the American Concrete Institute 318-11 building code (ACI 2011) for normal strength concretes (NSC). Substituting equations {3} and {5} into equation {4}, the total environmental impact $\tilde{I}$ of the reinforced concrete member can be defined as

$$\tilde{I} = 1.1\ l^2(wb)^{0.5}\left(\frac{i_c}{f_c^{0.25}}\right) - lA_s i_c + lA_s i_s\,. \quad \{6\}$$

The total environmental impact associated with building a member is a function of not only the impacts per unit volume associated with each concrete mixture and steel rebar, but also depends on the quantity required for each material (concrete and steel) to satisfy design performance and is thus inversely related to the compressive strength. Considering applications where the area and strength of steel are held constant while the environmental impacts of different concrete mixtures are assessed, the final term in equation {6} can be dropped. In addition, the area of steel required for a member designed exclusively for initial cracking is very small compared to the total area of concrete (<2%) and can be neglected. Therefore, the total environmental impact from a concrete member in such conditions can be approximated with the first term in equation {6}. Hence, Kourehpaz and Miller proposed an index to compare different concrete mixtures based on initial cracking due to bending as

$$\chi_{cracking} = \frac{i_c}{f_c^{0.25}}\,, \quad \{7\}$$

where the objective consists in minimizing this term to reduce environmental impact associated with the designed member. Indices were also developed for the yield and nominal stages of bending in (Kourehpaz and Miller 2019). At these stages of bending, the required volume of material is not only a function of the concrete compressive strength. For instance, for the yield stage, the required volume of material is also highly dependent of the yield strength and area of steel, for which moment-curvature relationships must be considered. The comparison indices for these stages of bending were developed from empirical relationships between volume, compressive strength of concrete and area of steel, with boundary conditions that limit these equations to normal strength concretes, constant steel area and strength, w/cm between 0.25 and 1.25, and depth as required by ACI (ACI 2011). To maintain focus on the tools developed in this study rather than developing a new set of indices for UHPC, indices associated with these stages of bending will not be evaluated in this study. Thus, neither the influence of mix proportioning on the amount of steel required nor the environmental impact associated with steel will be assessed in the present study.

One shortcoming from the approach used by Kourehpaz and Miller is that it assumes the same equation {5} to estimate the modulus of rupture f_r of very different classes of concretes, with strengths ranging from between 0 and 100 MPa, while this equation was empirically derived for normal strength concretes. State-of-the-art reports from ACI 363 (ACICommittee363 2010) and ACI 239 (Russell, Graybeal, and Russell 2013) have defined new equations to estimate the modulus of rupture associated with high strength concretes (HSC) and UHPC, respectively. Hence, modifications to the $\chi_{cracking}$ index will be proposed later in this study to account for the influence of compressive strength in estimating the modulus of rupture.

Similarly, an index that compares the axial capacity of short columns (effects of buckling are negligible) will be used in this study, assuming a constant strength and cross-sectional area of steel. Detailed derivations for this case are discussed in (Kourehpaz and Miller 2019), where the index was defined as

$$\chi_{column} = \frac{i_c}{f_c}. \quad \{8\}$$

6.1.2. Proposed Modification to the Initial Cracking Comparison Index

As mentioned earlier in section 6.1, the $\chi_{cracking}$ developed by Kourehpaz and Miller (Kourehpaz and Miller 2019) was derived assuming the same equation ({5}) to estimate the modulus of rupture f_r of very different classes of concretes, for which the compressive strengths ranged from 0 to 100 MPa, when this equation was empirically derived for normal strength concretes. According to ACI 363 (ACICommittee363 2010), the modulus of rupture f_r for concretes with compressive strengths over 55MPa (HSC) ranges between $0.62\sqrt{f_c(MPa)}$ and $0.99\sqrt{f_c(MPa)}$, and is typically estimated as

$$f_r = 0.94\sqrt{f_c(MPa)}. \quad \{9\}$$

On the other hand, ACI 239 (Russell, Graybeal, and Russell 2013) estimates f_r for UHPC as

$$f_r = 6.7\sqrt{f_c(ksi)}, \quad \{10\}$$

which is equivalent to $2.55\sqrt{f_c(MPa)}$. This equation is only applicable to untreated specimens. For UHPCs subjected to special curing, other coefficients are suggested (Russell, Graybeal, and Russell 2013).

Considering these different equations for different classes of concrete, the resulting coefficient in the first term (1.1) of equation {6} should be kept inside the comparison index to account for the differences in the equations for the modulus of rupture. Hence, substituting equation {3} along with equations {5} (for NSC), {9} (for HSC) or {10} (for UHPC) inside equation {4}, results in a modified $\chi_{cracking}$ equation, defined as

$$\chi_{cracking} = \lambda_{fr}\left(\frac{i_c}{f_c^{\,0.25}}\right), \qquad \{11\}$$

where λ_{fr} is a coefficient associated with the equation selected to estimate the modulus of rupture. For equations {5}, {9} and {10} the values of λ_{fr} are 1.1, 0.89 and 0.54, respectively. This modification makes a significant difference for UHPC materials considering that this coefficient is less than half of that from NSC.

Meanwhile, defining λ_{fr} as a step function is not the best approach. For instance, assuming a step function approach for λ_{fr} , if we define a 54 MPa as a NSC and a 56 MPa as an HSC, the difference in modulus of rupture would look significantly higher than what it actually is. Instead, it makes more sense to model λ_{fr} as a function of compressive strength. Between other performance requirements, ACI 363 (ACICommittee363 2010) defines HSC as concretes with compressive strengths over 55 MPa, while ACI 239 (Russell, Graybeal, and Russell 2013) defines UHPC for concretes with compressive strengths in excess of 150 MPa. Given the λ_{fr} values

obtained from the different f_r equations and the compressive strength boundaries defined for each class of concrete, an equation for λ_{fr} as a function of f_c was derived as

$$\lambda_{fr} = 1.1125e^{-0.004 f_c}, \quad \{12\}$$

with f_c in units of MPa.

Therefore, the environmental analysis performed in the study will consider the proposed modified $\chi_{cracking}$ from equation {11}, using a non-constant λ_{fr} as proposed in equation {12}, and the χ_{column} originally proposed by Kourehpaz and Miller (Kourehpaz and Miller 2019).

6.1.3. Proposed Comparison Indices for Cost-Efficient Design of Concrete Members

Equations {8} and {11} were adopted and modified to evaluate the cost-efficiency of different concrete mixtures. For these indices, the volumetric environmental impact is replaced with the volumetric cost associated with each concrete mixture. The volumetric cost for each mixture was calculated as a product of the quantities used for each ingredient per cubic meter of concrete and their unit price. For simplification purposes, the costs associated with concrete production, transportation and placement were assumed to be the same for all mixtures. In reality, proper considerations should be done in design stages to account for the differences in cost associated with different concrete technologies, required production equipment, amount of labor, construction timeframe, etc.

Hence, the comparison indices used to evaluate cost-efficiency for the applications aforementioned are defined as

$$\rho_{column} = \frac{u_c}{f_c} \quad \{13\}$$

and

$$\rho_{cracking} = \lambda_{fr} \left(\frac{u_c}{f_c^{0.25}} \right), \quad \{14\}$$

where u_c is the mixture cost per unit volume of concrete, ρ_{column} is the index used for short columns while $\rho_{cracking}$ is the index associated with the initial cracking of simply supported and uniformly loaded beams subjected to bending.

6.2. Environmental Impact Factors and Unit Costs of Raw Ingredients

To maintain the focus of this work on developing machine-learning-based tools that demonstrate the influence of mixture proportioning and mechanical performance on carbon footprint, the environmental factors used were derived from Celik et al. (Celik et al. 2015). In that study, a life-cycle assessment (LCA) approach was used to assess the environmental impact and use of resources (including raw materials, energy, and water) during the entire life-cycle of each product, that is, from raw material production, use, maintenance, recycling, and ultimate disposal. Table 6 illustrates how the volumetric GWP (kg CO_2-eq/m^3 of concrete) from each ingredient was used to derive the GWP factor on a weight basis (kg CO_2-eq/kg) required to assess the mixtures from the present study. The GWP associated with slag and microsilica were assumed equivalent to the one estimated for fly ash in Celik et al (Celik et al. 2015). While SCMs are often modeled as waste products with negligible GHG emissions associated, studies have shown that the impacts associated with their primary industrial processes can be significant and care should

be taken when making these assumptions (Brinkman and Miller 2021). To estimate the GWP associated with the crushed sand and ground quartz aggregates used in this study, a production GWP associated with the coarse aggregate from Celik et al. (Celik et al. 2015) was assumed while adding aggregate crushing factors estimated from Santero et al. (Santero et al. 2011) to differentiate the environmental impact from these aggregates. Considering that transportation is the main source of GWP associated with the use of aggregates, a transportation component was added to all aggregates. For simplicity purposes, unit costs from each ingredient were estimated based on various sources (alibaba.com ; District ; HomeGuide 2021; KEMCORE 2021; Tadros and Morcous 2009; van Oss 2003) rather than selecting a specific location and the corresponding associated costs.

Table 6. Estimated GWP and unit cost factors defined on a per-weight basis

	GWP (kg CO_2-eq/m^3 of concrete)				*batching weights	*GWP factor	[b]unit cost
	Production (Mix 70-30-0)	*Production (present work)	[a]Aggregate crushing	*Aggregate transportation	(kg/m^3 of concrete)	(kg CO_2-eq/kg)	($/ton)
Slag	-	3	-	-	-	0.022	$70
Microsilica	-	3	-	-	-	0.022	$600
Fly Ash	3	3	-	-	136	0.022	$25
cement	363	363	-	-	317	1.145	$90
Water	-	-	-	-	-	0.005[a]	$0.80
HRWR	5	5	-	-	6.3	0.794	$3,000
Fine Agg.	4.3	4.3	-	29.7	906	0.038	$10
Coarse Agg.	2.5	-	-	29.7	906	-	-
Crushed sand	-	2.5	2.72	29.7	906	0.039	$40
Ground quartz	-	2.5	5	29.7	-	0.041	$220

*derived from Celik et. al (Celik et al. 2015) using mix 70-30-0. This mixture contained:
- a binder proportioned (by wt.) at 70% cement and 30% fly ash.
- a fine-aggregate-to-cementitious ratio of 2.00
- a course-aggregate-to-cementitious ratio of 2.00
- a HRWR content of 1.39% by wt. of cementitious

[a]obtained from Santero et al. (Santero et al. 2011)

[b]estimated based on (alibaba.com ; District ; HomeGuide 2021; KEMCORE 2021; Tadros and Morcous 2009; van Oss 2003)

6.3. Predicted Compressive Strengths

As previously mentioned, all mixtures in Phase B have the same proportioning of cementitious materials in the binder phase (68.3% cement, 22.4% slag, 5.3% microsilica and 2% fly ash) and a fixed w/cm equaling 0.20. Fig. 21 illustrates the PDD describing the predicted compressive strengths of UHPC mortars with varying aggregate content, while faceting the ground quartz content.

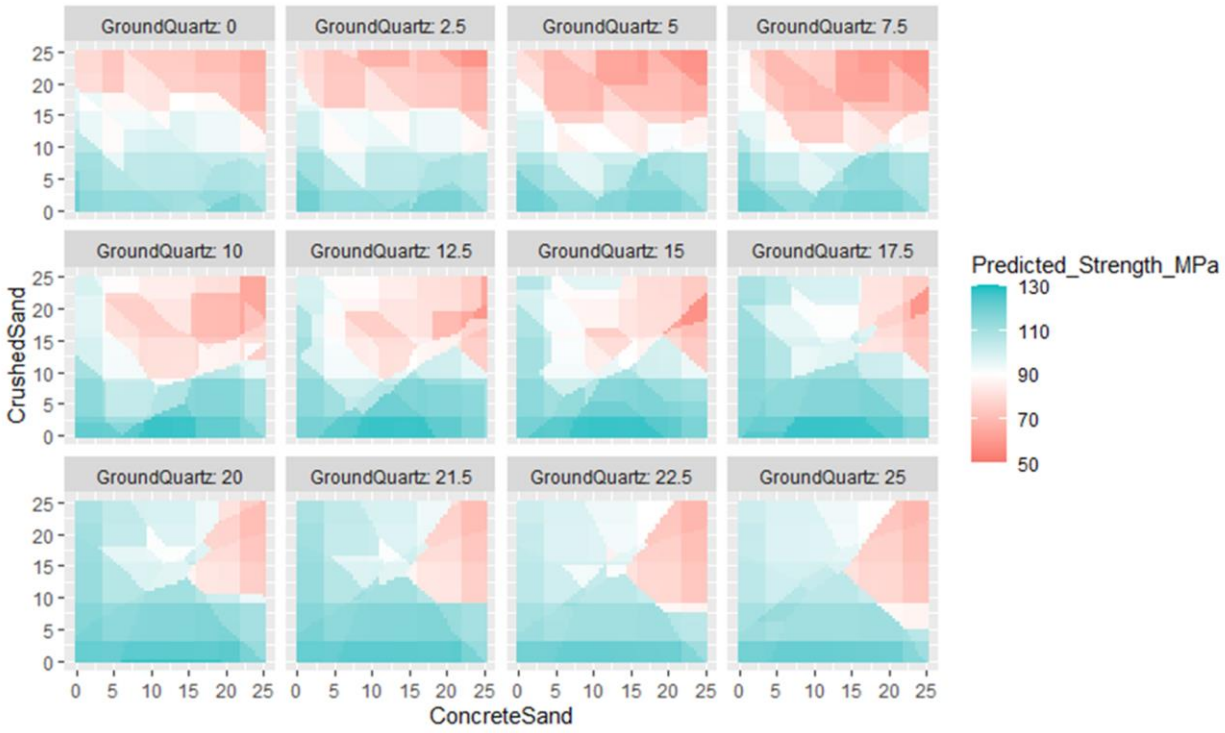


Fig. 21. PDD describing the predicted compressive strengths for varying contents of concrete sand, crushed sand and ground quartz

As shown, the mixtures with the highest compressive strengths are mostly associated with low contents of crushed sand (<10%) and intermediate contents of concrete sand and ground quartz (approximately 10% to 17.5%). High strengths can also be obtained with high crushed sand contents (20-25%) as long as sufficient ground quartz is introduced in the system (15 to 21.5%) while simultaneously keeping concrete sand at very low contents (<3%).

This trend intuitively makes sense. Increasing the content of both sands simultaneously decreases the strength of the matrix given the increasing area of interfacial transition zones (ITZs). Meanwhile, the overall improvement provided by ground quartz is expected due to its "filler" role, improving particle packing. The red half-diamond regions located in the right side of the 4 lowest blocks (ground quartz between 20 and 25%) suggest that strengths over 100 MPa can still be obtained as long as high contents of crushed sand and concrete sand are not used simultaneously and there is sufficient ground quartz in the system. Overall, decreased performance is observed when the high contents of both sands are used simultaneously.

6.4. Volumetric Environmental Impacts vs Eco-Efficiency Density Diagrams

Fig. 22 illustrates the linear relationship between volumetric GWP and mix proportions based on the contribution from each material, which is demonstrated in faceted counterplots to facilitate comparisons with the diagrams developed herein. As expected, lower values of volumetric GWP are associated with high sand contents (concrete sand and crushed sand), located at the top right corners of the lower-right blocks (increasing ground quartz content). However, eco-efficiency should not be defined on a per-volume of material basis. While mix proportioning defines mechanical properties, the effect of mechanical performance on design defines the amount of concrete required for a given application and the associated total GHG emissions. Therefore, eco-efficiency indices discussed in section 6.1 are used to allow proper environmental evaluations associated with different mix proportions while accounting for mechanical performance. To calculate these indices, the weight of each ingredient required to produce one cubic meter of concrete is multiplied by the corresponding GWP factors from Table 6. Summing the contribution

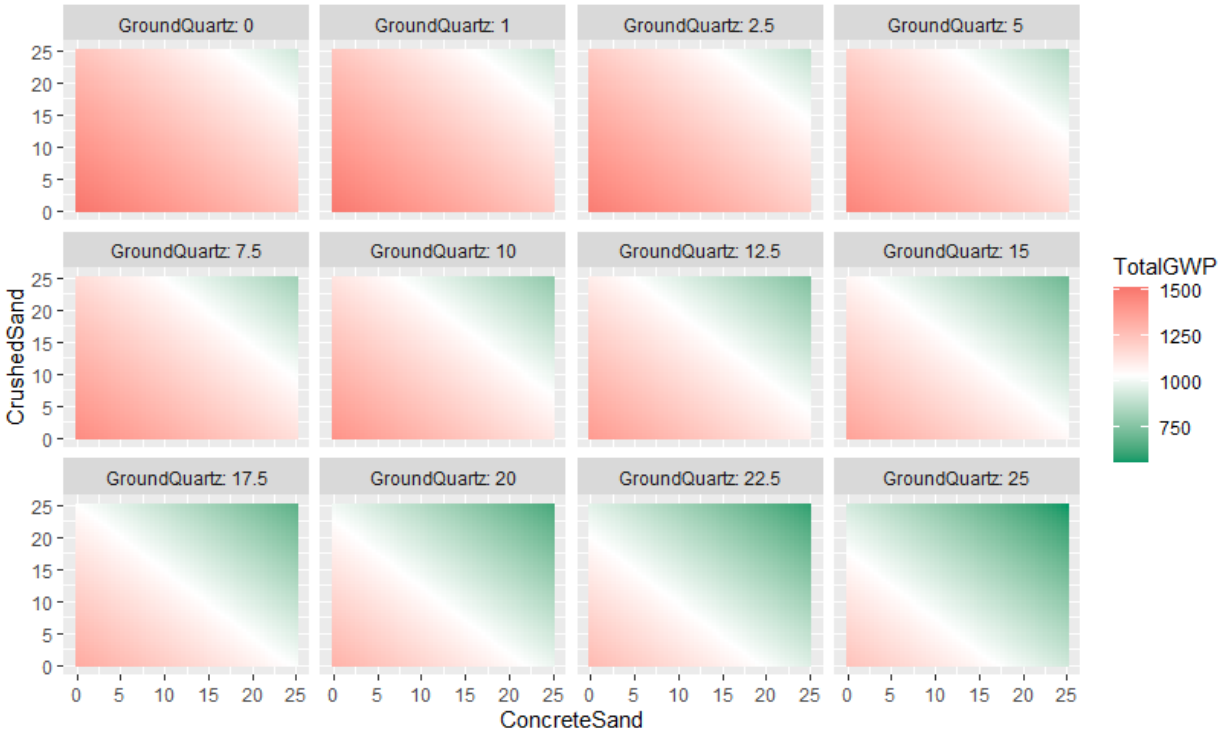


Fig. 22. Volumetric GWP (kg $CO_{2\text{-eq}}/m^3$) for varying mixture proportioning associated with the contribution from each ingredient

from all ingredients results in the volumetric GWP (kg CO2-eq/m^3), i_c, associated with the corresponding mixture. Meanwhile, the corresponding compressive strength is estimated as previously mentioned. The resulting χ_{column} index, as a function of mix proportioning, is illustrated in Fig. 23 through an Eco-Efficiency Density Diagram (EEDD). Its definition follows the one used for PDDs in, with exception of the objective function (outcome evaluated), which in this case consists in an eco-efficiency index. Observing Fig. 23, for short column applications, the most eco-efficient mixtures can be obtained in two ways: 1) using high contents of ground quartz (20 to 25%), high contents of crushed sand (20-25%), and medium-to-high contents of concrete sand (15-20%); or 2) using high contents of ground quartz (20 to 25%), low contents of crushed sand (<9%), and high contents of concrete sand (17.5-25%). The trends observed in Fig. 21 regarding the use of high contents of both sands simultaneously are still visible for this comparison index,

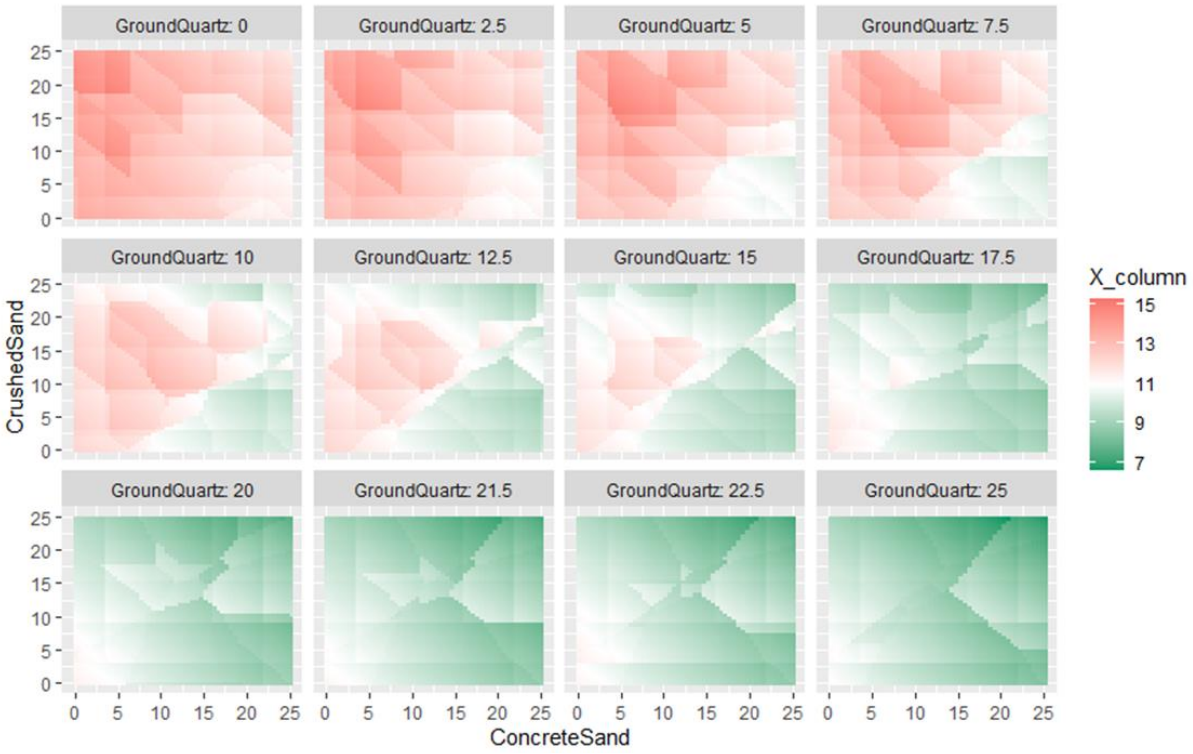


Fig. 23. EEDD to evaluate mix proportioning versus the eco-efficiency index χ_{column}

where half-diamond regions with lower densities are evident in the right side of the 4 lowest blocks, suggesting that the most eco-efficient solutions are obtained when high contents of both sands are not used simultaneously.

For $\chi_{cracking}$, an EEDD is built using the same method described above and is shown in Fig. 24. As it can be observed, this diagram is somewhat in between the EEDD from Fig. 23 and the linear relationship from Fig. 22, as the trends are similar to the ones observed for χ_{column}, although with a decreased influence from mechanical performance. This is expected considering the small differences between the GWP factors associated with these aggregates, along with the fact that f_c is to the power of ¼ in the $\chi_{cracking}$ equation. Yet, and similar to what is observed for χ_{column}, the EEDD for $\chi_{cracking}$ still suggests that eco-efficiency is not automatically improved by simply increasing aggregate content to replace binder.

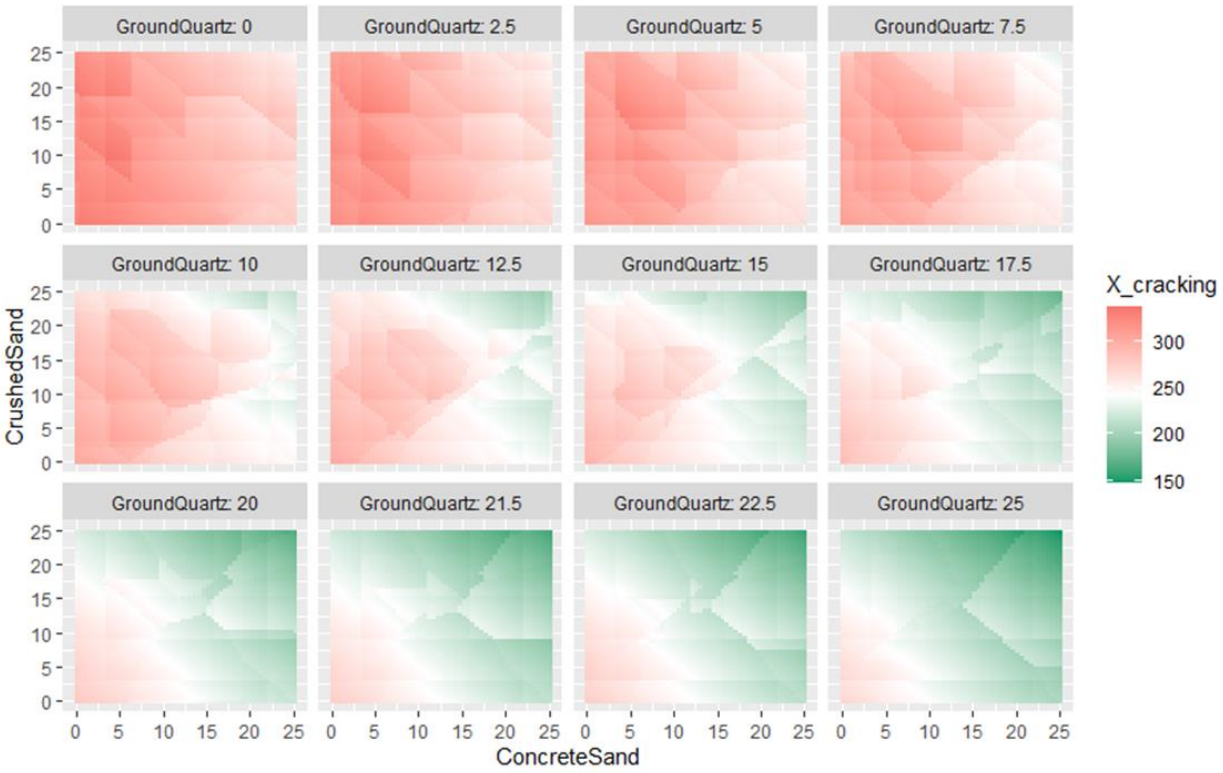


Fig. 24. EEDD illustrating mix proportioning effect on eco-efficiency index $\chi_{cracking}$

6.5. Volumetric Mixture Cost vs Cost-Efficiency Density Diagrams

The same approach followed to develop EEDDs can be used to develop Cost-Efficiency Density Diagrams (CEDDs) to evaluate the effect of mixture proportioning and mechanical performance on cost-efficiency associated with building a given structural member. Fig. 25a) and b) illustrate the CEDDs corresponding to the ρ_{column} and $\rho_{cracking}$ indices, respectively. Meanwhile, Fig. 26 shows the linear relationship existing between volumetric unit cost ($/m^3), u_c, and mix proportioning. The difference in trends observed for unit cost and cost-efficiency is even more accentuated than the one observed between volumetric environmental impact and eco-efficiency. This is mostly due to the significant differences in cost between these three aggregates, in addition to the effect of mechanical performance on the required volume of

concrete. As observed in Table 6, concrete sand costs about four times less than crushed sand, which in turn costs over five times less than ground quartz.

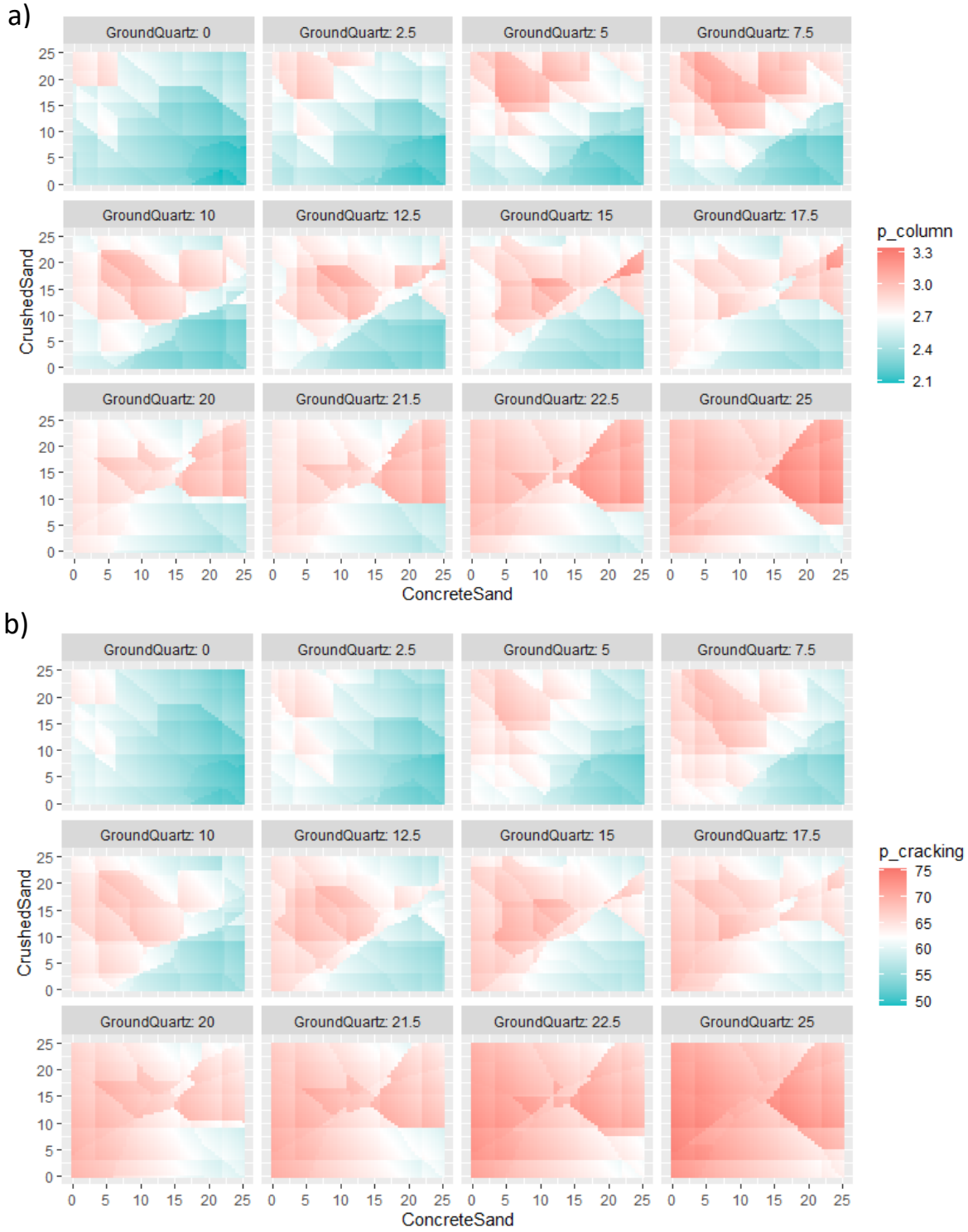


Fig. 25. CEDD illustrating mix proportioning effect on: a) ρ_{column}; b) $\rho_{cracking}$

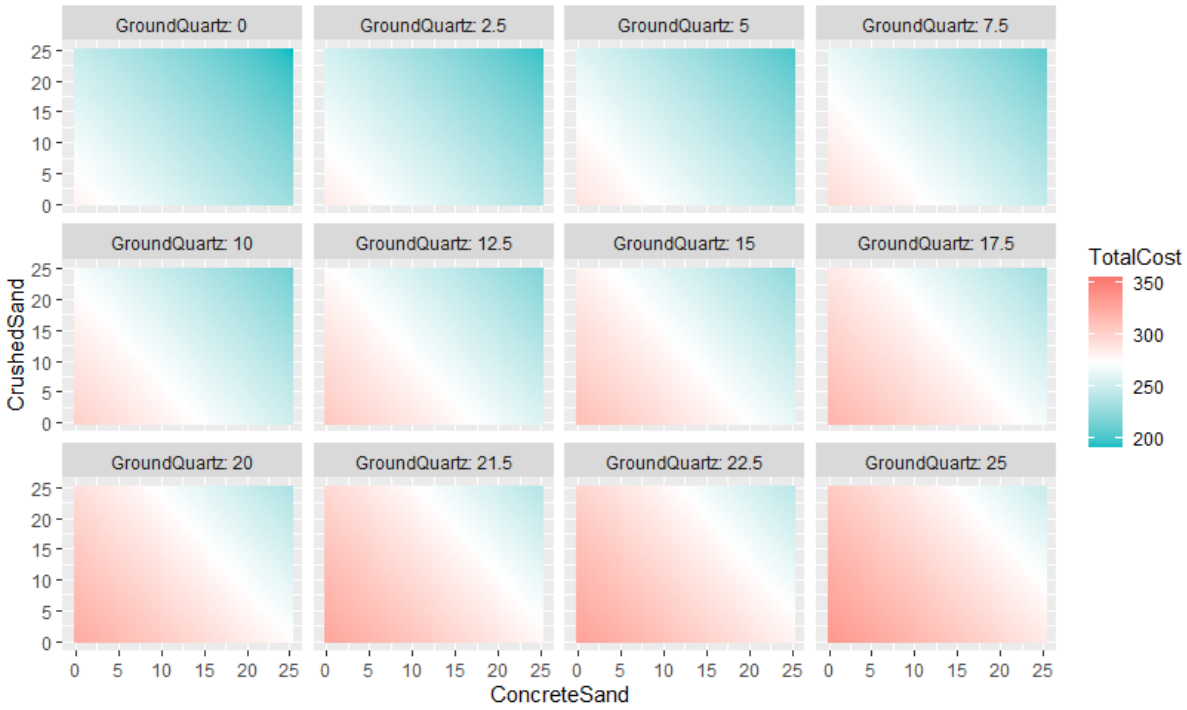


Fig. 26. Volumetric cost ($/m^3) for varying mixture proportioning associated with the contribution from each ingredient

As expected, Fig. 26 indicates that mixtures with the lowest cost per unit volume ($/m3) are associated with low contents of ground quartz (top left faceted blocks) and high contents of both sands (top right corners). In contrast, Fig. 25 indicates that the most cost-efficient solutions are associated with low contents of crushed sand (<7.5%) and ground quartz (<2.5%), combined with medium-to-high contents of concrete sand (17.5 to 22.5%). As previously noted, the costs defined in this study are associated with mix proportioning only and do not account for differences in required equipment, technology, labor and service-life maintenance. Proper considerations should be incorporated in the analysis when comparing concrete technologies that significantly differ in these aspects (e.g., NSC vs UHPC).

6.6. Comparing Solutions using Efficiency Indices vs Volumetric Indicators

Table 7 and Table 8 show mixtures of particular interest, selected to discuss efficiency indices.

Table 7. Optimum solutions for columns defined with different indicators

Mix #	Description	aggregates (% by wt) replacing cementitious content: concrete sand	crushed sand	ground quartz	Total Cost ($/m³)	ρ_column	fc (MPa)
ρ-Col-1	with min(ρ_column)	21.5	2.5	0	233	2.08	112
ρ-Col-2	with min(Total Cost)	25	25	0	190	2.37	80
ρ-Col-3	same ρ_column ρ-Col-2 & higher cost	11	2.5	11.5	279	2.37	118
ρ-Col-4	with lowest predicted strength	22	17.5	14	240	3.21	74.8
ρ-Col-5	with highest predicted strength	9.5	0	19	303	2.56	118
					Total GWP (kg CO_2e/m³)	χ_column	fc (MPa)
χ-Col-1	with min(χ_column)	20.5	25	25	625	6.58	95
χ-Col-2	with min(Total GWP)	25	25	25	552	6.75	82
χ-Col-3	same χ_column as χ-Col-2 & higher GWP	19.5	25	25	641	6.75	95
χ-Col-4	with lowest predicted strength	22	17.5	14	875	11.70	74.8
χ-Col-5	with highest predicted strength	9.5	0	19	1197	10.10	118

Table 8. Optimum solutions for initial cracking due to bending, defined with different indicators

Mix #	Description	aggregates (% by wt) replacing cementitious content: concrete sand	crushed sand	ground quartz	Total Cost ($/m³)	ρ_cracking	fc (MPa)
ρ-Crk-1	with min(ρ_cracking)	25	7	0	219	50	105
ρ-Crk-2	with min(Total Cost)	25	25	0	190	51.2	80
ρ-Crk-3	false inefficiency suggested by linear equations	7.5	0	15	297	62.7	118
ρ-Crk-4		0	0	10	300	64.5	115
ρ-Crk-5		4	13	19	297	67.9	105
ρ-Crk-6	false efficiency suggested by linear equations	25	21	15	230	64.4	74.9
ρ-Crk-7		25	25	7.5	236	65.7	75.7
ρ-Crk-8		15.5	3	10	266	56.1	118
					Total GWP (kg CO_2e/m³)	χ_cracking	fc (MPa)
χ-Crk-1	with min(χ_cracking)	25	25	25	552	147	81.7
χ-Crk-2	with min(Total GWP)	25	25	25	552	147	81.7

Table 7 confirms the trends previously identified in EEDDs and CEDDs, where mixtures with the minimum volumetric cost (ρ-Col-2) and minimum volumetric GWP (χ-Col-2) are neither the most

cost nor eco-efficient solutions, respectively. The ρ_{column} value for the mixture with the lowest u_c (ρ-Col-2) is 2.37, while the mixture with the minimum ρ_{column} (ρ-Col-1) has a corresponding ρ_{column} value of 2.08. The increased unit cost of ρ-Col-1 (u_c = \$219/m^3) compared to ρ-Col-2 (u_c = \$190/m^3) is offset by the increased compressive strength (f_c = 112 MPa vs f_c = 80 MPa, respectively). Therefore, this mixture would result in a lower overall cost associated with building this type of member, given the reduction in volume of concrete required. Similarly, mixture χ-Col-1, which has an i_c of 625 kg CO_2-eq/m^3, is shown to be more eco-efficient than the mixture with minimum i_c, χ-Col-2, in 552 kg CO_2-eq/m^3, based on their corresponding χ_{column} values (6.58 and 6.75, respectively). The optimum mixture proportion in terms of ρ_{column} follows the trends discussed for Fig. 25, where high contents of concrete sand and low contents of ground quartz and crushed sand tend to minimize ρ_{column}, and thus improve cost-efficiency. Similarly, the optimum χ_{column} follows the trend observed in Fig. 23. Mixture p-Col-3 shows that a cost-efficiency similar to the one obtained with the minimum total cost (ρ-Col-2) can be achieved with a much more expensive solution (u_c = \$279/m^3) on a unit volume basis, as long as the mechanical performance (f_c = 118 MPa) offsets the difference in unit cost. Meanwhile, ρ-Col-4 and ρ-Col-5, corresponding to the lowest and highest performing mixtures in compressive strengths, respectively, are neither amongst the cheapest solutions (u_c = \$303/m^3 and \$240/m^3, respectively) nor amongst the most cost-efficient solutions (ρ_{column} = 3.21 and 2.56, respectively). This is a strong indicator on how optimization efforts should not be oversimplified. Mix proportioning optimization should be conducted considering costs, emissions and mechanical

performance concurrently. Focusing all efforts on identifying a mixture with the best mechanical performance, alone, may result in inferior solutions in terms of cost-efficiency, while similar mechanical performance may be obtained through a properly designed, more cost-effective alternative.

Considering the initial cracking application, Table 8 indicates that $\chi_{cracking}$ is less affected by changes in mechanical performance. For this case, the most eco-efficient solution coincided with the mixture with minimum volumetric environmental impact. However, this is not an overall trend, as observed in Fig. 24. Moreover, this trend might not necessarily translate to cases where indices account for the environmental impact of steel in design (Kourehpaz and Miller 2019). This aspect is of vital importance for UHPC materials, considering the significant reductions attained in the required amount of steel. In addition, longer spans can be achieved with this material in bridge elements (beams, girders) due to the increased prestressed levels, which not only reduces the weight of the superstructure, but also reduces the number and volume of supporting substructure elements in columns, footing and piles. This trend might also change for other sets of ingredients with significantly different GWP factors. Further evidence supporting this argument is provided by Table 8, where cost-efficiency for cracking does not follow the u_c trends. While the cheapest solution is associated with mix ρ-Crk-2, characterized by the absence of ground quartz and high contents of concrete sand and crushed sand, the most efficient $\rho_{cracking}$ corresponds to mix ρ-Crk_1, which has a high content of concrete sand, medium-to-low content of crushed sand and no ground quartz.

EEDDs and CEDDs provide a flexible tool to evaluate different alternatives that may not necessarily target the minimum performance requirements. For instance, if the minimum f_c required by design is prescribed as 75 MPa for initial cracking due to bending, using the PDD in Fig. 21 alone to optimize f_c and Fig. 26 to evaluate unit cost, could lead to selecting a mixture somewhat similar to ρ-Crk-6, with an f_c of 75.7 MPa and a u_c equal to \$236/m^3. However, using the CEDD from Fig. 25 b) could lead to selecting mixture ρ-Crk-8, with an f_c of 118 MPa and u_c of \$266/m^3, while still more cost-effective than ρ-Crk-6.

From observing the developed CEDDs, EEDDs and Table 7 and Table 8, it is clear that the optimum mixtures for each one of the objective functions (compressive strength, cost and eco-efficiency), ρ-Col-5, ρ-Col/Crk-1 and χ-Col/Crk-1, respectively, can be significantly different. Similarly, the optimum mixtures for different structural members/applications can significantly differ from each other.

6.7. Summary of Important Outcomes

The emergence of UHPC as an attractive solution for precast and prestressed applications has coincided with global efforts towards sustainable construction. The increasing need for tools capable of intuitively demonstrating the effect of concrete mixture composition on mechanical performance, cost and eco-efficiency concurrently has motivated this work in an effort to promote design of more sustainable solutions to help meet environmental goals. Modifications to eco-efficiency indices developed by Kourehpaz and Miller (Kourehpaz and Miller 2019) were proposed, while cost-efficiency indices were adapted from these for a simplified demonstration

of cost implications. Eco- and cost-efficiency indices were used as outputs in orthogonal machine-learning based tools, developed to evaluate changes in cost and eco-efficiency associated with changes in mixture proportioning. Key findings from this work consist of:

- The EEDDs and CEDDs developed in this study have shown to be ideal tools for concrete mixture optimization, where cost-effective and eco-friendly mix designs can easily be determined. These diagrams also facilitate the identification of several alternative solutions to account for material availability and accessibility when production takes place in different locations;
- Optimum mixtures for cost and eco-efficiency can significantly differ from each other. Similarly, optimum mixtures, in terms of cost and eco-efficiency, for different structural members can significantly differ from each other;
- Due to differences in mechanical performance, mixtures with high volumetric environmental impact can still be more sustainable than mixtures with nearly half the volumetric environmental impact. The same trend was observed in terms of cost-efficiency. For policy makers, the implication is that simply targeting concrete mixtures with the lowest cost or lowest embodied emissions on a per volume basis does not, generally, result in the lowest cost or lowest embodied emissions for a given structural element or infrastructure project;
- Mix proportioning optimization should be conducted considering costs, emissions and mechanical performance concurrently. Focusing all efforts on identifying a mixture with the best mechanical performance alone may result in less efficient solutions, while similar

mechanical performance may be obtained with much more cost and eco-effective alternatives;

Future work should focus on developing new indices that account for the environmental impact of reinforcing steel on eco-efficiency and further development of machine-learning based density diagrams for material properties other than compressive strength.

As a contribution to concrete producers, project owners, practicing engineers, designers and regulating entities, the tools developed herein could serve as decision-making aids during mix design stages and provide proof of mixture optimization that can be introduced alongside Environmental Product Declarations (EPD).

6.8. References

ACI, ACI. 2011. '318-11: Building Code Requirements for Structural Concrete', *Farmington Hills, MI: American Concrete Institute*, 505.

ACICommittee363. 2010. "PRC-363-10 Report on High-Strength Concrete." In, 65.

alibaba.com. https://www.alibaba.com/product-detail/High-range-water-reducing-HRWR-admixture_60623774798.html.

Brinkman, Leah, and Sabbie A Miller. 2021. 'Environmental impacts and environmental justice implications of supplementary cementitious materials for use in concrete', *Environmental Research: Infrastructure and Sustainability*, 1: 025003.

Celik, Kemal, Cagla Meral, A. Petek Gursel, P. Kumar Mehta, Arpad Horvath, and Paulo J. M. Monteiro. 2015. 'Mechanical properties, durability, and life-cycle assessment of self-consolidating concrete mixtures made with blended portland cements containing fly ash and limestone powder', *Cement and Concrete Composites*, 56: 59-72.

District, North Texas Municipal Water. 'Water Rates'. https://www.ntmwd.com/water-rates/.

HomeGuide. 2021. 'How Much Does Crushed Stone or Gravel Cost?'. https://homeguide.com/costs/gravel-prices.

KEMCORE. 2021. https://kemcore.com/products/silica-flour-quartz-powder-99.

Kourehpaz, Pouria, and Sabbie A. Miller. 2019. 'Eco-efficient design indices for reinforced concrete members', *Materials and Structures*, 52: 96.

Miller, Sabbie A., Paulo J. M. Monteiro, Claudia P. Ostertag, and Arpad Horvath. 2016. 'Comparison indices for design and proportioning of concrete mixtures taking environmental impacts into account', *Cement and Concrete Composites*, 68: 131-43.

Russell, Henry G, Benjamin A Graybeal, and Henry G Russell. 2013. "Ultra-high performance concrete: A state-of-the-art report for the bridge community." In.: United States. Federal Highway Administration. Office of Infrastructure ….

Santero, Nicholas, Alexander Loijos, Mehdi Akbarian, and John Ochsendorf. 2011. "Methods, impacts, and opportunities in the concrete pavement life cycle." In.: MIT Concrete Sustainability Hub.

Tadros, Maher K, and George Morcous. 2009. "Application of ultra-high performance concrete to bridge girders." In.: Nebraska Transportation Center.

van Oss, Hendrik G. 2003. 'Slag-iron and steel', *US geological survey minerals yearbook*, 1.

CHAPTER VII.

MULTI-OBJECTIVE DENSITY DIAGRAMS

The process of methodically and concurrently optimizing a set of objective functions is known as multi-objective optimization or vector optimization. Different approaches can be followed in these types of problems to model a decision maker's preference depending on how the user articulates these preferences. *Prior articulation of preferences* implies that the relative importance of each objective function is determined by the decision-maker prior to the optimization process. On the other hand, a *posteriori articulation of preferences* involves selecting a particular solution from a set of mathematically equivalent solutions. Methods that require no articulation of preferences have also been investigated (Nash 1950; Rao 1987). Further discussion on multi-objective methods can be found in the literature (Marler and Arora 2003), including important concepts such as Pareto optimality (Laponce 1972). To maintain focus on the scope of this work, the multi-objective formulation herein follows a classic weighted summation of each objective function (cost- and eco-efficiency indices), which are normalized by the worst solutions available in their corresponding domains. This simplified approach was followed considering that both cost and emissions are objective functions for which optimum solutions are associated with minimizing their outcomes.

Other function transformation methods can be followed (Koski 1981; Koski and Silvennoinen 1987; Osyczka 1978; Proos et al. 2001) to improve robustness of the optimization process. In addition, other methods can be followed for multi-objective optimization problems involving competing objective functions. For instance, if an optimization problem consists in minimizing one outcome (e.g., cost) while maximizing another outcome (e.g., flowability), methods such as Grey relational analysis (Ju-Long 1982) can be used to normalize these outcomes to factors generated with different objectives (lower-the-better vs higher-the-better) and use them concurrently in one single objective function. Other approaches can

also be followed for the multi-objective function formulation to address problems where the weighted summation is unable to capture optimum points. Example of such methods are the ε -constraint (Haimes and Hall 1974), weighted min-max method (Kaliszewski 1987), exponential weighted criterion (Athan and Papalambros 1996) and the weighted product method (Bridgman 1922).

In this chapter, global optimizations will be performed through multi-objective and multi-member comparison indices and the obtained results will be compared to data in the literature to evaluate cost and eco-efficiency of different concrete technologies.

7.1. Proposed Comparison Indices

7.1.1. Combined Cost and Eco-Efficient Indices for Multi-Objective Mixture Design

The concept of sustainable design, as defined by the US Green Building Council (USGBC), involves three components: environmental, economic and health, and community impacts. Choices made in the structural design and construction phases directly affect all three of these facets. Therefore, mixture design selection should be a balanced optimization function to accommodate all three of these objectives. Recently, governments and policy makers have pushed towards low carbon policies, where intensive-energy consumption industries are likely to be subjected to higher pricing policies while eco-friendly industries will be receiving incentives for their sustainable solutions. From the concrete producer's perspective, it is important to meet performance and sustainability requirements while maintain a cost-efficient business. Therefore, design efficiency should ultimately be measured using comparison indices that account for cost and emissions concurrently. To address this challenge, a multi-objective comparison index for a short column member MO_{column} can be defined as

$$MO_{column} = \alpha.\overline{\rho}_{column} + \beta\overline{\chi}_{column} \quad \{1\}$$

where $\overline{\rho}_{column}$ is the cost-efficiency index for short columns (ρ_{column}) normalized by the worst solution ($\rho_{\max}$) within the experimental domain, $\overline{\chi}_{column}$ is the column eco-efficiency index χ_{column} normalized by the worst solution ($\chi_{column,\max}$) within the experimental domain, and α and β are the weighting coefficients. The approach used to normalize the indices ensures that MO_{column} varies between zero and one, facilitating comparisons involving indices with different magnitudes. The same logic can be applied to calculate $MO_{cracking}$. It is tempting to predefine the weighting coefficient α, either by allocating the same weight to cost and environmental efficiencies $\alpha = 0.5$, or by shifting most of the weight to the environmental index in an effort to pursue the absolute, most environmentally-friendly solution. As shown in the subsequent section, this approach is not the most efficient considering that small shifts in environmental efficiency can be accompanied by enormous cost implications.

7.1.2. Combined Indices for Multi-Member Mixture Design Optimization

In the next subsections, results will suggest that the optimum concrete mixture identified for a given structural member might differ from the optimum for a different member/application. Yet, implementing different concrete grades in the same structure is highly dependent on the application and on local regulations and codes. To account for applications where one type of concrete must be used for the entire infrastructure, multi-member indices must be considered. Hence, multi-member eco-efficiency, $MM_{emissions}$, and multi-member cost-efficiency, MM_{cost}, can be defined, respectively, as

$$MM_{emissions} = \alpha.\overline{\chi}_{column} + \beta\overline{\chi}_{cracking} \quad \{2\}$$

and

$$MM_{cost} = \alpha.\overline{\rho}_{column} + \beta\overline{\rho}_{cracking} . \quad \{3\}$$

7.2. Multi-Objective Density Diagrams for Sustainable, Cost-Effective Solutions

As described in the previous section, multi-objective indices are a function of weighting coefficients allocated to cost and eco-efficiency. Similarly to cost- and eco-efficiency indices, optimum mixtures are obtained by minimizing these multi-objective indices. Fig. 27 through Fig. 29 illustrate how changes in the weighting coefficients affect the MO_{column} .

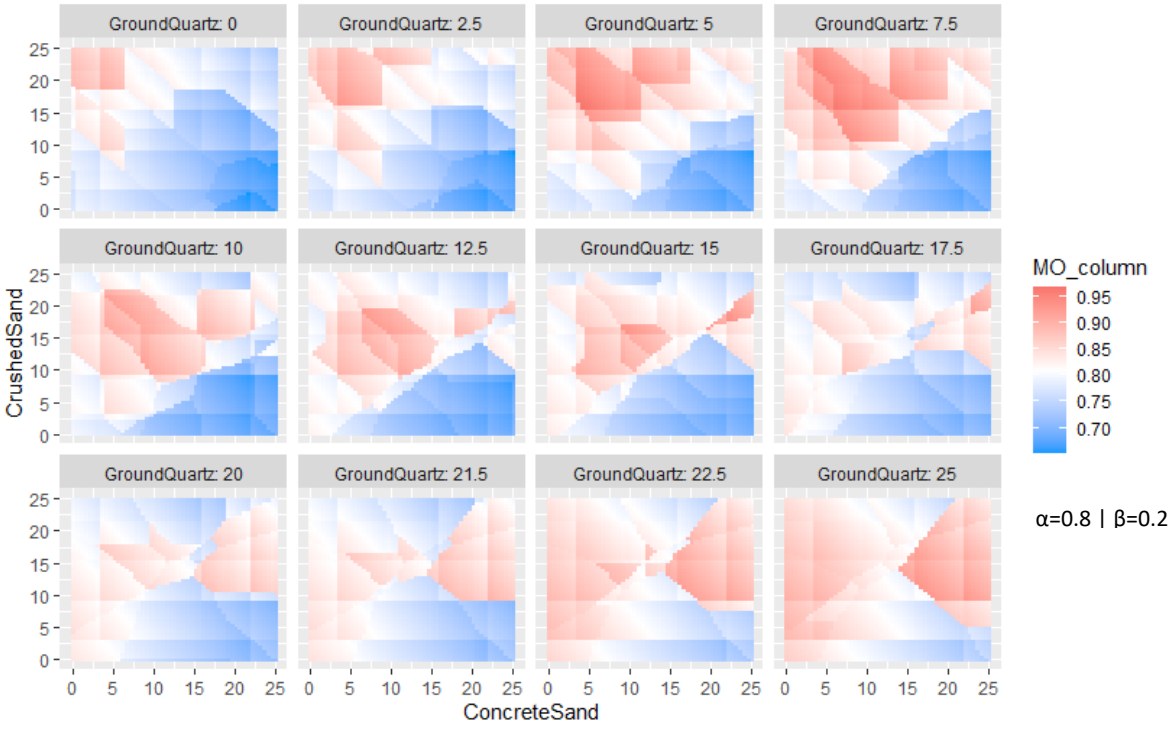


Fig. 27. MODD illustrating the mix proportioning effect on ρ_{column} and X_{column}, simultaneously, for α=0.8 and β=0.2

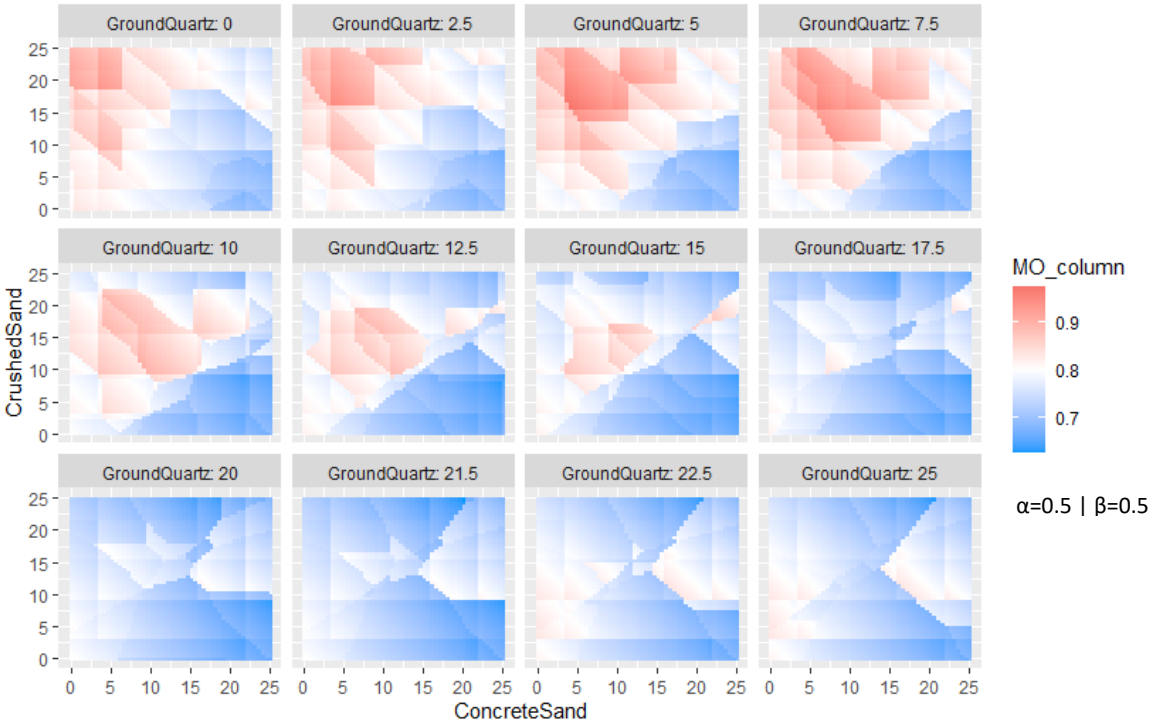


Fig. 28. MODD illustrating the mix proportioning effect on ρ_{column} and X_{column}, simultaneously, for α=0.5 and β=0.5

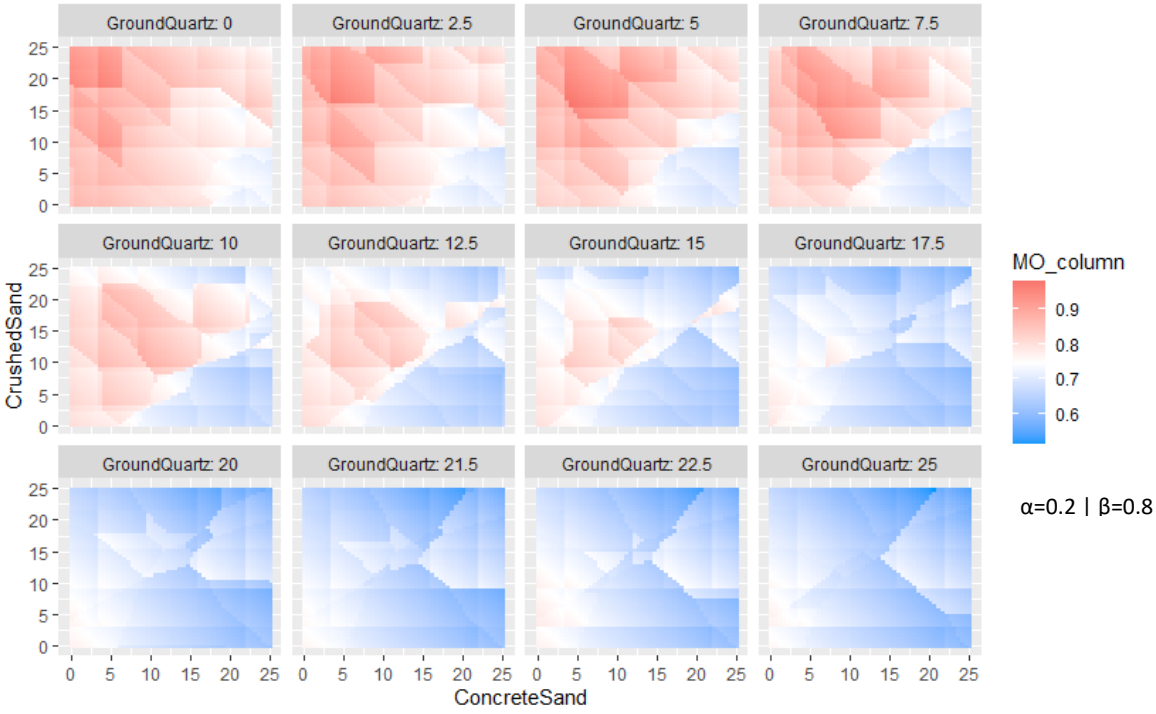


Fig. 29. MODD illustrating the mix proportioning effect on ρ_{column} and X_{column}, simultaneously, for α=0.2 and β=0.8

For developing policies and regulations, it would be tempting to fix these weighting coefficients. This could be done either by allocating the same weight to cost and environmental efficiencies, or by shifting most of the weight to the environmental index in an effort to pursue top eco-friendly solutions. However, there are advantages to viewing the optimization problem more holistically by carefully evaluating the tradeoffs when optimizing a given mixture for a given application. Table 9 provides information to support this argument.

Table 9. Optimum MO_column for varying values allocated to the weighting coefficients α and β

				optimized cost			optimized GWP			Agg content (% by wt. replacing OPC)			
Mix #	α	β	MO_{column}	ρ_{column}	*top pct	$\overline{\rho}_{column}$	χ_{column}	*top pct	$\overline{\chi}_{column}$	CS	CrS	GQ	f_c (MPa)
MO-Col-1	1	0	0.631	2.08	0%	0.631	11.1	52%	0.726	21.5	2.5	0	112
MO-Col-2	0.9	0.1	0.641	2.08	0%	0.631	11.1	52%	0.726	21.5	2.5	0	112
MO-Col-3	0.8	0.2	0.648	2.11	2%	0.642	10.2	42%	0.671	25	9	3.5	106
MO-Col-4	0.7	0.3	0.650	2.21	11%	0.673	9.08	29%	0.595	23.5	9	11.5	110
MO-Col-5	0.6	0.4	0.641	2.24	13%	0.680	8.90	27%	0.583	25	7.5	13	110
MO-Col-6	0.5	0.5	0.625	2.59	42%	0.787	7.08	6%	0.464	20	25	21.5	97
MO-Col-7	0.4	0.6	0.590	2.72	53%	0.828	6.58	0%	0.431	20.5	25	25	94.9
MO-Col-8	0.3	0.7	0.550	2.72	53%	0.828	6.58	0%	0.431	20.5	25	25	94.9
MO-Col-9	0.2	0.8	0.510	2.72	53%	0.828	6.58	0%	0.431	20.5	25	25	94.9
MO-Col-10	0.1	0.9	0.471	2.72	53%	0.828	6.58	0%	0.431	20.5	25	25	94.9
MO-Col-11	0	1	0.431	2.72	53%	0.828	6.58	0%	0.431	20.5	25	25	94.9

*top pct – top percentile
CS – Concrete Sand
CrS – Crushed Sand
GQ – Ground Quartz

	ρ_{column}	χ_{column}
min	2.08	6.58
max	3.29	15.3

For the short column case, while shifting most of the weight to the environmental index (β >0.5) can result in the best eco-efficient solution, it also results in significantly low rated mixtures in terms of cost-efficiency. As shown, the mixture corresponding to β >0.5 (MO-Col-7 to 11) is under the top 50th percentile in cost-efficient solutions, which is not an ideal scenario for any for-profit industry. Meanwhile, allocating equal weights to α and β helps identifying a better

solution in mixture MO-Col-6. While this mixture is still in the top 6th percentile for eco-efficiency, it represents a significant jump in cost-efficiency (11%) compared the previously mentioned mixtures, sitting at the top 42nd percentile. Yet, setting α at 0.6 and β at 0.4 provides, arguably, the most balanced solution for this scenario, where MO-Col-5 is in the top 27th percentile in terms of eco-efficiency, while sitting at the top 13th percentile for cost-efficiency.

Similar to what is provided in Table 9, Table 10 displays the optimum multi-objective index in terms of initial cracking due to bending, $MO_{cracking}$, for varying values of α and β.

Table 10. Optimum MO_cracking for varying values allocated to the weighting coefficients α and β

				optimized cost			optimized GWP			Agg content (% by wt. replacing OPC)			
Mix #	α	β	$MO_{cracking}$	$\rho_{cracking}$	*top pct	$\overline{\rho}_{cracking}$	$\chi_{cracking}$	*top pct	$\overline{\chi}_{cracking}$	CS	CrS	GQ	f_c (MPa)
MO-Crk-1	1	0	0.673	50.0	0%	0.673	263	61%	0.781	25	7	0	105
MO-Crk-2	0.9	0.1	0.683	50.3	1%	0.676	252	56%	0.748	25	9	2	105
MO-Crk-3	0.8	0.2	0.691	50.3	1%	0.676	252	56%	0.748	25	9	2	105
MO-Crk-4	0.7	0.3	0.697	50.7	3%	0.682	246	52%	0.732	25	9	3.5	106
MO-Crk-5	0.65	0.35	0.697	58.2	34%	0.785	179	17%	0.533	25	25	16	86
MO-Crk-6	0.6	0.4	0.684	60.4	43%	0.812	165	10%	0.491	20	25	21.5	97
MO-Crk-7	0.5	0.5	0.650	63.0	53%	0.848	152	3%	0.453	20.5	25	25	94.9
MO-Crk-8	0.4	0.6	0.611	63.0	53%	0.848	152	3%	0.453	20.5	25	25	94.9
MO-Crk-9	0.3	0.7	0.571	66.1	66%	0.848	147	0%	0.453	20.5	25	25	94.9
MO-Crk-10	0.2	0.8	0.528	66.1	66%	0.888	147	0%	0.438	25	25	25	81.7
MO-Crk-11	0.1	0.9	0.483	66.1	66%	0.888	147	0%	0.438	25	25	25	81.7
MO-Crk-12	0	1	0.438	66.1	66%	0.888	147	0%	0.438	25	25	25	81.7

*top pct – top percentile
CS – Concrete Sand
CrS – Crushed Sand
GQ – Ground Quartz

	$\rho_{cracking}$	$\chi_{cracking}$
min	50	147
max	74.4	336

For this case, finding a balanced cost and eco-efficient solution required further iterations between MO-Crk-4 and MO-Crk-6 to find the inflection point associated with the sudden changes observed (from top 3rd percentile in cost and top 52nd in eco-efficiency to top 43rd in cost and top

10th in eco-efficiency). Setting α at 0.65 and β at 0.35 results in a more balanced mixture (MO-Crk-5), which represents a top 17th percentile solution in terms of eco-efficiency while still in the top 34th percentile for cost-efficiency purposes.

As shown in this discussion, efficiency indices tied with machine-learning based diagrams (EEDDs, CEDDs and MODDs) can greatly facilitate communication between concrete producers, designers, owners and policy makers to promote efficient efforts towards sustainability without imposing unreasonable cost implications on any parts involved in a given project. The trends observed in this study suggest that, for the construction industry, thresholds on carbon footprint would preferentially be imposed on total GHG emissions associated with a given structural member type (bending, compressive, etc.) instead of a per volume basis. Additionally, the tools derived herein can provide proof of optimization to structural designers and concrete producers while giving them a reasonable margin for the optimization threshold (e.g., imposing a top 25th percentile solution in eco-efficiency). This provides enough flexibility to achieve the required mechanical performance, optimize the cost-efficiency of their mixtures and still meet environmental demands. Incentives could be added for achieving upper tier thresholds (e.g., top 10th percentile in eco-efficiency), promoting further innovation in sustainable solutions while fostering optimization efforts for currently available technologies.

There is still a major challenge related to using these diagrams as presented so far. When selecting an optimum mixture based on the density of the diagrams (which represents the magnitude of a given index) the mechanical performance is not intuitively determined. Deriving the compressive strength from the value of a given index requires the estimated volumetric impact of the corresponding mix. This iteration can become cumbersome if optimum mixtures

fail to meet performance requirements. To account for mechanical performance thresholds determined by design, these diagrams can be built while filtering the mixtures below a defined limit. For instance, if a given application requires an f_c over 100 MPa, a filtered version of Fig. 28 can be plotted as shown in Fig. 30.

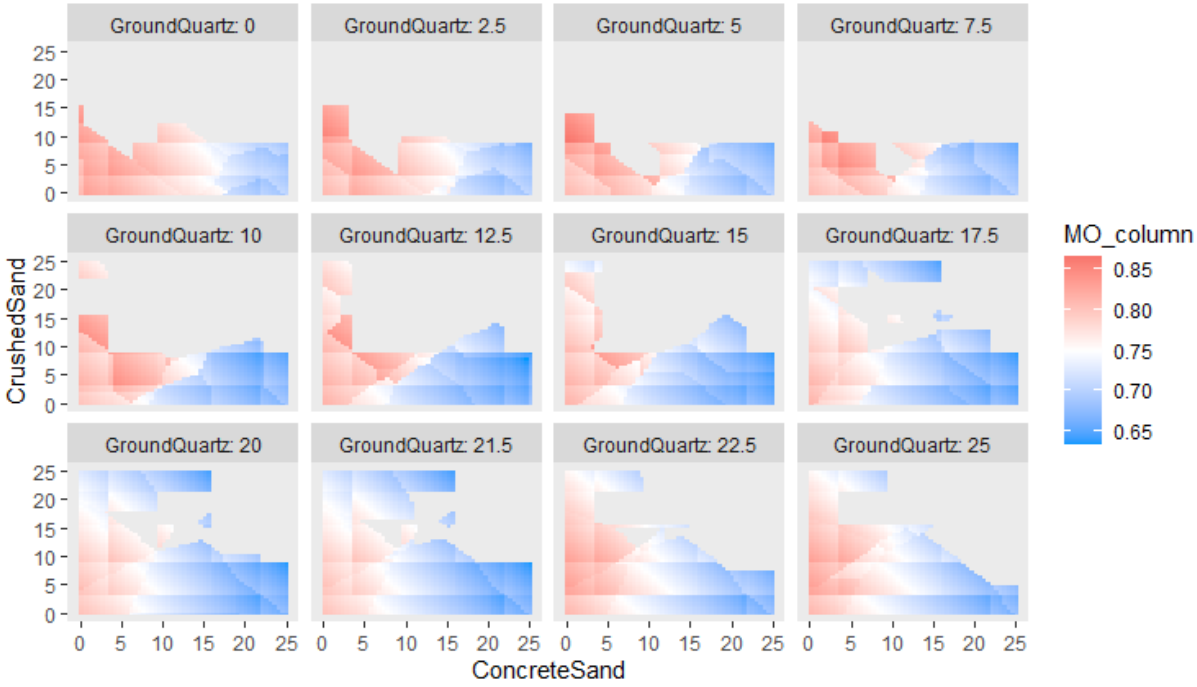


Fig. 30. MODD from Fig. 28 filtered for compressive strengths over 100 MPa

This diagram illustrates several optimum alternatives to obtain a combined cost and eco-efficient solution while still satisfying the performance demands imposed by design. Observing Fig. 30, examples of optimum mixtures for this case could be: 1) 20% concrete sand, 8% crushed sand and 10% ground quartz; or 2) 25% concrete sand, 8% crushed sand and 12.5% ground quartz; or 15% concrete sand, 25% crushed sand and 17.5% ground quartz. Similar analysis can be done for the multi-member indices developed in the previous section. APPENDIX C contains tables

illustrating the results for these indices, suggesting that optimum concrete mixtures for a given member may differ from the optimum obtained for a different application. Yet, the most efficient mixtures for $MM_{emissions}$ are also very efficient for MM_{cost} and vice-versa, in contrast to what is observed for multi-objective indices. Implementing different concrete grades in the same structure is highly dependent on the application and local regulations. To account for applications where one type of concrete must be used for an entire infrastructure, multi-member indices are ideal. From a practical standpoint, combined multi-member, multi-objective indices can be derived to evaluate the best solution for an entire project.

7.3. Comparison Between Different Concrete Technologies Using the Proposed Comparison Indices

Fig. 31 and Fig. 32 display the proposed indices, for different types of concretes, as a function of compressive strength, volumetric GWP and volumetric cost. As previously discussed, the best suited mixtures for each index/application shown in Fig. 31 and Fig. 32 are the ones that minimize these efficiency indices. The data were collected from various studies from the literature (Abdulkareem et al. 2018; Einsfeld and Velasco 2006; Lam, Wong, and Poon 1998; Miller, Horvath, and Monteiro 2016; Poon, Lam, and Wong 2000; Siddique 2004), including results from Phase B(II), and mixtures defined from Table 7 to Table 10.

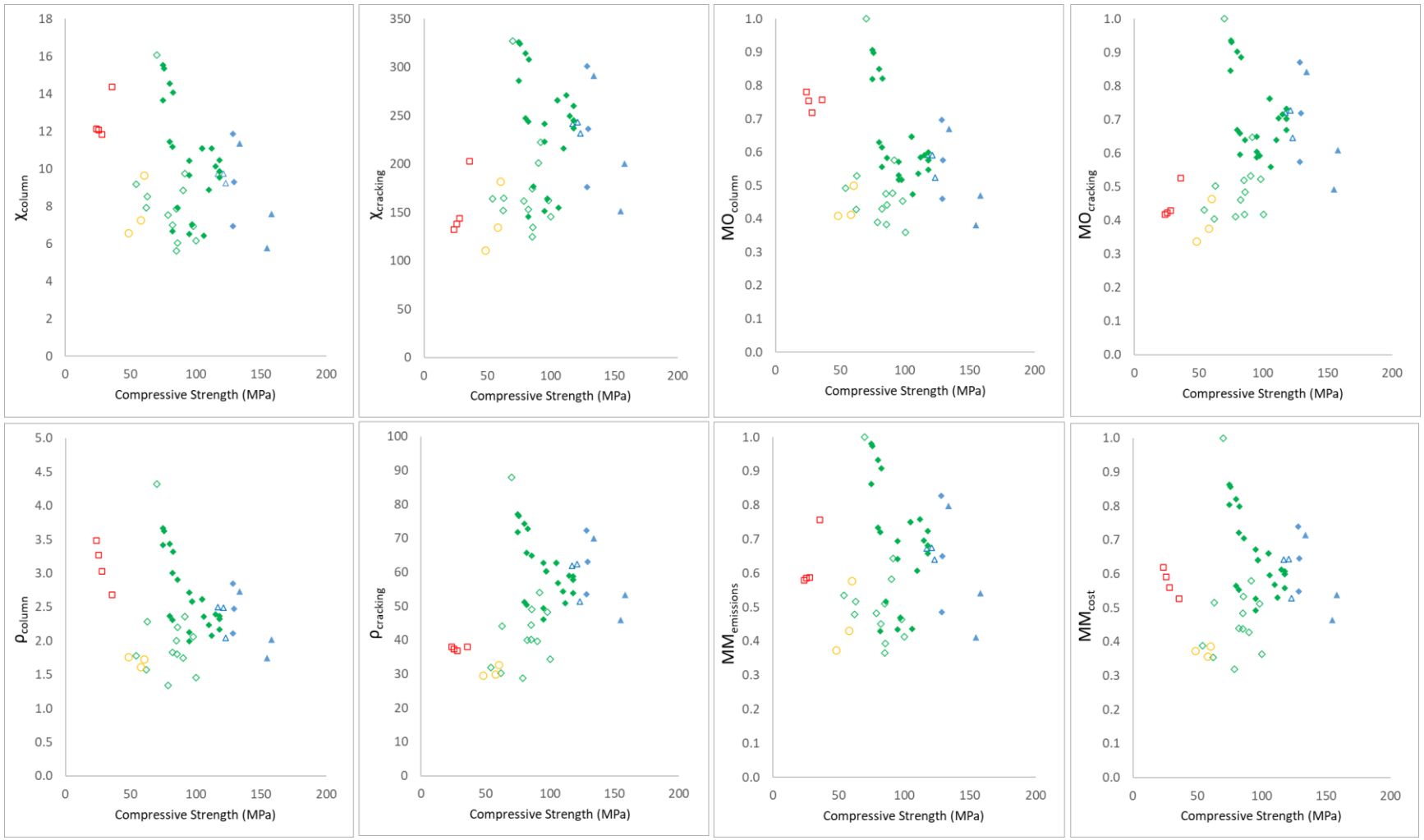


Fig. 31. Evaluation of the proposed comparison indices vs the compressive strength of different concrete technologies

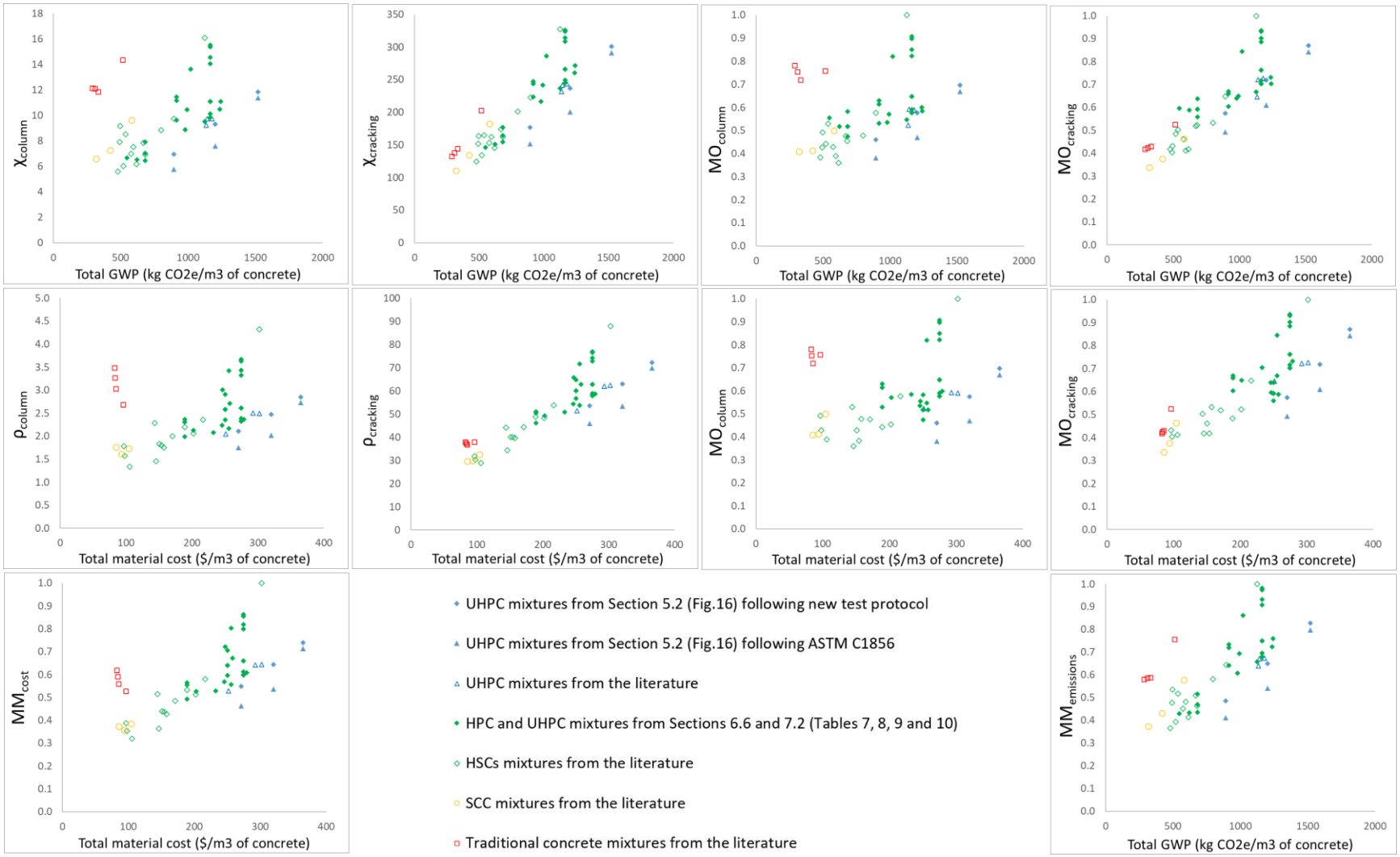


Fig. 32. Evaluation of the proposed comparison indices vs the total volumetric GWP (kg CO2-eq/m3) and cost ($/m3) associated with each mixture for different concrete technologies

Multi-objective and multi-member indices were calculated using equal weighting coefficients, that is, α and β equal to 0.5. Table C3 in APPENDIX C contains detailed information regarding source and properties from each mixture presented in Fig. 31 and Fig. 32. Mixtures from the literature were tested at ages 28 and 90 days. To allow comparison between these results and the results obtained in this study, strengths were averaged between these two testing periods to obtain an estimated 56th day strength.

Fig. 31 suggests that high paste content, high strength (and ultra-high strength) concrete technologies are not detrimental to cost or eco efficiencies and these are not a function of the type of concrete applied. For the different indices evaluated, optimum solutions can be obtained with almost all types of concrete, mostly with UHPCs, HSC and SCCs. Focusing on the UHPC mixtures developed in section 5.2 (solid blue diamonds and triangular icons) and the HPCs and UHPCs identified in this study through optimization of the proposed indices (green solid diamonds), it is evident that these technologies are between middle-of-the-pack to top solutions for all the proposed indices. For χ_{column}, the aforementioned mixtures are among the most eco-efficient solutions, along with a few HSCs from the literature (hollow green diamonds). Similar trends are observed for $\chi_{cracking}$, with the addition of NSC and SCCs as alternative top solutions for eco-efficiency. In terms of cost-efficiency, ρ_{column} shows that the most cost-efficient HPCs and UHPCs mixtures remain somewhere around the top 25th percentile of most cost-efficient solutions overall, while $\rho_{cracking}$ pushes these mixture to a middle-of-the-pack tier of ideal solutions. Meanwhile, MO_{column} and $MO_{cracking}$, suggest that the most efficient UHPC mixture

developed in section 5.2 is one of the top multi-objective solutions for short columns and a middle of the pack multi-objective solution for initial cracking applications. Likewise, $MM_{emissions}$ and MM_{cost} indicate that this mixture is one of the top multi-member solutions for eco-efficiency, while sitting in the middle of the pack for cost-efficiency.

Meanwhile, observing Fig. 32 and the plots associated with short column indices (χ_{column}, ρ_{column} and MO_{column}), it is evident that there is very little correlation between volumetric impact indicators (Total GWP and Total material cost) and these efficiency indices. Meanwhile, observing the plots associated with initial cracking indices ($\chi_{cracking}$, $\rho_{cracking}$ and $MO_{cracking}$), there is a stronger correlation between volumetric impact indicators and efficiency indices for initial cracking applications. This is mostly due to the reduced influence of compressive strength on these indicators, considering that f_c is to the power of ¼. Yet, there is a clear shift between the curves with changes in material type, which is associated with increased compressive strengths. This is due to the difference in the modulus of rupture between these different types of concrete, which was accounted for in this study using λ_{fr}. Therefore, cost and eco-efficiencies between different types of concrete should not be compared based on volumetric indicators. Evidence of this is given by observing Fig. 32 and the plots associated with initial cracking applications, where a UHPC mixture with approximately 893 kg CO_2-eq/m^3 appears to be as eco-efficient as an HSC with 500 kg CO_2-eq/m^3 or a NSC with 288 kg CO_2-eq/m^3. The same analysis can be done in terms of costs, where a UHPC mixture costing approximately \$330/m^3 is as cost-efficient as an HSC

costing around $180/m^3. Several other similar conclusions could be derived by using these scatter plots.

It is important to emphasize that the optimum UHPC mixture developed in section 5.2 was developed under several limitations in terms of materials and curing conditions, with a compressive strength value that is just inside the bottom threshold of UPHC materials (155 MPa). Meanwhile, mixtures with compressive strengths over 250 MPa have been reported with modern proprietary and non-proprietary mixtures. These were not included in this study due to the presence of different types of fibers and other fillers such as glass powder in their compositions, which require further considerations that are out of the scope of this work. The main message from these scatter plots is that, with proper optimization, advanced concrete materials such as HPCs and UHPCs can be the most eco-friendly and cost-efficient solutions for new infrastructure and superstructures, in applications where the gains in dimensionality meets constructability limitations. For instance, it may be difficult to justify these materials in small infrastructure, where the reductions in dimensions of the structural elements can only go as low as constructability permits (e.g., cross sectional area of small columns and beams as a function of required steel). On the other hand, superstructural elements such as bridge girders can greatly benefit from the outstanding mechanical properties of UHPC, especially considering the significant increase in prestressed levels, which contributes to longer spans and reduced cross sectional areas and, thus, lighter platforms. UHPC also reduces the amount of steel required, which is extremely important considering the embodied emissions associated with this material. Further reductions in required dimensionality of substructural elements can be obtained with UHPC given the longer spans attained with this material, which allows increased spacing between

supporting points, and thus, reduced number of structural elements required (e.g., columns and foundations). Such advantages are not yet captured in the analysis presented herein.

New indices that account for the impact of steel reinforced concrete on cost and eco-efficiency must be further developed to elucidate these arguments for applications involving UHPC. The path has been set with the initial efforts made by Kourehpaz and Miller (Kourehpaz and Miller 2019), where indices that account for steel in design have been developed. However, the currently available indices were developed while keeping the required area of steel constant, which inhibits accurate comparisons between NSC and UHPC solutions.

7.4. Summary of Important Outcomes

The concept of sustainable design involves environmental, economic and health, and community impacts. Knowing that decisions made during structural design and construction phases directly affect all these facets, mixture design selection should target a balanced optimization function to accommodate all three of these objectives. As governments and policy makers push towards low carbon policies, it is important for concrete producers to meet performance and sustainability requirements while maintain a cost-efficient business. Therefore, this study proposed a methodology based on multi-objective and multi-members comparison indices to develop machine-learning-based tools that demonstrate the effect of mix proportioning on performance, cost and emissions concurrently. Modifications to comparison indices developed by Kourehpaz and Miller (Kourehpaz and Miller 2019) were proposed and cost and eco-efficiency indices were combined to develop multi-objective and multi-member indices to be used as inputs in orthogonal machine-learning based tools. Key findings from this work consist of:

- MODDs developed in this study have shown to be an ideal tool for global concrete mixture optimizations, where cost-effective and eco-friendly mix designs can easily be identified while filtering out mixtures that do not meet mechanical performance requirements imposed by design. These diagrams also allow several alternative solutions to be easily identified to account for material availability and accessibility when production takes place in different locations;
- For multi-objective functions, iterations may be performed to find the most balanced solutions for cost and eco-efficiency. In this sense, MODDs can serve as proof of optimization tools to help justify the tradeoffs considered between cost and emissions when selecting a given mixture for a particular application;
- Results show that high paste content, high strength (and ultra-high strength) concrete technologies are not detrimental to cost or eco efficiencies. For the different indices evaluated, optimum solutions were mostly obtained with these types of concrete;
- Results showed that there is very little correlation between volumetric impact indicators and efficiency indices for the short column application. On the other hand, there is a stronger correlation between volumetric impact indicators and efficiency indices for initial cracking applications. This is mostly due to the reduced influence of compressive strength on cracking indices. Yet, there is still a visible shift/offset in the observed trends for different types of material, which is associated with increased compressive strengths and increased differences in the modulus of rupture between these different concretes. Therefore, cost and eco-efficiency comparisons between different types of concrete should not be performed on a volumetric impact basis.

Future work should focus on developing new indices that account for the environmental impact of reinforcing steel on eco-efficiency and further development of machine-learning based density diagrams for material properties other than compressive strength.

As a contribution to concrete producers, project owners, practicing engineers, designers and regulating entities, MODDs could serve as decision-making aids during mix design stages and provide proof of mixture optimization that can be introduced alongside Environmental Product Declarations (EPD).

7.5. References

Abdulkareem, Omar M., Amor Ben Fraj, Marwen Bouasker, and Abdelhafid Khelidj. 2018. 'Mixture design and early age investigations of more sustainable UHPC', *Construction and Building Materials*, 163: 235-46.

Athan, Timothy Ward, and Panos Y. Papalambros. 1996. 'A NOTE ON WEIGHTED CRITERIA METHODS FOR COMPROMISE SOLUTIONS IN MULTI-OBJECTIVE OPTIMIZATION', *Engineering Optimization*, 27: 155-76.

Bridgman, Percy Williams. 1922. *Dimensional analysis* (Yale university press).

Einsfeld, Ricardo A., and Marta S. L. Velasco. 2006. 'Fracture parameters for high-performance concrete', *Cement and Concrete Research*, 36: 576-83.

Haimes, Yacov Y., and Warren A. Hall. 1974. 'Multiobjectives in water resource systems analysis: The Surrogate Worth Trade Off Method', *Water Resources Research*, 10: 615-24.

Ju-Long, Deng. 1982. 'Control problems of grey systems', *Systems & Control Letters*, 1: 288-94.

Kaliszewski, Ignacy. 1987. 'A modified weighted tchebycheff metric for multiple objective programming', *Computers & Operations Research*, 14: 315-23.

Koski, Juhani. 1981. "Multicriterion optimization in structural design." In.: TAMPERE UNIV OF TECHNOLOGY (FINLAND).

Koski, Juhani, and Risto Silvennoinen. 1987. 'Norm methods and partial weighting in multicriterion optimization of structures', *International Journal for Numerical Methods in Engineering*, 24: 1101-21.

Kourehpaz, Pouria, and Sabbie A. Miller. 2019. 'Eco-efficient design indices for reinforced concrete members', *Materials and Structures*, 52: 96.

Lam, L., Y. L. Wong, and C. S. Poon. 1998. 'Effect of Fly Ash and Silica Fume on Compressive and Fracture Behaviors of Concrete', *Cement and Concrete Research*, 28: 271-83.

Laponce, J. A. 1972. 'Vilfredo Pareto, Manual of Political Economy. Trans. Ann S. Schwier, ed. Ann S. Schwier and Alfred N. Page. New York: Augustus M. Kelley, 1971, pp. xii, 504', *Canadian Journal of Political Science*, 5: 599-600.

Marler, RT, and JS Arora. 2003. 'Review of multi-objective optimization concepts and methods for engineering', *Iowa City, IA: University of Iowa, Optimal Design Laboratory*.

Miller, Sabbie A, Arpad Horvath, and Paulo JM Monteiro. 2016. 'Readily implementable techniques can cut annual CO2 emissions from the production of concrete by over 20%', *Environmental Research Letters*, 11: 074029.

Nash, John F. 1950. 'The Bargaining Problem', *Econometrica*, 18: 155-62.

Osyczka, Andrzej. 1978. 'An approach to multicriterion optimization problems for engineering design', *Computer Methods in Applied Mechanics and Engineering*, 15: 309-33.

Poon, C. S., L. Lam, and Y. L. Wong. 2000. 'A study on high strength concrete prepared with large volumes of low calcium fly ash', *Cement and Concrete Research*, 30: 447-55.

Proos, KA, GP Steven, OM Querin, and YM Xie. 2001. 'Multicriterion evolutionary structural optimization using the weighting and the global criterion methods', *AIAA journal*, 39: 2006-12.

Rao, S. S. 1987. 'Game theory approach for multiobjective structural optimization', *Computers & Structures*, 25: 119-27.

Siddique, Rafat. 2004. 'Performance characteristics of high-volume Class F fly ash concrete', *Cement and Concrete Research*, 34: 487-93.

CHAPTER VIII.

CONCLUSIONS & FUTURE RESEARCH

The emergence of UHPC as an attractive solution for precast and prestressed applications has coincided with global efforts towards sustainable construction. The increasing need for tools capable of intuitively demonstrating the effect of concrete mixture composition on mechanical performance, cost and eco-efficiency concurrently has motivated this work in an effort to promote design of more sustainable solutions to help meet environmental goals. In addition, this study was also motivated by the several challenges related to the use of AI models in predicting, describing and displaying concrete material's performance, as well as the resource and time-consuming nature of available standard methods to characterize certain material properties. The main objective consisted of developing new tools that facilitate evaluation of the tradeoffs between mechanical performance, cost and environmental impact simultaneously, as a function of mix proportioning when designing UHPC materials. Comparison indices proposed by Kourehpaz and Miller (Kourehpaz and Miller 2019) were modified to develop eco-efficiency, cost-efficiency and multi-objective indices, while compressive strengths were predicted with machine-learning models and used to calculate these indices. Finally, the calculated indices were used as outputs in orthogonal machine-learning-based tools. Key findings from this work consist of:

- The new test protocol developed for pastes and mortars using reduced size samples and simplified end specimen conditions agrees well with standard methods up to strengths of 125 MPa, promoting an expeditious and efficient assessment of UHPC binders and mortars that greatly facilitates data collection and advanced modeling of new

formulations. Further evaluation of the end-specimen conditions is required to improve this test protocol for materials with strengths over 125 MPa.

- The experimental framework used in this study, supported by orthogonal arrays and the use of surrogate samples has shown to be an effective method for rapidly generating data to support AI algorithms for concrete mixture design optimization. This method can be used to overcome the uncertainty of models generated with large, multiple-source datasets by promoting reduced experimental runs capable of effectively describing the experimental domain. Considering the high variability of material properties with source and region and the often prohibitive costs of nanomaterials, this protocol encourages innovative mixture designs with new materials in a constantly growing industry such as UHPC.
- PDDs developed in this study suggest that performance evaluation of regression models should not rely solely on RMSE. A potential data leakage occurrence was detected in the diagram, where inconsistent trends in the domain where observed. The balanced RMSEs obtained across training and testing sets are most probably consequence of the data leakage phenomena, where the same information is present in both training and testing sets.
- A PDD developed with an ensemble model enabled the design of an UHPC mixture averaging a compressive strength of 155 MPa (22,480 psi) at age of 56 days, while maintaining the fine-aggregate-to-cementitious ratio above unity (fa/cm = 1.04).
- The EEDDs, CEDDs and MODDs developed in this study have shown to be ideal tools for concrete mixture optimization, where cost-effective and eco-friendly mix designs can

easily be determined while filtering out mixtures that do not meet mechanical performance requirements imposed by design. These diagrams also facilitate the identification of several alternative solutions to account for material availability and accessibility when production takes place in different locations;

- Optimum mixtures for cost and eco-efficiency can significantly differ from each other. Similarly, optimum mixtures, in terms of cost and eco-efficiency, for different structural members can significantly differ from each other;
- Due to differences in mechanical performance, mixtures with high volumetric environmental impact can still be more sustainable than mixtures with nearly half the volumetric environmental impact. The same trend was observed in terms of cost-efficiency. For policy makers, the implication is that simply targeting concrete mixtures with the lowest cost or lowest embodied emissions on a per volume basis does not, generally, result in the lowest cost or lowest embodied emissions for a given structural element or infrastructure project;
- For multi-objective functions, iterations may be performed to find the most balanced solutions for cost- and eco-efficiency. In this sense, MODDs can serve as proof of optimization tools to help justify the tradeoffs considered between cost and emissions when selecting a given mixture for a particular application;
- Mix proportioning optimization should be conducted considering costs, emissions and mechanical performance concurrently. Focusing all efforts on identifying a mixture with the best mechanical performance alone may result in less efficient solutions, while similar

mechanical performance may be obtained with much more cost and eco-effective alternatives;

- Results show that high paste content, high strength (and ultra-high strength) concrete technologies are not detrimental to cost or eco efficiencies. For the different indices evaluated, optimum solutions were mostly obtained with these types of concrete;
- Results showed that there is very little correlation between volumetric impact indicators and efficiency indices for the short column application. On the other hand, there is a stronger correlation between volumetric impact indicators and efficiency indices for initial cracking applications. This is mostly due to the reduced influence of compressive strength on cracking indices. Yet, there is still a visible shift/offset in the observed trends for different types of material, which is associated with increased compressive strengths and increased differences in the modulus of rupture between these different concretes. Therefore, cost and eco-efficiency comparisons between different types of concrete should not be performed on a volumetric impact basis.

The methodology used in this study represents a significant opportunity for potential game-changing design protocols, capable of offering enough flexibility for exploring new constituents and account for differences in source, availability and cost of material per region. This is particularly important for advanced concrete materials such as UHPC, for which mixture optimization is key to justify its application, considering its high cost per unit volume along with its volumetric environmental impact. PDDs, CEDDs, EEDDs and MODDs can be used to evaluate mechanical and durability properties directly affecting cost and environmental impact and, combined with 3D density plots, they not only serve as decision-making aid tools to be used

during mix design stages, but also help overcome the challenges mentioned throughout this manuscript. In particular, these tools provided further insight on the inner workings of machine learning models by displaying their predictive structure, unveiling some of their black-box nature and helping identify overfitting and data leakage occurrences where the RMSE was not sufficiently informative.

As a contribution to practicing engineers, the tools developed herein can facilitate the decision making process surrounding material availability and cost-performance-sustainability trade-offs, promoting better design practices towards stronger, long-lasting and greener concrete products. Once properly developed by experienced engineers in the field of civil materials and machine learning, these diagrams can easily be used and interpreted by industry personnel on a regular basis for their day-to-day operations and decisions. They can also facilitate the communication between projects owners, regulating entities and designers, which ultimately will contribute to lift the existing mis-conceptional barriers regarding advanced concrete materials such as UHPC, and promote its application in cases where this material is clearly a better option over other traditional concrete technologies.

Future work is suggested as follows:

- Focus should be placed on developing new indices that: 1) account for the environmental impact of reinforcing steel on eco-efficiency of members/elements associated with typical UHPC applications; 2) account for the buckling effect in columns considering the potential application of UHPC in slender columns (Aboukifa et al. 2019; Hung, Hu, and Yen 2018); 3) account for the effect of increased prestress levels on the total span achieved in

bridge elements (beams, girders), which dictates the distance between structural supports and total weight of the superstructure, impacting the volume of required substructural elements (columns, footing, piles); and 4) account for life-cycle costs (maintenance and repair) and durability (service life) of structural elements;

- Machine-learning-based tools to optimize material properties other than compressive strength should be explored. This methodology has a great potential on fields such as fiber reinforced concretes (including UHPC), where simultaneous changes in the matrix proportioning and fiber content greatly impacts strain-hardening behavior;
- New models should be developed using inputs related to individual particle make-up (fineness, characteristic particle size and compound composition of the raw ingredients). These have the potential to overcome the variability in raw materials with source and region;
- Inclusion of nanomaterials and fibers should also be evaluated through this method and tested through standardized test protocols to develop eco-friendly and cost-efficient concretes with much higher strengths than the ones obtained with the limited resources in this study;
- Using large, trustworthy datasets from the literature, generated through standardized test methods, are also of great important to develop informative PDDs for material properties that are time- and resource-exhausting, and/or hard to measure. Making these diagrams available in technotes or state-of-the-art documents can greatly contribute to improve material performance and quality across the concrete industry;

- Developing software capable of automating the process of generating these diagrams while still providing enough flexibility to account for customized data and experimental campaigns can have a significant impact on mix design processes and on the concrete industry in general;
- Efforts should be made by entities representing each concrete specialty/application (committees, associations, etc.) to draft specifications for engineers/architects prescribing how to use these tools. Multi-objective comparison indices should be described and explained in these specifications.

8.1. References

Aboukifa, Mahmoud, Mohamed A. Moustafa, Ahmad M. Itani, and Negar Naeimi. 2019. 'Durable UHPC Columns with High-Strength Steel'.

Hung, Chung-Chan, Fuo-Yao Hu, and Cheng-Hao Yen. 2018. 'Behavior of slender UHPC columns under eccentric loading', *Engineering Structures*, 174: 701-11.

Kourehpaz, Pouria, and Sabbie A. Miller. 2019. 'Eco-efficient design indices for reinforced concrete members', *Materials and Structures*, 52: 96.

APPENDIX A

The following figures and tables were previously discussed and provide relevant information to this work.

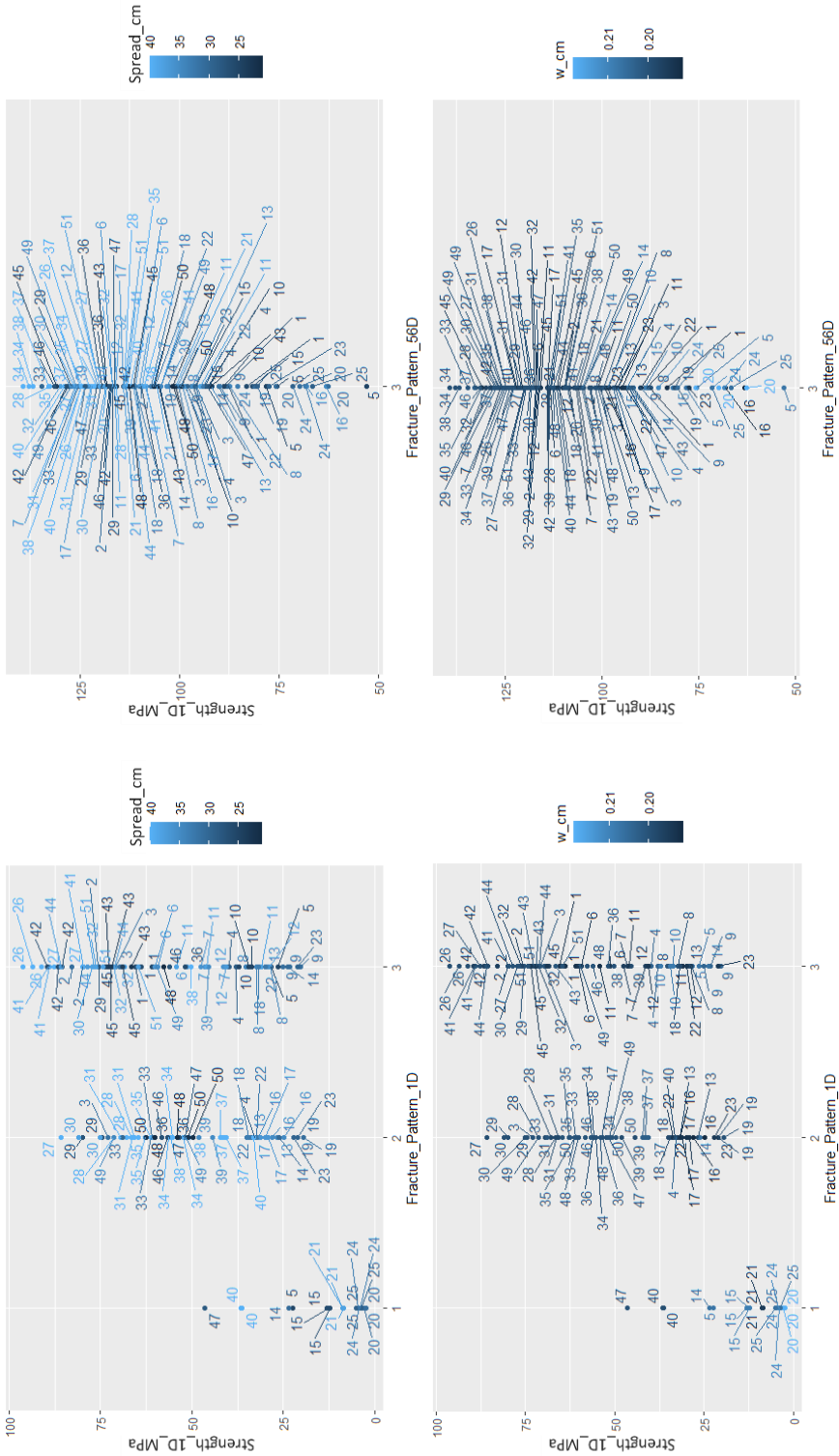


Fig. A1. Comparison between the fracture pattern vs the compressive strength measured for each specimen at ages 1 and 56 days as a function of the measured slump and w/cm ratio. The numbers in the plot represent the corresponding mix design.

Table A1. L25 orthogonal array followed in Phase A(I). Variables: slag, microsilica and fly ash

Mix Design #	Design levels		
	Slag	Microsilica	Fly Ash
Mix A-1	1	1	1
Mix A-2	1	2	2
Mix A-3	1	3	3
Mix A-4	1	4	4
Mix A-5	1	5	5
Mix A-6	2	1	2
Mix A-7	2	2	3
Mix A-8	2	3	4
Mix A-9	2	4	5
Mix A-10	2	5	1
Mix A-11	3	1	3
Mix A-12	3	2	4
Mix A-13	3	3	5
Mix A-14	3	4	1
Mix A-15	3	5	2
Mix A-16	4	1	4
Mix A-17	4	2	5
Mix A-18	4	3	1
Mix A-19	4	4	2
Mix A-20	4	5	3
Mix A-21	5	1	5
Mix A-22	5	2	1
Mix A-23	5	3	2
Mix A-24	5	4	3
Mix A-25	5	5	4

Table A4. L25 orthogonal array followed in Phase A(II). Variables: slag, microsilica and fly ash

Mix Design #	Design levels		
	Slag	Microsilica	Fly Ash
-	1	1	1
Mix A-26	1	2	2
Mix A-27	1	3	3
Mix A-28	1	4	4
Mix A-29	2	1	2
Mix A-30	2	2	3
Mix A-31	2	3	4
Mix A-32	2	4	1
Mix A-33	3	1	3
Mix A-34	3	2	4
Mix A-35	3	3	1
Mix A-36	3	4	2
Mix A-37	4	1	4
Mix A-38	4	2	1
Mix A-39	4	3	2
Mix A-40	4	4	3

Table A2. Design levels defined for Phase A(II)

Features	Levels			
	1	2	3	4
Slag	0	17.3	34.7	52
Microsilica	0	3.5	7	10.5
FA	0	3.75	7.5	15

Table A3. L25 orthogonal array followed in Phase B. Variables: ground quartz, concrete sand and crushed sand

Mix Design #	Variables		
	Ground Quartz	Concrete Sand	Crushed Sand
Mix B-1	1	1	1
Mix B-2	1	2	2
Mix B-3	1	3	3
Mix B-4	1	4	4
Mix B-5	1	5	5
Mix B-6	2	1	2
Mix B-7	2	2	3
Mix B-8	2	3	4
Mix B-9	2	4	5
Mix B-10	2	5	1
Mix B-11	3	1	3
Mix B-12	3	2	4
Mix B-13	3	3	5
Mix B-14	3	4	1
Mix B-15	3	5	2
Mix B-16	4	1	4
Mix B-17	4	2	5
Mix B-18	4	3	1
Mix B-19	4	4	2
Mix B-20	4	5	3
Mix B-21	5	1	5
Mix B-22	5	2	1
Mix B-23	5	3	2
Mix B-24	5	4	3
Mix B-25	5	5	4

Table A5. Design levels defined for Phase B

Features	Levels				
	1	2	3	4	5
Ground Quartz	0	6.25	12.5	18.75	25
Silica Sand	0	6.25	12.5	18.75	25
Crushed Sand	0	6.25	12.5	18.75	25

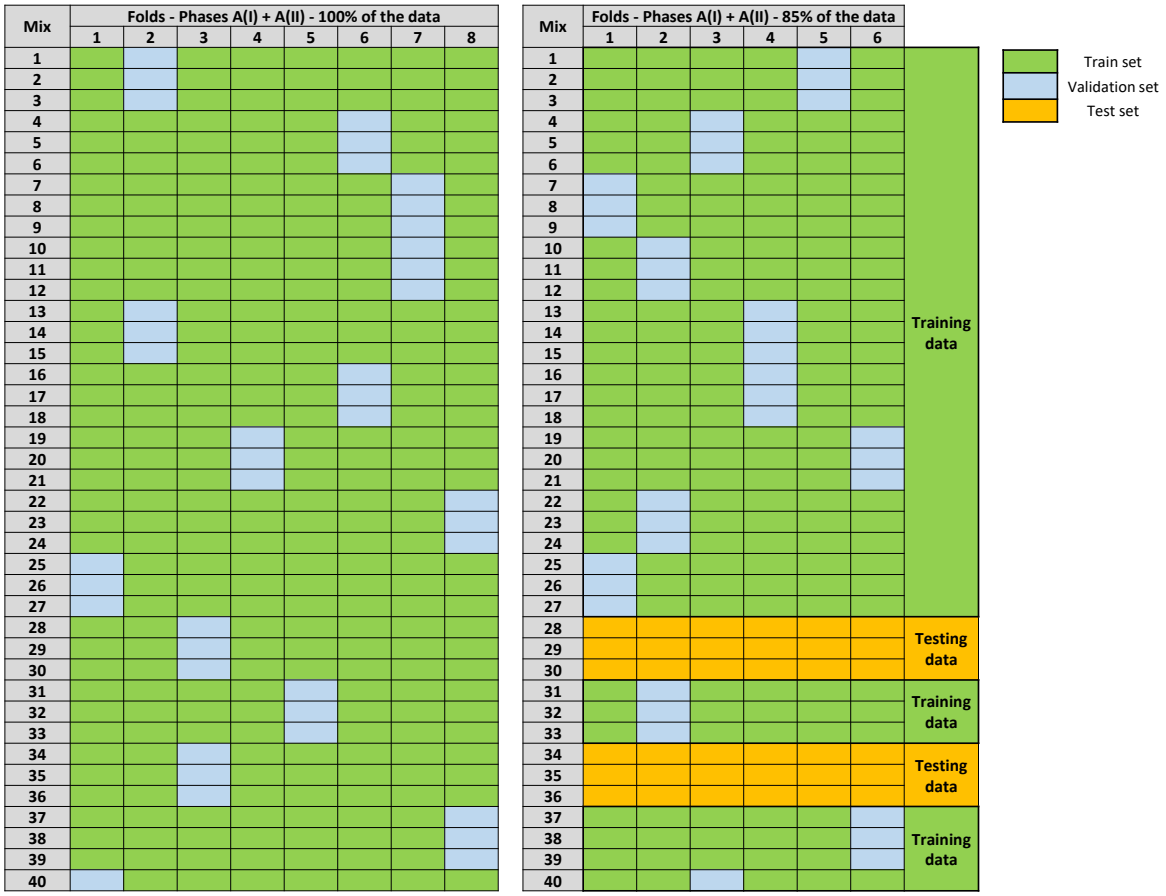


Fig. A2. Folds randomly created in performing a k-fold cross validation method used to train and test the models. Eight folds were created in the entire dataset combining Phases A(I) and A(II). Mixtures that were "blind" to Fold 3 were stored as test sets. Six new folds were created within the dataset available for training (training and validation sets) to optimize the tuning parameters

Table A6. Variable importance and variability explained of each feature in each model using the 56 days results

		Models without w/cm			Models with w/cm			
		Variable importance			Variable importance			
Models	# obs	Slag	Microsilica	Fly Ash	w/cm	Slag	Microsilica	Fly Ash
kNN_avg_84_Phase A(I)	21	12.1	100.0	0.0	89.4	12.1	100.0	0.0
kNN_avg_100_Phase A(I)	25	24.5	100.0	0.0	100.0	20.7	84.5	0.0
kNN_avg_80_Phase A(I)+A(II)	34	19.8	100.0	0.0	100.0	19.8	100.0	0.0
kNN_avg_80_Phase A(I)+A(II)_2.0	34	21.8	100.0	0.0	100.0	19.3	88.4	0.0
kNN_avg_100_Phase A(I)+A(II)	40	14.9	100.0	0.0	100.0	13.4	89.4	0.0
kNN_sp_84_Phase A(I)	63	0.2	100.0	0.0	100.0	0.1	45.4	0.0
kNN_sp_100_Phase A(I)	75	4.5	100.0	0.0	100.0	2.5	54.9	0.0
kNN_sp_80_Phase A(I)+A(II)_2.0	102	0.0	100.0	0.2	100.0	0.0	66.5	0.1
kNN_sp_100_Phase A(I)+A(II)	120	3.5	100.0	0.0	100.0	2.1	61.2	0.0
RF_avg_84_Phase A(I)	21	3.5	100.0	0.0	99.1	7.2	100.0	0.0
RF_avg_100_Phase A(I)	25	22.2	100.0	0.0	100.0	0.0	55.7	2.3
RF_avg_80_Phase A(I)+A(II)	34	23.6	100.0	0.0	93.7	20.5	100.0	0.0
RF_avg_80_Phase A(I)+A(II)_2.0	34	29.8	100.0	0.0	100.0	27.1	89.4	0.0
RF_avg_100_Phase A(I)+A(II)	40	22.5	100.0	0.0	100.0	16.5	94.0	0.0
RF_sp_84_Phase A(I)	63	3.3	100.0	0.0	100.0	0.0	29.3	5.8
RF_sp_100_Phase A(I)	75	0.0	100.0	14.7	100.0	0.0	61.5	5.2
RF_sp_80_Phase A(I)+A(II)_2.0	102	18.1	100.0	0.0	100.0	12.8	75.2	0.0
RF_sp_100_Phase A(I)+A(II)	120	20.5	100.0	0.0	100.0	14.8	83.8	0.0

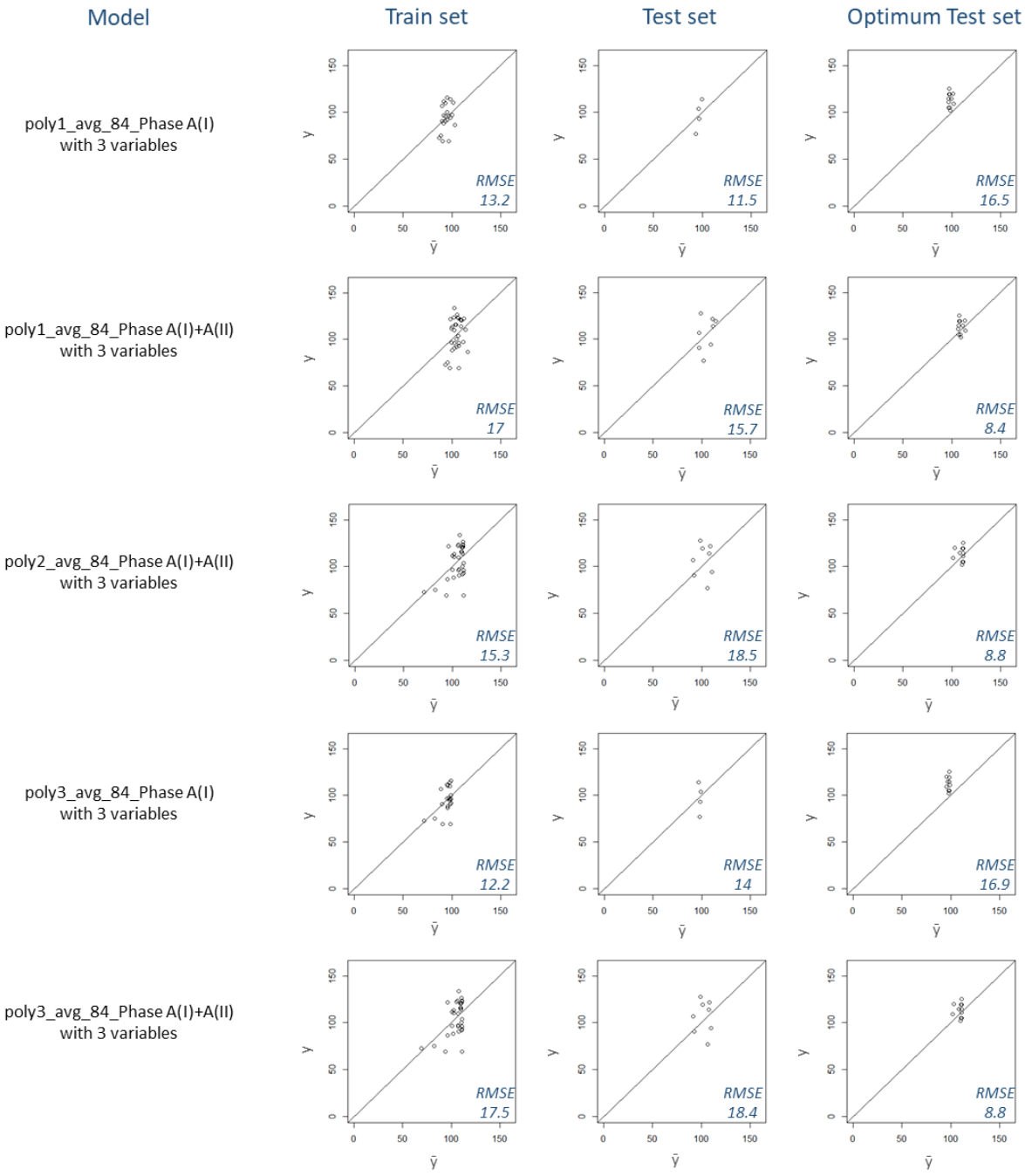


Fig. A3. Predicted compressive strengths (ȳ) versus actual outcomes (y), in MPa, and RMSEs for the best performing linear models in this study when predicting the compressive strength of mixtures cast in Phase A and tested at age 56 days. These models were built using only SCMs as the features.

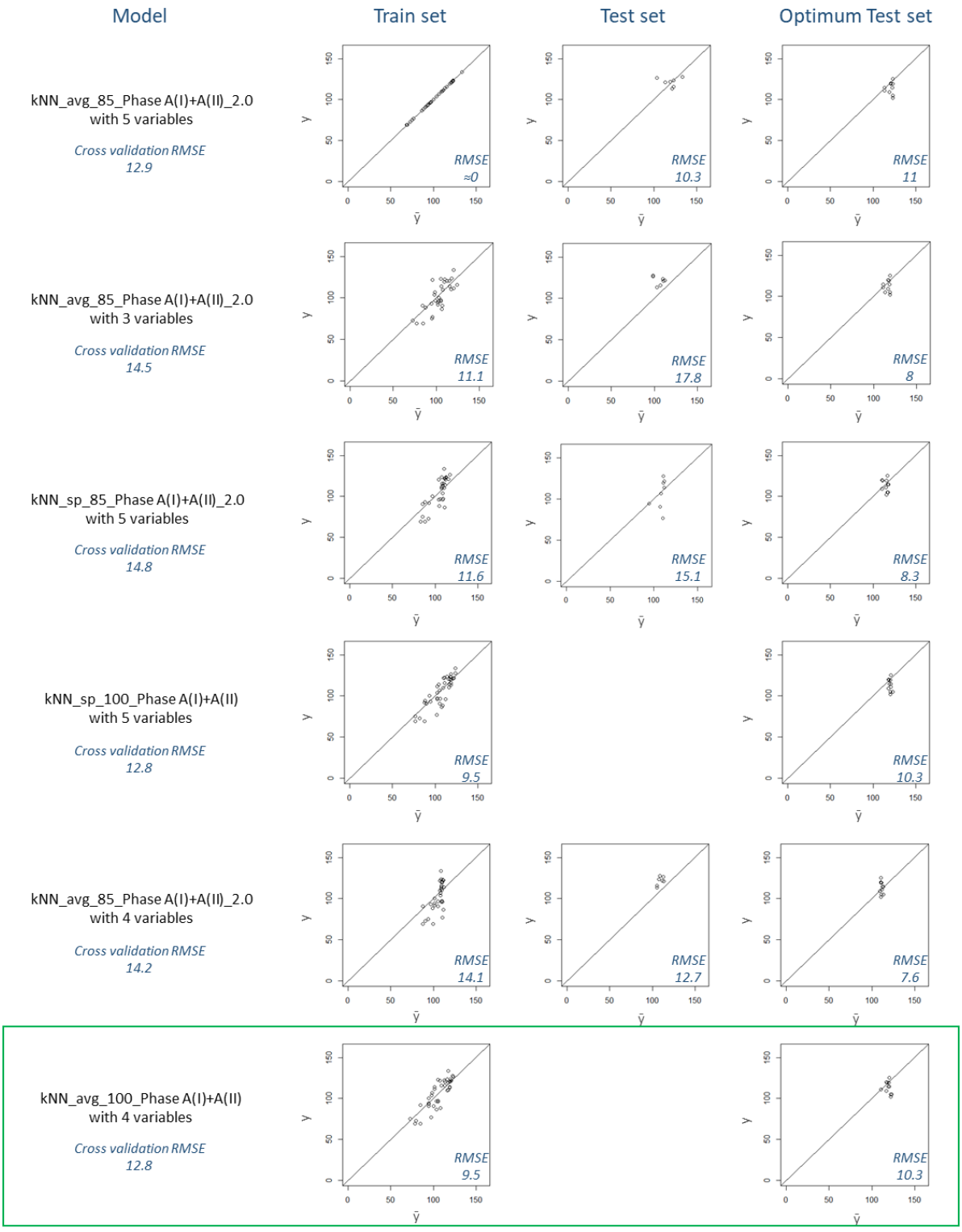


Fig. A4. Predicted compressive strengths (ȳ) versus actual outcomes (y), in MPa, and RMSEs for the best performing kNN models in this study when predicting the compressive strength of mixtures cast in Phase A and tested at age 56 days. These models were built considering three scenarios: 1) using only SCMs as the features; 2) adding w/cm as the 4th variable; 3) adding w/cm and HRWR/cm as the 4th and 5th variable, respectively

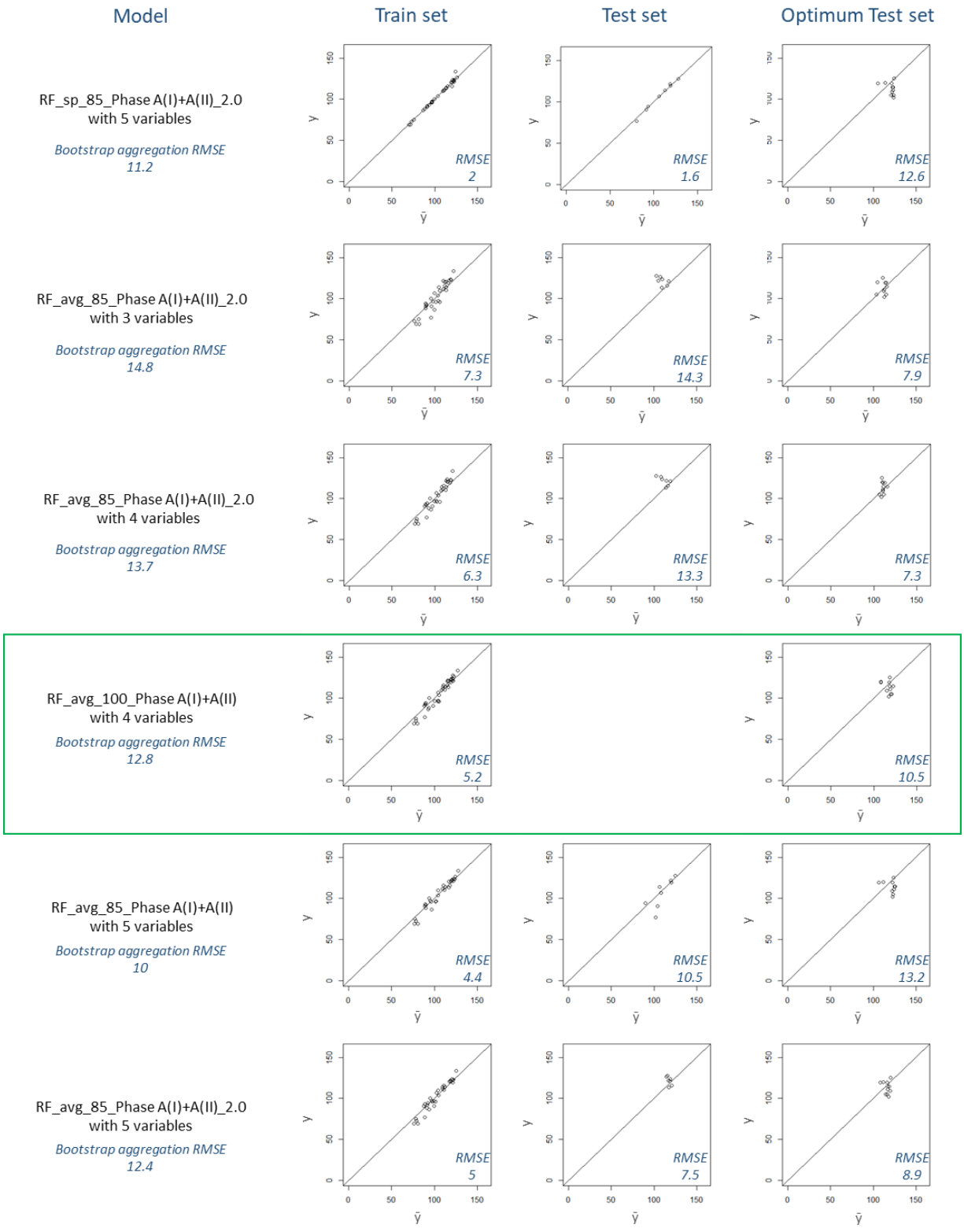


Fig. A5. Predicted compressive strengths (ȳ) versus actual outcomes (y), in MPa, and RMSEs for the best performing random forest models in this study when predicting the compressive strength of mixtures cast in Phase A and tested at age 56 days. These models were built considering three scenarios: 1) using only SCMs as the features; 2) adding w/cm as the 4th variable; 3) adding w/cm and HRWR/cm as the 4th and 5th variable, respectively

APPENDIX B

Models were developed for Phase B in a similar manner to what was discussed in subsection 4.2.1 for Phase A. In particular, models were developed for six scenarios: 1) excluding mix #25 from the dataset (due to very high HRWR content); 2) Using three fine aggregates (ground quartz, concrete sand and crushed sand) as the variables; 3) Using the three fine aggregates and the w/cm as the variables; 4) Using the three fine aggregates, the w/cm and the HRWR/cm as the variables; 5) Using the three fine aggregates and the cementitious content as the variables; and 6) Using the three fine aggregates, the w/cm and the cementitious content as the variables. Table B1 show the RMSE obtained following these approaches. The final models selected to be ensembled were models *"kNN_avg_80_Phase B"* and *"RF_avg_80_Phase B"*. Fig. B1 illustrates the correlation between models' prediction versus actual experimental outcomes, along with the corresponding RMSE obtained in the training and testing sets.

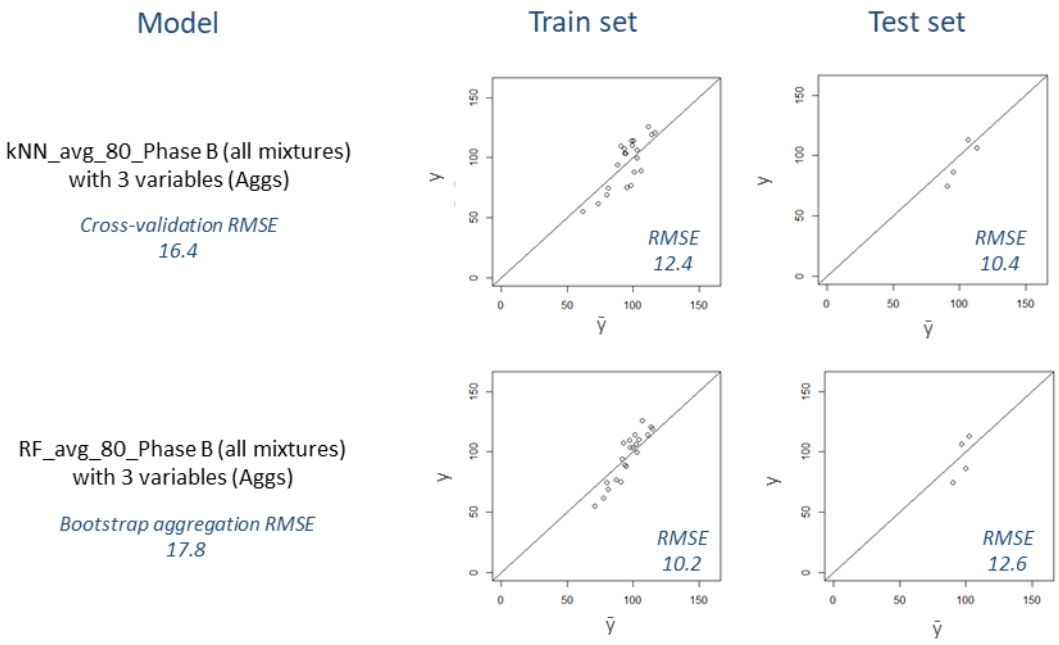


Fig. B1. Predicted compressive strengths (ȳ) versus actual outcomes (y), in MPa, and RMSEs for the best performing kNN and RF models, used to develop PDDs for Phase B. These models were built using three fine aggregates (ground quartz, concrete sand and crushed sand) as the variables.

These models were selected due to the balanced RMSE obtained across training and testing sets. Predictive performance could have been greatly improved with an additional iteration, similar to what was done with Phase A(II), or with higher design levels in the orthogonal array. Yet, time and resource limitations prevented further experimental campaigns for this research. Nevertheless, Fig. B1 indicates that the trends existing in the domain are well captured by these models, satisfying their purpose for optimization tools.

Table B1. RMSE obtained for each model when predicting the compressive strength of mixtures cast in Phase B and tested at age 56 days. These models were built considering six scenarios in which different variables are evaluated (as described in Appendix B)

	#obs		Models with 3 variables				Models with 4 variables				Models with 5 variables			
				train set		test set		train set		test set		train set		test set
Models	train set	test set	tp	RMSE$^{(cv\|ba)}$	RMSE	RMSE	tp	RMSE$^{(cv\|ba)}$	RMSE	RMSE	tp	RMSE$^{(cv\|ba)}$	RMSE	RMSE
				Var = 3 Aggs (filtering out mix 25)				Var = (3 Aggs + cm content)				Var = (3 Aggs + cm content + w/cm)		
kNN_avg_84_Phase B	20	4	7	16.3	16	22.1	3	16.1	12.3	10.4	4	15.9	13.8	11.3
kNN_avg_100_Phase B	25	-	3	15.4	12.9	-	3	14.7	10.7	-	3	15.3	11.6	-
kNN_sp_84_Phase B	60	4	30	17.8	15.6	16.6	18	16.4	15	8.6	15	16.9	15.4	7.7
kNN_sp_100_Phase B	75	-	7	14.5	12.9	-	9	13.8	10.7	-	12	14.8	11.7	-
RF_avg_84_Phase B	20	4	2	16.4	8.7	17.1	3	16	7.3	11.1	3	15.9	7.2	12.8
RF_avg_100_Phase B	25	-	2	16.6	8.3	-	2	14.9	7.8	-	2	16.8	7.6	-
				Var = 3 Aggs (fitting all mixtures)				Var = (3 Aggs + w/cm)				Var = (3 Aggs + w/cm + hrwr/cm)		
kNN_avg_84_Phase B	21	3	3	16.4	12.4	10.4	5	18.1	16.4	11.1	3	16.9	12.6	11.9
kNN_avg_100_Phase B	25	-	3	15.2	11.4	-	5	17.1	15	-	3	15.4	11	-
kNN_sp_84_Phase B	63	3	30	17.8	17.6	10.4	27	17.6	18.6	11	18	18.2	16.1	8.4
kNN_sp_100_Phase B	75	-	9	14.1	11.4	-	6	14.9	10.2	-	12	14.6	11.8	-
RF_avg_84_Phase B	21	3	1	17.8	10.2	12.6	3	17.5	7.8	13.9	3	16.1	6.9	15.2
RF_avg_100_Phase B	25	-	3	16.5	7.7	-	2	17.4	7.9	-	3	15.9	7.1	-

tp - tuning parameter ("k" for kNN; "mtry" for RF)
#obs - number of observations
opt test set - optimum test set
RMSE$^{(cv)}$ - root mean squared error of the model estimated during cross validation (kNN models)
RMSE$^{(ba)}$ - root mean squared error of the model estimated during bootstrap aggregation (RF models)
Aggs - aggregates (ground quartz, concrete sand, crushed sand)
cm - cementitious
Var - variables

APPENDIX C

Table C1. Optimum MM emissions for varying values allocated to the weighting coefficients α and β

Mix #	α	β	$MM_{emissions}$	Optimized cost χ_{column}	*top pct	$\overline{\chi}_{column}$	Optimized GWP $\chi_{cracking}$	*top pct	$\overline{\chi}_{cracking}$	CS	CrS	GQ	f_c (MPa)
MM-GWP-1	1	0	0.431	6.58	0%	0.431	152	3%	0.453	20.5	25	25	94.9
MM-GWP-2	0.9	0.1	0.431	6.58	0%	0.431	152	3%	0.453	20.5	25	25	94.9
MM-GWP-3	0.8	0.2	0.431	6.58	0%	0.431	152	3%	0.453	20.5	25	25	94.9
MM-GWP-4	0.7	0.3	0.431	6.58	0%	0.431	152	3%	0.453	20.5	25	25	94.9
MM-GWP-5	0.6	0.4	0.431	6.58	0%	0.431	152	3%	0.453	20.5	25	25	94.9
MM-GWP-6	0.5	0.5	0.44	6.75	2%	0.442	147	0%	0.442	25	25	25	81.7
MM-GWP-7	0.4	0.6	0.44	6.75	2%	0.442	147	0%	0.442	25	25	25	81.7
MM-GWP-8	0.3	0.7	0.44	6.75	2%	0.442	147	0%	0.442	25	25	25	81.7
MM-GWP-9	0.2	0.8	0.44	6.75	2%	0.442	147	0%	0.442	25	25	25	81.7
MM-GWP-10	0.1	0.9	0.44	6.75	2%	0.442	147	0%	0.442	25	25	25	81.7
MM-GWP-11	0	1	0.44	6.75	2%	0.442	147	0%	0.442	25	25	25	81.7

*top pct - top percentile
CS – Concrete Sand
CrS – Crushed Sand
GQ – Ground Quartz

	χ_{column}	$\chi_{cracking}$
min	6.58	147
max	15.3	336

Table C2. Optimum MM_cost for varying values allocated to the weighting coefficients α and β

Mix #	α	β	MM_{cost}	Optimized cost ρ_{column}	*top pct	$\overline{\rho}_{column}$	Optimized GWP $\rho_{cracking}$	*top pct	$\overline{\rho}_{cracking}$	CS	CrS	GQ	f_c (MPa)
MM-Cost-1	1	0	0.632	2.08	0%	0.632	50.9	4%	0.684	21.5	2.5	0	112
MM-Cost-2	0.9	0.1	0.632	2.08	0%	0.632	50.9	4%	0.684	21.5	2.5	0	112
MM-Cost-3	0.8	0.2	0.632	2.08	0%	0.632	50.9	4%	0.684	21.5	2.5	0	112
MM-Cost-4	0.7	0.3	0.643	2.09	1%	0.635	50	0%	0.673	25	7	0	105
MM-Cost-5	0.6	0.4	0.643	2.09	1%	0.635	50	0%	0.673	25	7	0	105
MM-Cost-6	0.5	0.5	0.643	2.09	1%	0.635	50	0%	0.673	25	7	0	105
MM-Cost-7	0.4	0.6	0.643	2.09	1%	0.635	50	0%	0.673	25	7	0	105
MM-Cost-8	0.3	0.7	0.643	2.09	1%	0.635	50	0%	0.673	25	7	0	105
MM-Cost-9	0.2	0.8	0.643	2.09	1%	0.635	50	0%	0.673	25	7	0	105
MM-Cost-10	0.1	0.9	0.643	2.09	1%	0.635	50	0%	0.673	25	7	0	105
MM-Cost-11	0	1	0.643	2.09	1%	0.635	50	0%	0.673	25	7	0	105

*top pct - top percentile
CS – Concrete Sand
CrS – Crushed Sand
GQ – Ground Quartz

	ρ_{column}	$\rho_{cracking}$
min	2.08	50
max	3.29	74.4

Table C3. Mixtures illustrated in Fig. 29 and Fig. 30. NSC, SCC and HSC from the literature have coarse aggregates (CA) in their composition instead of crushed sand (differentiated with an outlined box in the table)

References	SCMs (% by wt) replacing cement content						aggregates (% by wt) replacing cm content			Batching weigths (kg/m3) - after adjusting for yield								
	SL	SF	FA	w/ cm	HRWR/ cm	C	CcS	CrS	GQ	SL	SF	FA	C	W	HRWR (kg/m3)	CcS	CrS	GQ
	UHPC mixtures tested following New Test protocol																	
Section 5.2 - Fig.16	0	10.1	0	0.2	1.38%	89.9	0	6.5	22	0.0	145.0	0.0	1290.8	275.9	19.8	0.0	132.5	441.8
Section 5.2 - Fig.16	22.4	5.25	1.97	0.2	1.38%	70.4	0	6.5	22	320.1	75.0	28.2	1005.9	274.6	19.7	0.0	131.9	439.8
Section 5.2 - Fig.16	22.4	5.25	1.97	0.2	1.38%	70.4	22	8.5	21.5	230.7	54.1	20.3	724.9	203.3	14.2	476.1	185.1	461.4
	UHPC mixtures tested following ASTM C1856																	
Section 5.2 - Fig.16	0	10.1	0	0.2	1.38%	89.9	0	6.5	22	0.0	145.0	0.0	1290.8	275.9	19.8	0.0	132.5	441.8
Section 5.2 - Fig.16	22.4	5.25	1.97	0.2	1.38%	70.4	0	6.5	22	320.1	75.0	28.2	1005.9	274.6	19.7	0.0	131.9	439.8
Section 5.2 - Fig.16	22.4	5.25	1.97	0.2	1.38%	70.4	22	8.5	21.5	230.7	54.1	20.3	724.9	203.3	14.2	476.1	185.1	461.4
	HPCs and UHPCs developed in sections 6.6 and 7.2 following new test protocol																	
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.38	21.5	2.5	0	332.7	78.0	29.3	1045.2	287.6	20.5	423.6	49.6	0.0
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	0	238.3	55.9	21.0	748.8	215.6	14.7	536.5	540.0	0.0
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.38	11	2.5	11.5	330.6	77.5	29.1	1038.9	284.1	20.4	218.3	49.9	226.3
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.4	20.5	25	25	151.7	35.6	13.3	476.8	141.9	9.3	474.7	582.7	574.1
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	25	130.5	30.6	11.5	410.2	125.0	8.0	587.7	591.5	582.8
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.4	19.5	25	0	260.1	61.0	22.9	817.4	233.0	16.0	411.5	531.0	0.0
Section 6.6 - Table 8	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	0	238.3	55.9	21.0	748.8	215.6	14.7	536.5	540.0	0.0
Section 6.6 - Table 8	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	0	238.3	55.9	21.0	748.8	215.6	14.7	536.5	540.0	0.0
Section 6.6 - Fig.21	22.4	5.25	1.97	0.2	1.38%	70.4	15	15	12.5	268.2	62.9	23.6	842.7	235.8	16.5	315.0	317.0	260.3
Section 6.6 - Fig.21	22.4	5.25	1.97	0.2	1.38%	70.4	19	5	10	298.7	70.0	26.3	938.6	259.2	18.4	387.1	102.5	202.1
Section 6.6 - Fig.21	22.4	5.25	1.97	0.2	1.38%	70.4	16	1.5	13.5	309.6	72.6	27.2	972.7	266.8	19.1	323.2	30.5	270.4
Section 6.6 - Table 8	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	25	130.5	30.6	11.5	410.2	125.0	8.0	587.7	591.5	582.8
Section 6.6 - Table 8	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	25	130.5	30.6	11.5	410.2	125.0	8.0	587.7	591.5	582.8
Section 7.2 - Table 9	22.4	5.25	1.97	0.2	1.38%	70.4	25	7.5	13	255.8	60.0	22.5	803.7	224.7	15.8	528.3	159.5	272.4
Section 7.2 - Table 9	22.4	5.25	1.97	0.2	1.38%	70.4	20	25	21.5	169.8	39.8	14.9	533.5	157.0	10.5	456.4	574.2	486.5
	NSC mixturesfrom the literature							CA										
Siddique 2004	0	0	0	0.41	0.56%	100	27.5	54.7	0	0.0	0.0	0.0	389.9	177.2	2.2	605.6	1210.4	0.0
Siddique 2004	0	0	40	0.4	0.64%	60	27.4	54.7	0	0.0	0.0	153.1	229.6	169.8	2.4	592.4	1184.0	0.0
Siddique 2004	0	0	45	0.41	0.66%	55	27.5	54.7	0	0.0	0.0	170.6	208.5	172.0	2.5	588.9	1177.0	0.0
Siddique 2004	0	0	50	0.4	0.69%	50	27.5	54.7	0	0.0	0.0	189.7	189.7	168.3	2.6	589.3	1177.9	0.0
	SCC mixtures from the literature																	
Miller 2016	0	0	0	0.35	1.43%	100	40	40	0	0.0	0.0	0.0	443.7	172.1	6.3	894.8	900.7	0.0
Miller 2016	0	0	30	0.35	1.43%	70	40	40	0	0.0	0.0	130.3	304.1	168.5	6.2	876.1	881.8	0.0
Miller 2016	0	0	50	0.35	1.32%	50	40	40	0	0.0	0.0	214.2	214.2	166.5	5.7	864.1	869.7	0.0
	HSCs mixtures from the literature																	
Poon (2000)	0	0	0	0.24	2.95%	100	31.1	41.0	0	0.0	0.0	0.0	631.6	156.4	18.6	710.9	942.0	0.0
Poon (2000)	0	0	25.0	0.24	2.95%	75	30.4	41.3	0	0.0	0.0	156.6	470.7	155.9	18.5	680.6	929.5	0.0
Poon (2000)	0	0	44.9	0.23	3.84%	55	29.2	42.0	0	0.0	0.0	281.9	345.6	150.6	24.1	641.8	930.1	0.0
Poon (2000)	0	0	0	0.19	5.10%	100	27.97	41.40	0	0.0	0.0	0.0	704.5	131.4	35.9	648.7	966.7	0.0
Poon (2000)	0	0	25.3	0.19	5.17%	75	27.7	41.7	0	0.0	0.0	174.0	515.0	129.5	35.6	628.9	951.6	0.0
Poon (2000)	0	0	45.1	0.19	5.09%	55	27.5	41.9	0	0.0	0.0	305.9	373.2	126.6	34.6	615.0	943.8	0.0
Lam (1997)	0	5.0	0	0.30	0.00%	95	31.3	47.0	0	0.0	24.3	0.0	461.5	163.7	0.0	704.5	1061.0	0.0
Lam (1997)	0	5.0	0	0.4	0.00%	95	30.6	51.5	0	0.0	19.5	0.0	370.9	174.8	0.0	677.3	1141.9	0.0
Lam (1997)	0	5.0	0	0.5	0.00%	95.0	28.2	52.7	0	0.0	19.9	0.0	378.0	216.4	0.0	592.0	1110.6	0.0
Einsfeld (2006)	0	10.1	0	0.3	2.51%	89.9	37.1	42.8	0	0.0	45.0	0.0	402.4	151.2	11.2	831.0	961.0	0.0
Einsfeld (2006)	0	9.9	0	0.3	2.49%	90.1	35.2	42.9	0	0.0	47.8	0.0	437.3	155.8	12.1	785.5	959.8	0.0
Einsfeld (2006)	0	10.0	0	0.3	2.52%	90.0	31.3	42.8	0	0.0	57.7	0.0	519.5	152.4	14.5	702.5	965.0	0.0
	UHPC mixtures from the literature																	
Abdulkareem (2018)	0	15.8	0	0.13	2.38%	84.2	46.8	0	2.7	0.0	181.3	0.0	967.9	135.0	27.3	1073.9	0.0	60.4
Abdulkareem (2018)	0	16.1	0	0.14	1.01%	83.9	47.3	0	2.7	0.0	181.0	0.0	941.4	160.0	11.4	1071.9	0.0	60.3
Abdulkareem (2018)	0	16.4	0	0.13	2.19%	83.6	47.7	0	2.7	0.0	183.4	0.0	937.3	141.4	24.6	1086.2	0.0	61.1
Abdulkareem (2018)	0	16.7	0	0.13	2.52%	83.3	48.2	0	2.7	0.0	185.2	0.0	921.4	137.9	27.9	1097.1	0.0	61.7

References	SCMs (% by wt) replacing cement content			w/cm	HRWR/cm	C	aggregates (% by wt) replacing cm content			56th-day f_c (MPa)	λ_{fr}	Total GWP for the mixture (kg CO2e/m3)	χ_{column}	$\chi_{cracking}$	Total cost for the mixture ($/m3)	ρ_{column}	$\rho_{cracking}$
	SL	SF	FA				CcS	CrS	GQ								
	UHPC mixtures tested following New Test protocol																
Section 5.2 - Fig.16	0	10.1	0	0.2	1.38%	89.9	0	6.5	22	128.4	0.54	1521.5	11.8	244.1	365.3	2.8	58.6
Section 5.2 - Fig.16	22.4	5.25	1.97	0.2	1.38%	70.4	0	6.5	22	129.2	0.54	1201.2	9.3	192.4	320.1	2.5	51.3
Section 5.2 - Fig.16	22.4	5.25	1.97	0.2	1.38%	70.4	22	8.5	21.5	128.5	0.54	893.0	6.9	143.2	270.8	2.1	43.4
	UHPC mixtures tested following ASTM C1856																
Section 5.2 - Fig.16	0	10.1	0	0.2	1.38%	89.9	0	6.5	22	133.8	0.54	1521.5	11.4	241.6	365.3	2.7	58.0
Section 5.2 - Fig.16	22.4	5.25	1.97	0.2	1.38%	70.4	0	6.5	22	158.1	0.54	1201.2	7.6	182.9	320.1	2.0	48.7
Section 5.2 - Fig.16	22.4	5.25	1.97	0.2	1.38%	70.4	22	8.5	21.5	154.8	0.54	893.0	5.8	136.7	270.8	1.7	41.5
	HPCs and UHPCs developed in sections 6.6 and 7.2 following new test protocol																
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.38	21.5	2.5	0	112.0	0.72	1241.8	11.1	274.8	232.8	2.1	51.5
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	0	80.0	0.72	916.4	11.5	220.6	189.3	2.4	45.6
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.38	11	2.5	11.5	118.0	0.72	1236.0	10.5	270.0	279.2	2.4	61.0
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.4	20.5	25	25	95.0	0.72	620.5	6.5	143.1	257.7	2.7	59.4
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	25	82.0	0.72	547.4	6.7	131.0	246.7	3.0	59.0
Section 6.6 - Table 7	22.4	5.25	1.97	0.2	1.38%	70.4	19.5	25	0	95.0	0.72	991.6	10.4	228.7	202.6	2.1	46.7
Section 6.6 - Table 8	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	0	80.0	0.72	916.4	11.5	220.6	189.3	2.4	45.6
Section 6.6 - Table 8	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	0	80.0	0.72	916.4	11.5	220.6	189.3	2.4	45.6
Section 6.6 - Fig.21	22.4	5.25	1.97	0.2	1.38%	70.4	15	15	12.5	88.9	0.72	1020.7	11.5	239.3	255.8	2.9	60.0
Section 6.6 - Fig.21	22.4	5.25	1.97	0.2	1.38%	70.4	19	5	10	111.0	0.72	1125.7	10.1	249.7	255.9	2.3	56.8
Section 6.6 - Fig.21	22.4	5.25	1.97	0.2	1.38%	70.4	16	1.5	13.5	118.0	0.72	1163.6	9.9	254.2	274.8	2.3	60.0
Section 6.6 - Table 8	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	25	81.7	0.72	547.4	6.7	131.1	246.7	3.0	59.1
Section 6.6 - Table 8	22.4	5.25	1.97	0.2	1.38%	70.4	25	25	25	81.7	0.72	547.4	6.7	131.1	246.7	3.0	59.1
Section 7.2 - Table 9	22.4	5.25	1.97	0.2	1.38%	70.4	25	7.5	13	110.0	0.72	978.0	8.9	217.4	245.8	2.2	54.7
Section 7.2 - Table 9	22.4	5.25	1.97	0.2	1.38%	70.4	20	25	21.5	97.0	0.72	682.4	7.0	156.6	250.2	2.6	57.4
	NSC mixtures from the literature							CA									
Siddique 2004	0	0	0	0.41	0.56%	100	27.5	54.7	0	35.9	1.10	514.7	14.4	231.4	96.3	2.7	43.3
Siddique 2004	0	0	40	0.4	0.64%	60	27.4	54.7	0	28.2	1.10	333.3	11.8	159.2	85.2	3.0	40.7
Siddique 2004	0	0	45	0.41	0.66%	55	27.5	54.7	0	25.6	1.10	309.2	12.1	151.2	83.7	3.3	40.9
Siddique 2004	0	0	50	0.4	0.69%	50	27.5	54.7	0	23.8	1.10	288.2	12.1	143.6	82.8	3.5	41.3
	SCC mixtures from the literature																
Miller 2016	0	0	0	0.35	1.43%	100	40	40	0	60.1	0.89	579.5	9.6	185.2	104.1	1.7	33.3
Miller 2016	0	0	30	0.35	1.43%	70	40	40	0	57.8	0.89	420.1	7.3	135.6	93.4	1.6	30.2
Miller 2016	0	0	50	0.35	1.32%	50	40	40	0	48.2	0.89	317.8	6.6	107.3	85.2	1.8	28.8
	HSCs mixtures from the literature																
Poon (2000)	0	0	0	0.24	2.95%	100	31.1	41.0	0	90.2	0.72	798.9	8.9	186.7	157.6	1.7	36.8
Poon (2000)	0	0	25.0	0.24	2.95%	75	30.4	41.3	0	100.1	0.72	616.4	6.2	140.3	145.9	1.5	33.2
Poon (2000)	0	0	44.9	0.23	3.84%	55	29.2	42.0	0	85.4	0.72	478.9	5.6	113.4	154.1	1.8	36.5
Poon (2000)	0	0	0	0.19	5.10%	100	27.97	41.40	0	91.8	0.72	894.5	9.7	208.1	216.5	2.4	50.4
Poon (2000)	0	0	25.3	0.19	5.17%	75	27.7	41.7	0	98.1	0.72	679.8	6.9	155.6	202.0	2.1	46.2
Poon (2000)	0	0	45.1	0.19	5.09%	55	27.5	41.9	0	85.9	0.72	518.6	6.0	122.7	189.0	2.2	44.7
Lam (1997)	0	5.0	0	0.30	0.00%	95	31.3	47.0	0	78.8	0.89	593.8	7.5	177.4	105.7	1.3	31.6
Lam (1997)	0	5.0	0	0.4	0.00%	95	30.6	51.5	0	62.2	0.89	491.9	7.9	155.9	97.7	1.6	31.0
Lam (1997)	0	5.0	0	0.5	0.00%	95.0	28.2	52.7	0	54.1	0.89	496.0	9.2	162.8	96.5	1.8	31.7
Einsfeld (2006)	0	10.1	0	0.3	2.51%	89.9	37.1	42.8	0	62.9	0.89	536.7	8.5	169.6	143.8	2.3	45.5
Einsfeld (2006)	0	9.9	0	0.3	2.49%	90.1	35.2	42.9	0	82.3	0.72	575.6	7.0	137.6	150.7	1.8	36.0
Einsfeld (2006)	0	10.0	0	0.3	2.52%	90.0	31.3	42.8	0	85.2	0.72	669.0	7.9	158.5	170.7	2.0	40.5
	UHPC mixtures from the literature																
Abdulkareem (2018)	0	15.8	0	0.13	2.38%	84.2	46.8	0	2.7	121.0	0.54	1177.4	9.7	191.7	302.0	2.5	49.2
Abdulkareem (2018)	0	16.1	0	0.14	1.01%	83.9	47.3	0	2.7	123.0	0.54	1134.3	9.2	183.9	251.5	2.0	40.8
Abdulkareem (2018)	0	16.4	0	0.13	2.19%	83.6	47.7	0	2.7	117.0	0.72	1140.7	9.7	249.7	292.5	2.5	64.0
Abdulkareem (2018)	0	16.7	0	0.13	2.52%	83.3	48.2	0	2.7	70.0	0.89	1125.6	16.1	346.3	302.5	4.3	93.1

APPENDIX D

Codes used to develop the models in the study herein

This report summarizes the results from Phase A of the Optimization study on UHPC samples tested at the age of 56 day old.

```{r setup, include=FALSE}
knitr::opts_chunk$set(
	echo = FALSE,
	message = FALSE,
	warning = FALSE
)
```

```{r loading results, echo=FALSE, message=FALSE, warning=FALSE}
library(dslabs)
library(tidyverse)
library(dplyr)
library(tidyverse)
library(tidytext)
library(ggthemes)
library(ggrepel)
library(gridExtra)
library(gtools)
library(rvest)
library(purrr)
library(pdftools)
library(stringr)
library(lubridate)
library(tinytex)
library(readxl)
library(broom)
library(reshape2)
library(lpSolve)
library(caret)
library(e1071)
library(matrixStats)
library(randomForest)
library(openxlsx)
library(rio)
library(gam)
library("plot3D")
library("writexl")

options(digits = 3)
```

```
Phase_A_Results <- read_excel("D:/NSF Project/UHPC/PASTE EVALUATION/taguchi method - Mostafa paper/Results/Phase A - 1 day results - Rstudio.xlsx")

Phase_AA_Results <- read_excel("D:/NSF Project/UHPC/PASTE EVALUATION/taguchi method - Mostafa paper/Results/Phase AA - 1 day results - Rstudio.xlsx")

Phase_O_Results <- read_excel("D:/NSF Project/UHPC/PASTE EVALUATION/taguchi method - Mostafa paper/Results/Phase O - 1 day results - Rstudio.xlsx")
```

```{r data wrangling, echo=FALSE, message=FALSE, warning=FALSE}
Phase_A_Results <- data.frame(Phase_A_Results,stringsAsFactors = TRUE)
Phase_A_Results$Original_description<-as.factor(Phase_A_Results$Original_description)
Phase_A_Results$Material_number<-as.factor(Phase_A_Results$Material_number)
Phase_A_Results$Test_Protocol<-as.factor(Phase_A_Results$Test_Protocol)
Phase_A_Results$End_specimen_condition<-as.factor(Phase_A_Results$End_specimen_condition)
Phase_A_Results$Pre_treatment<-as.factor(Phase_A_Results$Pre_treatment)
Phase_A_Results$Molding_material<-as.factor(Phase_A_Results$Molding_material)
Phase_A_Results$Reference<-as.factor(Phase_A_Results$Reference)
Phase_A_Results$Mixing_Process<-as.factor(Phase_A_Results$Mixing_Process)
Phase_A_Results$Fracture_Pattern_56D<-as.factor(Phase_A_Results$Fracture_Pattern_56D)

Phase_AA_Results <- data.frame(Phase_AA_Results,stringsAsFactors = TRUE)
Phase_AA_Results$Original_description<-as.factor(Phase_AA_Results$Original_description)
Phase_AA_Results$Material_number<-as.factor(Phase_AA_Results$Material_number)
Phase_AA_Results$Test_Protocol<-as.factor(Phase_AA_Results$Test_Protocol)
Phase_AA_Results$End_specimen_condition<-as.factor(Phase_AA_Results$End_specimen_condition)
Phase_AA_Results$Pre_treatment<-as.factor(Phase_AA_Results$Pre_treatment)
Phase_AA_Results$Molding_material<-as.factor(Phase_AA_Results$Molding_material)
Phase_AA_Results$Reference<-as.factor(Phase_AA_Results$Reference)
Phase_AA_Results$Mixing_Process<-as.factor(Phase_AA_Results$Mixing_Process)
Phase_AA_Results$Fracture_Pattern_56D<-as.factor(Phase_AA_Results$Fracture_Pattern_56D)

Phase_O_Results <- data.frame(Phase_O_Results,stringsAsFactors = TRUE)
Phase_O_Results$Original_description<-as.factor(Phase_O_Results$Original_description)
Phase_O_Results$Material_number<-as.factor(Phase_O_Results$Material_number)
Phase_O_Results$Test_Protocol<-as.factor(Phase_O_Results$Test_Protocol)
Phase_O_Results$End_specimen_condition<-as.factor(Phase_O_Results$End_specimen_condition)
Phase_O_Results$Pre_treatment<-as.factor(Phase_O_Results$Pre_treatment)
Phase_O_Results$Molding_material<-as.factor(Phase_O_Results$Molding_material)
Phase_O_Results$Reference<-as.factor(Phase_O_Results$Reference)
Phase_O_Results$Mixing_Process<-as.factor(Phase_O_Results$Mixing_Process)
Phase_O_Results$Fracture_Pattern_56D<-as.factor(Phase_O_Results$Fracture_Pattern_56D)

Phase_A_Results_combined <- full_join(Phase_A_Results,Phase_AA_Results)
```

Experimental results

```{r Summary of experimental results, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

options(digits = 2)
Experimental_mix_rank <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear mixing")%>% group_by(Material_number)%>% summarize(Slag=mean(Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAsh_percent_wt),wcm=mean(w_cm),Stress_1D=mean(Stress_1D_MPa),Stress_56D=mean(Stress_56D_MPa)) %>% as.data.frame()

Phase_A_Results %>% filter(Mixing_Process=="High shear mixing")%>% group_by(Material_number)%>% summarize(Stress_MPa=mean(Stress_56D_MPa),sd_MPa=sd(Stress_56D_MPa)) %>% select(Material_number,Stress_MPa,sd_MPa) %>% as.matrix()

Phase_A_Results %>% filter(Mixing_Process=="High shear mixing")%>% group_by(Material_number)%>% summarize(Stress_MPa=mean(Stress_56D_MPa),sd_MPa=sd(Stress_56D_MPa)) %>% select(sd_MPa) %>% summarize(sd_mean=mean(sd_MPa))

Phase_AA_Results %>% filter(Mixing_Process=="High shear mixing")%>% group_by(Material_number)%>% summarize(Stress_MPa=mean(Stress_56D_MPa),sd_MPa=sd(Stress_56D_MPa)) %>% select(Material_number,Stress_MPa,sd_MPa) %>% as.matrix()

Phase_AA_Results %>% filter(Mixing_Process=="High shear mixing")%>% group_by(Material_number)%>% summarize(Stress_MPa=mean(Stress_56D_MPa),sd_MPa=sd(Stress_56D_MPa)) %>% select(sd_MPa) %>% summarize(sd_mean=mean(sd_MPa))

Phase_O_Results %>% filter(Mixing_Process=="High shear mixing")%>% group_by(Material_number)%>% summarize(Stress_MPa=mean(Stress_56D_MPa),sd_MPa=sd(Stress_56D_MPa)) %>% select(Material_number,Stress_MPa,sd_MPa) %>% as.matrix()

Phase_O_Results %>% filter(Mixing_Process=="High shear mixing")%>% group_by(Material_number)%>% summarize(Stress_MPa=mean(Stress_56D_MPa),sd_MPa=sd(Stress_56D_MPa)) %>% select(sd_MPa) %>% summarize(sd_mean=mean(sd_MPa))
```

Fracture Pattern vs Stress

```{r Fracture Pattern vs Stress, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}
```

```
full_join(Phase_A_Results_combined,Phase_O_Results) %>% filter(Mixing_Process=="High shear
mixing") %>% mutate(Material_number=reorder(Material_number,Stress_56D_MPa,FUN = mean)) %>%
ggplot(aes(Fracture_Pattern_56D,Stress_56D_MPa,color=w_cm))+geom_point()+geom_text_repel(aes(
Fracture_Pattern_56D,Stress_56D_MPa,label=Material_number))

full_join(Phase_A_Results_combined,Phase_O_Results) %>% filter(Mixing_Process=="High shear
mixing") %>% mutate(Material_number=reorder(Material_number,Stress_56D_MPa,FUN = mean)) %>%
ggplot(aes(Fracture_Pattern_56D,Stress_56D_MPa,color=HRWR_per_cm_by_wt))+geom_point()+geom
_text_repel(aes(Fracture_Pattern_56D,Stress_56D_MPa,label=Material_number))

full_join(Phase_A_Results_combined,Phase_O_Results) %>% filter(Mixing_Process=="High shear
mixing") %>% mutate(Material_number=reorder(Material_number,Stress_56D_MPa,FUN = mean)) %>%
ggplot(aes(Fracture_Pattern_56D,Stress_56D_MPa,color=Slump_cm))+geom_point()+geom_text_repel(
aes(Fracture_Pattern_56D,Stress_56D_MPa,label=Material_number))

```
```

Cross validation method with k-folds to optimize Knn model model to fit matrix with average stress results per Mix

```{r Evaluate k values for matrix with average stress, echo=FALSE, fig.height=8, fig.width=16,
message=FALSE, warning=FALSE}

set.seed(1997, sample.kind = "Rounding")
FSixDaysPredictors <- Phase_A_Results %>% filter(Mixing_Process=="High shear mixing")%>%
group_by(Material_number)%>% summarize(Stress=mean(Stress_56D_MPa),Slag=mean(
Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAsh_percent_wt),wcm=m
ean(w_cm)) %>% select(Slag,Microsilica,FlyAsh,wcm) %>% as.matrix()
FSixDaysOutcome <- Phase_A_Results %>% filter(Mixing_Process=="High shear mixing")%>%
group_by(Material_number)%>% summarize(stress=mean(Stress_56D_MPa)) %>% select(stress) %>%
as.matrix()
FSixDaysOutcome <- as.numeric(FSixDaysOutcome)

set.seed(1997, sample.kind = "Rounding")
inTrain_80 <- createDataPartition(FSixDaysOutcome,p=0.8,list = FALSE)
trainX_80 <- FSixDaysPredictors[inTrain_80,]
trainY_80 <- FSixDaysOutcome[inTrain_80]
testX_80 <- FSixDaysPredictors[-inTrain_80,]
testY_80 <- FSixDaysOutcome[-inTrain_80]
train_set_80 <- cbind(trainX_80,stress=trainY_80) %>% as.data.frame()
test_set_80 <- cbind(testX_80,stress=testY_80) %>% as.data.frame()
trControl <- trainControl(method = "repeatedcv", number = 5, repeats = 10, p=0.8)
fit_knn_avg_80 <- train(stress ~. ,data =
train_set_80,tuneGrid=expand.grid(k=1:15),method="knn",trControl=trControl,preProc=c('center','scale
'))

set.seed(1997, sample.kind = "Rounding")
inTrain_100 <- createDataPartition(FSixDaysOutcome,p=1,list = FALSE)
```
```

```
trainX_100 <- FSixDaysPredictors[inTrain_100,]
trainY_100 <- FSixDaysOutcome[inTrain_100]
testX_100 <- FSixDaysPredictors[-inTrain_100,]
testY_100 <- FSixDaysOutcome[-inTrain_100]
train_set_100 <- cbind(trainX_100,stress=trainY_100) %>% as.data.frame()
test_set_100 <- cbind(testX_100,stress=testY_100) %>% as.data.frame()
trControl <- trainControl(method = "repeatedcv", number = 5, repeats = 10, p=0.8)
fit_knn_avg_100 <- train(stress ~. ,data =
train_set_100,tuneGrid=expand.grid(k=1:18),method="knn",trControl=trControl,preProc=c('center','scal
e'))

set.seed(1997, sample.kind = "Rounding")
FSixDaysPredictorscombined <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear
mixing")%>% group_by(Material_number)%>% summarize(Stress=mean(Stress_56D_MPa),Slag=mean(
Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAsh_percent_wt),wcm=m
ean(w_cm)) %>% select(Slag,Microsilica,FlyAsh,wcm) %>% as.matrix()
FSixDaysOutcomecombined <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear
mixing")%>% group_by(Material_number)%>% summarize(stress=mean(Stress_56D_MPa)) %>%
select(stress) %>% as.matrix()
FSixDaysOutcomecombined <- as.numeric(FSixDaysOutcomecombined)

set.seed(1997, sample.kind = "Rounding")
inTrain_80_combined <- createDataPartition(FSixDaysOutcomecombined,p=0.8,list = FALSE)
trainX_80_combined <- FSixDaysPredictorscombined[inTrain_80_combined,]
trainY_80_combined <- FSixDaysOutcomecombined[inTrain_80_combined]
testX_80_combined <- FSixDaysPredictorscombined[-inTrain_80_combined,]
testY_80_combined <- FSixDaysOutcomecombined[-inTrain_80_combined]
train_set_80_combined <- cbind(trainX_80_combined,stress=trainY_80_combined) %>% as.data.frame()
test_set_80_combined <- cbind(testX_80_combined,stress=testY_80_combined) %>% as.data.frame()
trControl_80 <- trainControl(method = "repeatedcv", number = 8, repeats = 8, p=0.8)
fit_knn_avg_80_combined <- train(stress ~. ,data =
train_set_80_combined,tuneGrid=expand.grid(k=1:24),method="knn",trControl=trControl_80,preProc=c
('center','scale'))

set.seed(1997, sample.kind = "Rounding")
FSixDaysPredictorsAA <- Phase_AA_Results %>% filter(Mixing_Process=="High shear mixing")%>%
group_by(Material_number)%>% summarize(Stress=mean(Stress_56D_MPa),Slag=mean(
Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAsh_percent_wt),wcm=m
ean(w_cm)) %>% select(Slag,Microsilica,FlyAsh,wcm) %>% as.matrix()
FSixDaysOutcomeAA <- Phase_AA_Results %>% filter(Mixing_Process=="High shear mixing")%>%
group_by(Material_number)%>% summarize(stress=mean(Stress_56D_MPa)) %>% select(stress) %>%
as.matrix()
FSixDaysOutcomeAA <- as.numeric(FSixDaysOutcomeAA)

set.seed(1997, sample.kind = "Rounding")
inTrain_AA <- createDataPartition(FSixDaysOutcomeAA,p=(10/15),list = FALSE)
```

```
trainX_final <-
as.matrix(full_join(as.data.frame(FSixDaysPredictors),as.data.frame(FSixDaysPredictorsAA[inTrain_AA,]))
)
FSixDaysOutcomeFinal <- FSixDaysOutcome %>% as.data.frame() %>% mutate(stress=FSixDaysOutcome)
%>% select(stress)
FSixDaysOutcomeFinalAA <- FSixDaysOutcomeAA[inTrain_AA] %>% as.data.frame() %>%
mutate(stress=FSixDaysOutcomeAA[inTrain_AA]) %>% select(stress)
trainY_final <-
as.matrix(full_join(as.data.frame(FSixDaysOutcomeFinal),as.data.frame(FSixDaysOutcomeFinalAA)))
testX_final <- FSixDaysPredictorsAA[-inTrain_AA,]
testY_final <- FSixDaysOutcomeAA[-inTrain_AA]
train_set_final <- cbind(trainX_final,stress=trainY_final) %>% as.data.frame()
test_set_final <- cbind(testX_final,stress=testY_final) %>% as.data.frame()
trControl <- trainControl(method = "repeatedcv", number = 8, repeats = 8, p=0.875)
fit_knn_avg_final <- train(stress ~. ,data =
train_set_final,tuneGrid=expand.grid(k=1:24),method="knn",trControl=trControl,preProc=c('center','sca
le'))

set.seed(1997, sample.kind = "Rounding")
inTrain_100_combined <- createDataPartition(FSixDaysOutcomecombined,p=1,list = FALSE)
trainX_100_combined <- FSixDaysPredictorscombined[inTrain_100_combined,]
trainY_100_combined <- FSixDaysOutcomecombined[inTrain_100_combined]
testX_100_combined <- FSixDaysPredictorscombined[-inTrain_100_combined,]
testY_100_combined <- FSixDaysOutcomecombined[-inTrain_100_combined]
train_set_100_combined <- cbind(trainX_100_combined,stress=trainY_100_combined) %>%
as.data.frame()
test_set_100_combined <- cbind(testX_100_combined,stress=testY_100_combined) %>%
as.data.frame()
trControl <- trainControl(method = "repeatedcv", number = 10, repeats = 10, p=0.9)
fit_knn_avg_100_combined <- train(stress ~. ,data =
train_set_100_combined,tuneGrid=expand.grid(k=1:24),method="knn",trControl=trControl,preProc=c('c
enter','scale'))

set.seed(1997, sample.kind = "Rounding")
FSixDaysPredictorsO <- Phase_O_Results %>% filter(Mixing_Process=="High shear mixing")%>%
group_by(Material_number)%>% summarize(Stress=mean(Stress_56D_MPa),Slag=mean(
Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAsh_percent_wt),wcm=m
ean(w_cm)) %>% select(Slag,Microsilica,FlyAsh,wcm) %>% as.matrix()
FSixDaysOutcomeO <- Phase_O_Results %>% filter(Mixing_Process=="High shear mixing")%>%
group_by(Material_number)%>% summarize(stress=mean(Stress_56D_MPa)) %>% select(stress) %>%
as.matrix()
FSixDaysOutcomeO <- as.numeric(FSixDaysOutcomeO)

testX_O <- FSixDaysPredictorsO
testY_O <- FSixDaysOutcomeO
test_set_O <- cbind(testX_O,stress=testY_O) %>% as.data.frame()

```

Evaluate knn model generated by fitting matrix with average stress results

```{r Evaluate knn model fitting matrix with average stress results, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

options(digits = 3)
knn_avg_methods <-
c("fit_knn_avg_80","fit_knn_avg_100","fit_knn_avg_80_combined","fit_knn_avg_final","fit_knn_avg_100_combined")
best_tunes <-
c(as.numeric(fit_knn_avg_80$bestTune),as.numeric(fit_knn_avg_100$bestTune),as.numeric(fit_knn_avg_80_combined$bestTune),as.numeric(fit_knn_avg_final$bestTune),as.numeric(fit_knn_avg_100_combined$bestTune))
knn_avg_train_rmse <-
c(min(fit_knn_avg_80$results$RMSE),min(fit_knn_avg_100$results$RMSE),min(fit_knn_avg_80_combined$results$RMSE),min(fit_knn_avg_final$results$RMSE),min(fit_knn_avg_100_combined$results$RMSE))

pred_knn_avg_80 <- predict(fit_knn_avg_80,newdata = test_set_80)
pred_knn_avg_80_combined <- predict(fit_knn_avg_80_combined,newdata = test_set_80_combined)
pred_knn_avg_final <- predict(fit_knn_avg_final,newdata = test_set_final)

pred_knn_avg_80_O <- predict(fit_knn_avg_80,newdata = test_set_O)
pred_knn_avg_100_O <- predict(fit_knn_avg_100,newdata = test_set_O)
pred_knn_avg_80_combined_O <- predict(fit_knn_avg_80_combined,newdata = test_set_O)
pred_knn_avg_final_O <- predict(fit_knn_avg_final,newdata = test_set_O)
pred_knn_avg_100_combined_O <- predict(fit_knn_avg_100_combined,newdata = test_set_O)

knn_avg_test_rmse <-
c(RMSE(pred_knn_avg_80,test_set_80$stress),NA,RMSE(pred_knn_avg_80_combined,test_set_80_combined$stress),RMSE(pred_knn_avg_final,test_set_final$stress),NA)

knn_avg_O_rmse <-
c(RMSE(pred_knn_avg_80_O,test_set_O$stress),RMSE(pred_knn_avg_100_O,test_set_O$stress),RMSE(pred_knn_avg_80_combined_O,test_set_O$stress),RMSE(pred_knn_avg_final_O,test_set_O$stress),RMSE(pred_knn_avg_100_combined_O,test_set_O$stress))

knn_avg_methods_rmse_summary <-
data.frame(method=knn_avg_methods,k=best_tunes,train_rmse=knn_avg_train_rmse,test_rmse=knn_avg_test_rmse,optimum_test_rmse=knn_avg_O_rmse)

knn_avg_methods_rmse_summary
plot(fit_knn_avg_80)
plot(fit_knn_avg_100)
plot(fit_knn_avg_80_combined)
plot(fit_knn_avg_final)
plot(fit_knn_avg_100_combined)
varImp(fit_knn_avg_80)
```

```
varImp(fit_knn_avg_100)
varImp(fit_knn_avg_80_combined)
varImp(fit_knn_avg_final)
varImp(fit_knn_avg_100_combined)

```

Evaluate RMSE on test data knn avg

```{r Evaluate RMSE on test data knn avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

plot(pred_knn_avg_80,test_set_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_knn_avg_80_combined,test_set_80_combined$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_knn_avg_final,test_set_final$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)

plot(pred_knn_avg_80_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_knn_avg_100_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_knn_avg_80_combined_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_knn_avg_final_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_knn_avg_100_combined_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)

```

Generating expanded matrix with predictors

```{r Generating expanded matrix with predictors, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

SlagPredictor <- as.matrix(seq(0,60,length.out = 100))
MicrosilicaPredictor <- as.matrix(seq(0,20,length.out = 100))
FlyAshPredictor <- as.matrix(seq(0,15,length.out = 100))
PredictorsMatrix <-
data.frame(Slag=SlagPredictor,Microsilica=MicrosilicaPredictor,FlyAsh=FlyAshPredictor)
options(digits = 1)
ExpandedPredictorsMatrix <- expand.grid(df=PredictorsMatrix) %>% mutate(wcm=0.2)
```

Generating the predictive vector for the expanded matrix knn avg

```{r Generating the predictive vector for the expanded matrix knn avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysYhat_knn_avg_80 <- predict(fit_knn_avg_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_knn_avg_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_knn_avg_80)
FSixDaysTrialResults_knn_avg_80 <- FSixDaysTrialResults_knn_avg_80 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_knn_avg_100 <- predict(fit_knn_avg_100,ExpandedPredictorsMatrix)
FSixDaysTrialResults_knn_avg_100 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_knn_avg_100)
FSixDaysTrialResults_knn_avg_100 <- FSixDaysTrialResults_knn_avg_100 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_knn_avg_80_combined <- predict(fit_knn_avg_80_combined,ExpandedPredictorsMatrix)
FSixDaysTrialResults_knn_avg_80_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_knn_avg_80_combined)
FSixDaysTrialResults_knn_avg_80_combined <- FSixDaysTrialResults_knn_avg_80_combined %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_knn_avg_final <- predict(fit_knn_avg_final,ExpandedPredictorsMatrix)
FSixDaysTrialResults_knn_avg_final <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_knn_avg_final)
FSixDaysTrialResults_knn_avg_final <- FSixDaysTrialResults_knn_avg_final %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_knn_avg_100_combined <-
predict(fit_knn_avg_100_combined,ExpandedPredictorsMatrix)
FSixDaysTrialResults_knn_avg_100_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_knn_avg_100_combined)
FSixDaysTrialResults_knn_avg_100_combined <- FSixDaysTrialResults_knn_avg_100_combined %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)
```

Performance Density Diagram - faceted by Slag Content

```{r Performance Density Diagram - faceted by Slag Content, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}
FSixDaysTrialResults_knn_avg_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)
```

```
FSixDaysTrialResults_knn_avg_100 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_knn_avg_80_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_knn_avg_final %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_knn_avg_100_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)
```

Performance Density Diagram - faceted by Fly Ash Content

```
{r Performance Density Diagram - faceted by Fly Ash Content, echo=FALSE, fig.height=8, fig.width=16,
message=FALSE, warning=FALSE}
FSixDaysTrialResults_knn_avg_80 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_knn_avg_100 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_knn_avg_80_combined %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
```

l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_knn_avg_final %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_knn_avg_100_combined %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)
```

### Performance Density Diagram - faceted by Microsilica Content

```{r Performance Density Diagram - faceted by Microsilica Content, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_knn_avg_80 %>% filter(Microsilica %in% c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],ExpandedPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>% ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller = label_both)

FSixDaysTrialResults_knn_avg_100 %>% filter(Microsilica %in% c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],ExpandedPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>% ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller = label_both)

FSixDaysTrialResults_knn_avg_80_combined %>% filter(Microsilica %in% c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],ExpandedPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>% ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller = label_both)

FSixDaysTrialResults_knn_avg_final %>% filter(Microsilica %in% c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],ExpandedPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>% ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller = label_both)

```
FSixDaysTrialResults_knn_avg_100_combined %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)
```

###Optimum mix knn avg

```{r Optimum mix, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

optimum_knn_avg_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_knn_avg_80$Expected_Stress
_MPa),])
optimum_knn_avg_100 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_knn_avg_100$Expected_Stre
ss_MPa),])
optimum_knn_avg_80_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_knn_avg_80_combined$Expected_Stress_M
Pa),])
optimum_knn_avg_final <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_knn_avg_final$Expected_Stre
ss_MPa),])
optimum_knn_avg_100_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_knn_avg_100_combined$Expected_Stress_
MPa),])

optimum_knn_avg_mixtures <-
rbind(optimum_knn_avg_80,optimum_knn_avg_100,optimum_knn_avg_80_combined,optimum_knn_a
vg_final,optimum_knn_avg_100_combined) %>% as.data.frame()

optimum_knn_avg_max <-
c(max(FSixDaysTrialResults_knn_avg_80),max(FSixDaysTrialResults_knn_avg_100),max(FSixDaysTrialRes
ults_knn_avg_80_combined),max(FSixDaysTrialResults_knn_avg_final),max(FSixDaysTrialResults_knn_a
vg_100_combined))

optimum_knn_avg_summary <-
data.frame(methods=knn_avg_methods,optimum_knn_avg_mixtures,predicted_stress_MPa=optimum_
knn_avg_max)
optimum_knn_avg_summary

test_set_O

Experimental_mix_rank <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear
mixing")%>% group_by(Material_number)%>%
summarize(Slag=mean(Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAs
```

h_percent_wt),Stress=mean(Stress_56D_MPa),wcm=mean(w_cm)) %>% arrange(desc(Stress)) %>%
as.data.frame() %>% head(n=10)
Experimental_mix_rank

```
### Cross validation method with groupKfolds (by mix) to optimize Knn model with individual specimen results. All Phase A data was used as training data, while test data will be evaluated from optimum mixes batching

```{r Evaluate k values for matrix with individual stress results, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

set.seed(1997, sample.kind = "Rounding")

FSixDaysPredictors_sp <- Phase_A_Results %>% filter(Mixing_Process=="High shear mixing") %>%
select(Slag=Slag_percent_wt,Microsilica=Microsilica_percent_wt,FlyAsh=FlyAsh_percent_wt,wcm=w_cm)%>% as.matrix()
FSixDaysOutcome_sp <- Phase_A_Results %>% filter(Mixing_Process=="High shear mixing") %>%
select(stress=Stress_56D_MPa)%>% as.matrix()
FSixDaysOutcome_sp <- as.numeric(FSixDaysOutcome_sp)

FSixDaysPredictors_sp_combined <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear mixing") %>%
select(Slag=Slag_percent_wt,Microsilica=Microsilica_percent_wt,FlyAsh=FlyAsh_percent_wt,wcm=w_cm)%>% as.matrix()
FSixDaysOutcome_sp_combined <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear mixing") %>% select(stress=Stress_56D_MPa)%>% as.matrix()
FSixDaysOutcome_sp_combined <- as.numeric(FSixDaysOutcome_sp_combined)

set.seed(1997, sample.kind = "Rounding")
x <- Phase_A_Results %>% filter(Mixing_Process=="High shear mixing") %>%
select(Material_number)%>% mutate(Material_number=as.numeric(Material_number))%>%
as.data.frame()
x <- x$Material_number
folds <- groupKFold(x,k=5)
inTrain_sp_80 <- which((Phase_A_Results$Material_number %in% folds$Fold1))
trainX_sp_80 <- FSixDaysPredictors_sp[inTrain_sp_80,]
trainY_sp_80 <- FSixDaysOutcome_sp[inTrain_sp_80]
testX_sp_80 <- FSixDaysPredictors_sp[-inTrain_sp_80,]
testY_sp_80 <- FSixDaysOutcome_sp[-inTrain_sp_80]
train_set_sp_80 <- cbind(trainX_sp_80,stress=trainY_sp_80) %>% as.data.frame()
test_set_sp_80 <- cbind(testX_sp_80,stress=testY_sp_80) %>% as.data.frame()

set.seed(1996, sample.kind = "Rounding")
x_80 <- x[which((Phase_A_Results$Material_number %in% folds$Fold1))]
folds_80 <- groupKFold((x_80),k=4)

trControl_sp_80 <- trainControl(method = "cv", number = 4, index = (folds_80))

```
fit_knn_sp_80 <- train(stress ~. ,data =
train_set_sp_80,tuneGrid=expand.grid(k=1:38),method="knn",trControl=trControl_sp_80,preProc=c('ce
nter','scale'))

#I changed foldsFt to folds
Phase_A_Results_80 <- Phase_A_Results[which((Phase_A_Results$Material_number %in%
folds$Fold1)),]

mix_excluded_fold1_80 <- Phase_A_Results_80 %>% filter (!(Material_number %in% folds_80$Fold1))
%>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold2_80 <- Phase_A_Results_80 %>% filter (!(Material_number %in% folds_80$Fold2))
%>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold3_80 <- Phase_A_Results_80 %>% filter (!(Material_number %in% folds_80$Fold3))
%>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold4_80 <- Phase_A_Results_80 %>% filter (!(Material_number %in% folds_80$Fold4))
%>% summarize (unique(Material_number)) %>% as.matrix

trainX_sp_100 <- FSixDaysPredictors_sp
trainY_sp_100 <- FSixDaysOutcome_sp
train_set_sp_100 <- cbind(trainX_sp_100,stress=trainY_sp_100) %>% as.data.frame()

set.seed(1997, sample.kind = "Rounding")
trControl_sp_100 <- trainControl(method = "cv", number = 5, index = (folds))
fit_knn_sp_100 <- train(stress ~. ,data =
train_set_sp_100,tuneGrid=expand.grid(k=1:45),method="knn",trControl=trControl_sp_100,preProc=c('
center','scale'))

set.seed(1997, sample.kind = "Rounding")
x_combined <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear mixing") %>%
select(Material_number)%>% mutate(Material_number=as.numeric(Material_number))%>%
as.data.frame()
x_combined <- x_combined$Material_number
folds_combined <- groupKFold(x_combined,k=8)

#double check the test and train folds prior to further grouping
mix_excluded_fold1_100_combined <- Phase_A_Results_combined %>% filter (!(Material_number
%in% folds_combined$Fold1)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold2_100_combined <- Phase_A_Results_combined %>% filter (!(Material_number
%in% folds_combined$Fold2)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold3_100_combined <- Phase_A_Results_combined %>% filter (!(Material_number
%in% folds_combined$Fold3)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold4_100_combined <- Phase_A_Results_combined %>% filter (!(Material_number
%in% folds_combined$Fold4)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold5_100_combined <- Phase_A_Results_combined %>% filter (!(Material_number
%in% folds_combined$Fold5)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold6_100_combined <- Phase_A_Results_combined %>% filter (!(Material_number
%in% folds_combined$Fold6)) %>% summarize (unique(Material_number)) %>% as.matrix
```

```
mix_excluded_fold7_100_combined <- Phase_A_Results_combined %>% filter (!(Material_number
%in% folds_combined$Fold7)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold8_100_combined <- Phase_A_Results_combined %>% filter (!(Material_number
%in% folds_combined$Fold8)) %>% summarize (unique(Material_number)) %>% as.matrix

inTrain_sp_80_combined <- which((Phase_A_Results_combined$Material_number %in%
folds_combined$Fold3))
trainX_sp_80_combined <- FSixDaysPredictors_sp_combined[inTrain_sp_80_combined,]
trainY_sp_80_combined <- FSixDaysOutcome_sp_combined[inTrain_sp_80_combined]
testX_sp_80_combined <- FSixDaysPredictors_sp_combined[-inTrain_sp_80_combined,]
testY_sp_80_combined <- FSixDaysOutcome_sp_combined[-inTrain_sp_80_combined]
train_set_sp_80_combined <- cbind(trainX_sp_80_combined,stress=trainY_sp_80_combined) %>%
as.data.frame()
test_set_sp_80_combined <- cbind(testX_sp_80_combined,stress=testY_sp_80_combined) %>%
as.data.frame()

set.seed(1996, sample.kind = "Rounding")
x_80_combined <- x_combined[which(Phase_A_Results_combined$Material_number %in%
folds_combined$Fold3)]
folds_80_combined <- groupKFold((x_80_combined),k=6)

trControl_80_combined <- trainControl(method = "cv", number = 8, index = (folds_80_combined))
fit_knn_sp_80_combined <- train(stress ~. ,data =
train_set_sp_80_combined,tuneGrid=expand.grid(k=1:80),method="knn",trControl=trControl_80_comb
ined,preProc=c('center','scale'))

Phase_A_Results_80_combined <-
Phase_A_Results_combined[which((Phase_A_Results_combined$Material_number %in%
folds_combined$Fold3)),]

mix_excluded_fold1_80_combined <- Phase_A_Results_80_combined %>% filter (!(Material_number
%in% folds_80_combined$Fold1)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold2_80_combined <- Phase_A_Results_80_combined %>% filter (!(Material_number
%in% folds_80_combined$Fold2)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold3_80_combined <- Phase_A_Results_80_combined %>% filter(!(Material_number
%in% folds_80_combined$Fold3)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold4_80_combined <- Phase_A_Results_80_combined %>% filter(!(Material_number
%in% folds_80_combined$Fold4)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold5_80_combined <- Phase_A_Results_80_combined %>% filter(!(Material_number
%in% folds_80_combined$Fold5)) %>% summarize (unique(Material_number)) %>% as.matrix
mix_excluded_fold6_80_combined <- Phase_A_Results_80_combined %>% filter(!(Material_number
%in% folds_80_combined$Fold6)) %>% summarize (unique(Material_number)) %>% as.matrix

trainX_sp_100_combined <- FSixDaysPredictors_sp_combined
trainY_sp_100_combined <- FSixDaysOutcome_sp_combined
train_set_sp_100_combined <- cbind(trainX_sp_100_combined,stress=trainY_sp_100_combined) %>%
as.data.frame()
```

```
set.seed(1996, sample.kind = "Rounding")
trControl_sp_100_combined <- trainControl(method = "cv", number = 8, index = (folds_combined))
fit_knn_sp_100_combined <- train(stress ~. ,data =
train_set_sp_100_combined,tuneGrid=expand.grid(k=1:45),method="knn",trControl=trControl_sp_100_
combined,preProc=c('center','scale'))
```

Evaluate knn model generated by fitting matrix with individual stress results

```{r Evaluate knn model generated by fitting matrix with individual stress results, echo=FALSE,
fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

options(digits = 3)
knn_sp_methods <-
c("fit_knn_sp_80","fit_knn_sp_100","fit_knn_sp_80_combined","fit_knn_sp_100_combined")
best_tunes_sp <-
c(as.numeric(fit_knn_sp_80$bestTune),as.numeric(fit_knn_sp_100$bestTune),as.numeric(fit_knn_sp_8
0_combined$bestTune),as.numeric(fit_knn_sp_100_combined$bestTune))
knn_sp_train_rmse <-
c(min(fit_knn_sp_80$results$RMSE),min(fit_knn_sp_100$results$RMSE),min(fit_knn_sp_80_combined$
results$RMSE),min(fit_knn_sp_100_combined$results$RMSE))

pred_knn_sp_80 <- predict(fit_knn_sp_80,newdata = test_set_sp_80)
pred_knn_sp_80_combined <- predict(fit_knn_sp_80_combined,newdata = test_set_sp_80_combined)

pred_knn_sp_80_O <- predict(fit_knn_sp_80,newdata = test_set_O)
pred_knn_sp_100_O <- predict(fit_knn_sp_100,newdata = test_set_O)
pred_knn_sp_80_combined_O <- predict(fit_knn_sp_80_combined,newdata = test_set_O)
pred_knn_sp_100_combined_O <- predict(fit_knn_sp_100_combined,newdata = test_set_O)

knn_sp_test_rmse <-
c(RMSE(pred_knn_sp_80,test_set_sp_80$stress),NA,RMSE(pred_knn_sp_80_combined,test_set_sp_80_
combined$stress),NA)

knn_sp_O_rmse <-
c(RMSE(pred_knn_sp_80_O,test_set_O$stress),RMSE(pred_knn_sp_100_O,test_set_O$stress),RMSE(pr
ed_knn_sp_80_combined_O,test_set_O$stress),RMSE(pred_knn_sp_100_combined_O,test_set_O$stre
ss))

knn_sp_methods_rmse_summary <-
data.frame(method=knn_sp_methods,k=best_tunes_sp,train_rmse=knn_sp_train_rmse,test_rmse=knn
_sp_test_rmse,optimum_test_rmse=knn_sp_O_rmse)

knn_sp_methods_rmse_summary
plot(fit_knn_sp_80)
plot(fit_knn_sp_100)
plot(fit_knn_sp_80_combined)
```

```
plot(fit_knn_sp_100_combined)
varImp(fit_knn_sp_80)
varImp(fit_knn_sp_100)
varImp(fit_knn_sp_80_combined)
varImp(fit_knn_sp_100_combined)


```

Evaluate RMSE on test data knn sp

```{r Evaluate RMSE on test data with individual results, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

plot(pred_knn_sp_80,test_set_sp_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_knn_sp_80_combined,test_set_sp_80_combined$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)

plot(pred_knn_sp_80_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_knn_sp_100_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_knn_sp_80_combined_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_knn_sp_100_combined_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)

```

Generating the predictive vector for the expanded matrix knn sp

```{r Generating the predictive vector for the expanded matrix knn sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysYhat_knn_sp_80 <- predict(fit_knn_sp_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_knn_sp_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_knn_sp_80)
FSixDaysTrialResults_knn_sp_80 <- FSixDaysTrialResults_knn_sp_80 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_knn_sp_100 <- predict(fit_knn_sp_100,ExpandedPredictorsMatrix)
FSixDaysTrialResults_knn_sp_100 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_knn_sp_100)
FSixDaysTrialResults_knn_sp_100 <- FSixDaysTrialResults_knn_sp_100 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_knn_sp_80_combined <- predict(fit_knn_sp_80_combined,ExpandedPredictorsMatrix)
```

```
FSixDaysTrialResults_knn_sp_80_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_knn_sp_80_combined)
FSixDaysTrialResults_knn_sp_80_combined <- FSixDaysTrialResults_knn_sp_80_combined %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_knn_sp_100_combined <- predict(fit_knn_sp_100_combined,ExpandedPredictorsMatrix)
FSixDaysTrialResults_knn_sp_100_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_knn_sp_100_combined)
FSixDaysTrialResults_knn_sp_100_combined <- FSixDaysTrialResults_knn_sp_100_combined %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

```

Performance Density Diagram - faceted by Slag Content knn sp

```{r Performance Density Diagram - faceted by Slag Content knn sp, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_knn_sp_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_knn_sp_100 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_knn_sp_80_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_knn_sp_100_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

```

Performance Density Diagram - faceted by Fly Ash Content knn sp

```{r Performance Density Diagram - faceted by Fly Ash Content knn sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}
FSixDaysTrialResults_knn_sp_80 %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_knn_sp_100 %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_knn_sp_80_combined %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_knn_sp_100_combined %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_knn_sp_100_combined %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)
```

Performance Density Diagram - faceted by Microsilica Content knn sp

```{r Performance Density Diagram - faceted by Microsilica Content knn sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}
FSixDaysTrialResults_knn_sp_80 %>% filter(Microsilica %in% c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],ExpandedPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>% ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller = label_both)

FSixDaysTrialResults_knn_sp_100 %>% filter(Microsilica %in% c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
```

```
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller = label_both)

FSixDaysTrialResults_knn_sp_80_combined %>% filter(Microsilica %in% c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],ExpandedPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller = label_both)

FSixDaysTrialResults_knn_sp_100_combined %>% filter(Microsilica %in% c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],ExpandedPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller = label_both)
```

Optimum mix knn sp

```{r Optimum mix  knn sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

optimum_knn_sp_80 <- as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_knn_sp_80$Expected_Stress_MPa),])
optimum_knn_sp_100 <- as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_knn_sp_100$Expected_Stress_MPa),])
optimum_knn_sp_80_combined <- as.data.frame( ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_knn_sp_80_combined$Expected_Stress_MPa),])
optimum_knn_sp_100_combined <- as.data.frame( ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_knn_sp_100_combined$Expected_Stress_MPa),])

optimum_knn_sp_mixtures <- rbind(optimum_knn_sp_80,optimum_knn_sp_100,optimum_knn_sp_80_combined,optimum_knn_sp_100_combined) %>% as.data.frame()

optimum_knn_sp_max <- c(max(FSixDaysTrialResults_knn_sp_80),max(FSixDaysTrialResults_knn_sp_100),max(FSixDaysTrialResults_knn_sp_80_combined),max(FSixDaysTrialResults_knn_sp_100_combined))

optimum_knn_sp_summary <- data.frame(methods=knn_sp_methods,optimum_knn_sp_mixtures,predicted_stress_MPa=optimum_knn_sp_max)
```

```
optimum_knn_sp_summary

test_set_O

Experimental_mix_rank <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear
mixing")%>% group_by(Material_number)%>%
summarize(Slag=mean(Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAs
h_percent_wt),Stress=mean(Stress_56D_MPa)) %>% arrange(desc(Stress)) %>% as.data.frame() %>%
head(n=10)
Experimental_mix_rank
```

Random Forest method - optimize mtry parameter to fit matrix with average stress results per Mix

```{r Random Forest method - optimize mtry parameter to fit matrix with average stress results per Mix, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

#FSixDaysPredictors <- Phase_A_Results %>% filter(Mixing_Process=="High shear mixing")%>%
group_by(Material_number)%>% summarize(Stress=mean(Stress_56D_MPa),Slag=mean(
Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAsh_percent_wt),wcm=m
ean(w_cm)) %>% select(Slag,Microsilica,FlyAsh,wcm) %>% as.matrix()
#FSixDaysOutcome <- Phase_A_Results %>% filter(Mixing_Process=="High shear mixing")%>%
group_by(Material_number)%>% summarize(stress=mean(Stress_56D_MPa)) %>% select(stress) %>%
as.matrix()
#FSixDaysOutcome <- as.numeric(FSixDaysOutcome)
#inTrain <- createDataPartition(FSixDaysOutcome,p=1,list = FALSE)
#trainX <- FSixDaysPredictors[inTrain,]
#trainY <- FSixDaysOutcome[inTrain]
#testX <- FSixDaysPredictors[-inTrain,]
#testY <- FSixDaysOutcome[-inTrain]
#train_set <- cbind(trainX,stress=trainY) %>% as.data.frame()
#test_set <- cbind(testX,stress=testY) %>% as.data.frame()

set.seed(1997, sample.kind = "Rounding")
tuning <- data.frame(mtry = c(1:100))

fit_rf_avg_80 <- train(stress ~. ,data =
train_set_80,tuneGrid=tuning,method="rf",preProc=c('center','scale'))

fit_rf_avg_100 <- train(stress ~. ,data =
train_set_100,tuneGrid=tuning,method="rf",preProc=c('center','scale'))

fit_rf_avg_80_combined <- train(stress ~. ,data =
train_set_80_combined,tuneGrid=tuning,method="rf",preProc=c('center','scale'))

fit_rf_avg_final <- train(stress ~. ,data =
train_set_final,tuneGrid=tuning,method="rf",preProc=c('center','scale'))
```

```
fit_rf_avg_100_combined <- train(stress ~. ,data =
train_set_100_combined,tuneGrid=tuning,method="rf",preProc=c('center','scale'))

```

Evaluate random forest model generated by fitting matrix with average stress results

```{r Evaluate random forest model generated by fitting matrix with average stress results, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

options(digits = 3)
rf_avg_methods <-
c("fit_rf_avg_80","fit_rf_avg_100","fit_rf_avg_80_combined","fit_rf_avg_final","fit_rf_avg_100_combin
ed")
best_tunes_rf <-
c(as.numeric(fit_rf_avg_80$bestTune),as.numeric(fit_rf_avg_100$bestTune),as.numeric(fit_rf_avg_80_c
ombined$bestTune),as.numeric(fit_rf_avg_final$bestTune),as.numeric(fit_rf_avg_100_combined$bestT
une))
rf_avg_train_rmse <-
c(min(fit_rf_avg_80$results$RMSE),min(fit_rf_avg_100$results$RMSE),min(fit_rf_avg_80_combined$re
sults$RMSE),min(fit_rf_avg_final$results$RMSE),min(fit_rf_avg_100_combined$results$RMSE))

pred_rf_avg_80 <- predict(fit_rf_avg_80,newdata = test_set_80)
pred_rf_avg_80_combined <- predict(fit_rf_avg_80_combined,newdata = test_set_80_combined)
pred_rf_avg_final <- predict(fit_rf_avg_final,newdata = test_set_final)

pred_rf_avg_80_O <- predict(fit_rf_avg_80,newdata = test_set_O)
pred_rf_avg_100_O <- predict(fit_rf_avg_100,newdata = test_set_O)
pred_rf_avg_80_combined_O <- predict(fit_rf_avg_80_combined,newdata = test_set_O)
pred_rf_avg_final_O <- predict(fit_rf_avg_final,newdata = test_set_O)
pred_rf_avg_100_combined_O <- predict(fit_rf_avg_100_combined,newdata = test_set_O)

rf_avg_test_rmse <-
c(RMSE(pred_rf_avg_80,test_set_80$stress),NA,RMSE(pred_rf_avg_80_combined,test_set_80_combine
d$stress),RMSE(pred_rf_avg_final,test_set_final$stress),NA)

rf_avg_O_rmse <-
c(RMSE(pred_rf_avg_80_O,test_set_O$stress),RMSE(pred_rf_avg_100_O,test_set_O$stress),RMSE(pred
_rf_avg_80_combined_O,test_set_O$stress),RMSE(pred_rf_avg_final_O,test_set_O$stress),RMSE(pred
_rf_avg_100_combined_O,test_set_O$stress))

rf_avg_methods_rmse_summary <-
data.frame(method=rf_avg_methods,mtry=best_tunes_rf,train_rmse=rf_avg_train_rmse,test_rmse=rf_
avg_test_rmse,optimum_test_rmse=rf_avg_O_rmse)

rf_avg_methods_rmse_summary
plot(fit_rf_avg_80)
plot(fit_rf_avg_100)
```

```
plot(fit_rf_avg_80_combined)
plot(fit_rf_avg_final)
plot(fit_rf_avg_100_combined)
varImp(fit_rf_avg_80)
varImp(fit_rf_avg_100)
varImp(fit_rf_avg_80_combined)
varImp(fit_rf_avg_final)
varImp(fit_rf_avg_100_combined)
fit_rf_avg_80$finalModel
fit_rf_avg_100$finalModel
fit_rf_avg_80_combined$finalModel
fit_rf_avg_final$finalModel
fit_rf_avg_100_combined$finalModel

```

Evaluate RMSE on test data rf avg

```{r Evaluate RMSE on test data rf avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

plot(pred_rf_avg_80,test_set_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_rf_avg_80_combined,test_set_80_combined$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_rf_avg_final,test_set_final$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)

plot(pred_rf_avg_80_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_rf_avg_100_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_rf_avg_80_combined_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_rf_avg_final_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_rf_avg_100_combined_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)


```

Generating the predictive vector for the expanded matrix rf avg

```{r Generating the predictive vector for the expanded matrix rf avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysYhat_rf_avg_80 <- predict(fit_rf_avg_80,ExpandedPredictorsMatrix)


```

```
FSixDaysTrialResults_rf_avg_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_rf_avg_80)
FSixDaysTrialResults_rf_avg_80 <- FSixDaysTrialResults_rf_avg_80 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_rf_avg_100 <- predict(fit_rf_avg_100,ExpandedPredictorsMatrix)
FSixDaysTrialResults_rf_avg_100 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_rf_avg_100)
FSixDaysTrialResults_rf_avg_100 <- FSixDaysTrialResults_rf_avg_100 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_rf_avg_80_combined <- predict(fit_rf_avg_80_combined,ExpandedPredictorsMatrix)
FSixDaysTrialResults_rf_avg_80_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_rf_avg_80_combined)
FSixDaysTrialResults_rf_avg_80_combined <- FSixDaysTrialResults_rf_avg_80_combined %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_rf_avg_final <- predict(fit_rf_avg_final,ExpandedPredictorsMatrix)
FSixDaysTrialResults_rf_avg_final <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_rf_avg_final)
FSixDaysTrialResults_rf_avg_final <- FSixDaysTrialResults_rf_avg_final %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_rf_avg_100_combined <- predict(fit_rf_avg_100_combined,ExpandedPredictorsMatrix)
FSixDaysTrialResults_rf_avg_100_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_rf_avg_100_combined)
FSixDaysTrialResults_rf_avg_100_combined <- FSixDaysTrialResults_rf_avg_100_combined %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)
```

Performance Density Diagram - faceted by Slag Content rf avg

```{r Performance Density Diagram - faceted by Slag Content rf avg, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_rf_avg_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_rf_avg_100 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)


```

```
FSixDaysTrialResults_rf_avg_80_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_rf_avg_final %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_rf_avg_100_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

```

Performance Density Diagram - faceted by Fly Ash Content rf avg

```{r Performance Density Diagram - faceted by Fly Ash Content rf avg, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_rf_avg_80 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_rf_avg_100 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_rf_avg_80_combined %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)
```

```
FSixDaysTrialResults_rf_avg_final %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_rf_avg_100_combined %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

```

Performance Density Diagram - faceted by Microsilica Content rf avg

```{r Performance Density Diagram - faceted by Microsilica Content rf avg, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_rf_avg_80 %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_rf_avg_100 %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_rf_avg_80_combined %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_rf_avg_final %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)
```

```
FSixDaysTrialResults_rf_avg_100_combined %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)
```

###Optimum mix rf avg

```{r Optimum mix rf avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

optimum_rf_avg_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_rf_avg_80$Expected_Stress_
MPa),])
optimum_rf_avg_100 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_rf_avg_100$Expected_Stress
_MPa),])
optimum_rf_avg_80_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_rf_avg_80_combined$Expected_Stress_MP
a),])
optimum_rf_avg_final <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_rf_avg_final$Expected_Stress
_MPa),])
optimum_rf_avg_100_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_rf_avg_100_combined$Expected_Stress_M
Pa),])

optimum_rf_avg_mixtures <-
rbind(optimum_rf_avg_80,optimum_rf_avg_100,optimum_rf_avg_80_combined,optimum_rf_avg_final,
optimum_rf_avg_100_combined) %>% as.data.frame()

optimum_rf_avg_max <-
c(max(FSixDaysTrialResults_rf_avg_80),max(FSixDaysTrialResults_rf_avg_100),max(FSixDaysTrialResult
_rf_avg_80_combined),max(FSixDaysTrialResults_rf_avg_final),max(FSixDaysTrialResults_rf_avg_100_co
mbined))

optimum_rf_avg_summary <-
data.frame(methods=rf_avg_methods,optimum_rf_avg_mixtures,predicted_stress_MPa=optimum_rf_a
vg_max)
optimum_rf_avg_summary

test_set_O

Experimental_mix_rank <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear
mixing")%>% group_by(Material_number)%>%
summarize(Slag=mean(Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAs
```

```
h_percent_wt),Stress=mean(Stress_56D_MPa)) %>% arrange(desc(Stress)) %>% as.data.frame() %>%
head(n=10)
Experimental_mix_rank

```

Random Forest method - optimize mtry parameter to fit matrix with individual specimens

```{r Random Forest method - optimize mtry parameter to fit matrix with individual specimens, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

#set.seed(1997, sample.kind = "Rounding")
#FSixDaysPredictors <- Phase_A_Results %>% filter(Mixing_Process=="High shear mixing") %>%
select(Slag=Slag_percent_wt,Microsilica=Microsilica_percent_wt,FlyAsh=FlyAsh_percent_wt,wcm=w_c
m)%>% as.matrix()
#FSixDaysOutcome <- Phase_A_Results %>% filter(Mixing_Process=="High shear mixing") %>%
#select(stress=Stress_56D_MPa)%>% as.matrix()
#FSixDaysOutcome <- as.numeric(FSixDaysOutcome)
#inTrain <- createDataPartition(FSixDaysOutcome,p=1,list = FALSE)
#trainX <- FSixDaysPredictors[inTrain,]
#trainY <- FSixDaysOutcome[inTrain]
#testX <- FSixDaysPredictors[-inTrain,]
#testY <- FSixDaysOutcome[-inTrain]
#train_set <- cbind(trainX,stress=trainY) %>% as.data.frame()
#test_set <- cbind(testX,stress=testY) %>% as.data.frame()
#tuning <- data.frame(mtry = c(1:100))
#fit_rf_sp <- train(stress ~. ,data = #train_set,tuneGrid=tuning,method="rf",preProc=c('center','scale'))

set.seed(1997, sample.kind = "Rounding")
tuning <- data.frame(mtry = c(1:100))

fit_rf_sp_80 <- train(stress ~. ,data =
train_set_sp_80,tuneGrid=tuning,method="rf",preProc=c('center','scale'))

fit_rf_sp_100 <- train(stress ~. ,data =
train_set_sp_100,tuneGrid=tuning,method="rf",preProc=c('center','scale'))

fit_rf_sp_80_combined <- train(stress ~. ,data =
train_set_sp_80_combined,tuneGrid=tuning,method="rf",preProc=c('center','scale'))

fit_rf_sp_100_combined <- train(stress ~. ,data =
train_set_sp_100_combined,tuneGrid=tuning,method="rf",preProc=c('center','scale'))

```

Evaluate random forest model generated by fitting matrix with individual specimens

```{r Evaluate random forest model generated by fitting matrix with individual specimens, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

options(digits = 3)
rf_sp_methods <- c("fit_rf_sp_80","fit_rf_sp_100","fit_rf_sp_80_combined","fit_rf_sp_100_combined")
best_tunes_rf_sp <-
c(as.numeric(fit_rf_sp_80$bestTune),as.numeric(fit_rf_sp_100$bestTune),as.numeric(fit_rf_sp_80_combined$bestTune),as.numeric(fit_rf_sp_100_combined$bestTune))
rf_sp_train_rmse <-
c(min(fit_rf_sp_80$results$RMSE),min(fit_rf_sp_100$results$RMSE),min(fit_rf_sp_80_combined$results$RMSE),min(fit_rf_sp_100_combined$results$RMSE))

pred_rf_sp_80 <- predict(fit_rf_sp_80,newdata = test_set_sp_80)
pred_rf_sp_80_combined <- predict(fit_rf_sp_80_combined,newdata = test_set_sp_80_combined)

pred_rf_sp_80_O <- predict(fit_rf_sp_80,newdata = test_set_O)
pred_rf_sp_100_O <- predict(fit_rf_sp_100,newdata = test_set_O)
pred_rf_sp_80_combined_O <- predict(fit_rf_sp_80_combined,newdata = test_set_O)
pred_rf_sp_100_combined_O <- predict(fit_rf_sp_100_combined,newdata = test_set_O)

rf_sp_test_rmse <-
c(RMSE(pred_rf_sp_80,test_set_sp_80$stress),NA,RMSE(pred_rf_sp_80_combined,test_set_sp_80_combined$stress),NA)

rf_sp_O_rmse <-
c(RMSE(pred_rf_sp_80_O,test_set_O$stress),RMSE(pred_rf_sp_100_O,test_set_O$stress),RMSE(pred_rf_sp_80_combined_O,test_set_O$stress),RMSE(pred_rf_sp_100_combined_O,test_set_O$stress))

rf_sp_methods_rmse_summary <-
data.frame(method=rf_sp_methods,mtry=best_tunes_rf_sp,train_rmse=rf_sp_train_rmse,test_rmse=rf_sp_test_rmse,optimum_test_rmse=rf_sp_O_rmse)

rf_sp_methods_rmse_summary
plot(fit_rf_sp_80)
plot(fit_rf_sp_100)
plot(fit_rf_sp_80_combined)
plot(fit_rf_sp_100_combined)
varImp(fit_rf_sp_80)
varImp(fit_rf_sp_100)
varImp(fit_rf_sp_80_combined)
varImp(fit_rf_sp_100_combined)
fit_rf_sp_80$finalModel
fit_rf_sp_100$finalModel
fit_rf_sp_80_combined$finalModel
fit_rf_sp_100_combined$finalModel
```

```

### Evaluate RMSE on test data rf sp

```{r Evaluate RMSE on test data rf sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

plot(pred_rf_sp_80,test_set_sp_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_rf_sp_80_combined,test_set_sp_80_combined$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)

plot(pred_rf_sp_80_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_rf_sp_100_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_rf_sp_80_combined_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)
plot(pred_rf_sp_100_combined_O,test_set_O$stress,ylim = c(0,160),xlim=c(0,160))
abline(0,1)

```

Generating the predictive vector for the expanded matrix rf sp

```{r Generating the predictive vector for the expanded matrix rf sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysYhat_rf_sp_80 <- predict(fit_rf_sp_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_rf_sp_80 <- ExpandedPredictorsMatrix %>% mutate(y_hat=FSixDaysYhat_rf_sp_80)
FSixDaysTrialResults_rf_sp_80 <- FSixDaysTrialResults_rf_sp_80 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_rf_sp_100 <- predict(fit_rf_sp_100,ExpandedPredictorsMatrix)
FSixDaysTrialResults_rf_sp_100 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_rf_sp_100)
FSixDaysTrialResults_rf_sp_100 <- FSixDaysTrialResults_rf_sp_100 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_rf_sp_80_combined <- predict(fit_rf_sp_80_combined,ExpandedPredictorsMatrix)
FSixDaysTrialResults_rf_sp_80_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_rf_sp_80_combined)
FSixDaysTrialResults_rf_sp_80_combined <- FSixDaysTrialResults_rf_sp_80_combined %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_rf_sp_100_combined <- predict(fit_rf_sp_100_combined,ExpandedPredictorsMatrix)
FSixDaysTrialResults_rf_sp_100_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_rf_sp_100_combined)
```
```

```
FSixDaysTrialResults_rf_sp_100_combined <- FSixDaysTrialResults_rf_sp_100_combined %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)
```

Performance Density Diagram - faceted by Slag Content rf sp

```{r Performance Density Diagram - faceted by Slag Content rf sp, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_rf_sp_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_rf_sp_100 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_rf_sp_80_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_rf_sp_100_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)
```

Performance Density Diagram - faceted by Fly Ash Content rf sp

```{r Performance Density Diagram - faceted by Fly Ash Content rf sp, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_rf_sp_80 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)
```

```
FSixDaysTrialResults_rf_sp_100 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_rf_sp_80_combined %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_rf_sp_100_combined %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)
```

Performance Density Diagram - faceted by Microsilica Content rf sp

```{r Performance Density Diagram - faceted by Microsilica Content rf sp, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_rf_sp_80 %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_rf_sp_100 %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_rf_sp_80_combined %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_rf_sp_100_combined %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
```

```
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller = label_both)
```

###Optimum mix rf sp

```{r Optimum mix rf sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

optimum_rf_sp_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_rf_sp_80$Expected_Stress_MPa),])
optimum_rf_sp_100 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_rf_sp_100$Expected_Stress_MPa),])
optimum_rf_sp_80_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_rf_sp_80_combined$Expected_Stress_MPa),])
optimum_rf_sp_100_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_rf_sp_100_combined$Expected_Stress_MPa),])

optimum_rf_sp_mixtures <-
rbind(optimum_rf_sp_80,optimum_rf_sp_100,optimum_rf_sp_80_combined,optimum_rf_sp_100_combined) %>% as.data.frame()

optimum_rf_sp_max <-
c(max(FSixDaysTrialResults_rf_sp_80),max(FSixDaysTrialResults_rf_sp_100),max(FSixDaysTrialResults_rf_sp_80_combined),max(FSixDaysTrialResults_rf_sp_100_combined))

optimum_rf_sp_summary <-
data.frame(methods=rf_sp_methods,optimum_rf_sp_mixtures,predicted_stress_MPa=optimum_rf_sp_max)
optimum_rf_sp_summary

test_set_O

Experimental_mix_rank <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear mixing")%>% group_by(Material_number)%>%
summarize(Slag=mean(Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAsh_percent_wt),Stress=mean(Stress_56D_MPa)) %>% arrange(desc(Stress)) %>% as.data.frame() %>%
head(n=10)
Experimental_mix_rank

```

Linear models to fit matrix with average stress results per Mix

```{r Linear models to fit matrix with average stress results per Mix, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

set.seed(1997, sample.kind = "Rounding")
grid <- expand.grid(span = 5, degree = 2)

fit_loess_avg_80 <- train(stress ~. ,data = train_set_80,method="gamLoess",tuneGrid=grid, preProc=c('center','scale'))

fit_glm_avg_80 <- train(stress ~. ,data = train_set_80,method="glm",preProc=c('center','scale'))

fit_lm_poly1_avg_80 <- train_set_80 %>% lm(stress~poly(Slag+Microsilica+FlyAsh, degree=1, raw=TRUE), data = .)
fit_lm_poly2_avg_80 <- train_set_80 %>% lm(stress~poly(Slag+Microsilica+FlyAsh, degree=2, raw=TRUE), data = .)
fit_lm_poly3_avg_80 <- train_set_80 %>% lm(stress~poly(Slag+Microsilica+FlyAsh, degree=3, raw=TRUE), data = .)

```

Evaluate linear models by fitting matrix with average stress results

```{r Evaluate linear models by fitting matrix with average stress results, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

options(digits = 3)
linearmodels_avg_methods_80 <- c("fit_loess_avg_80","fit_glm_avg_80","fit_lm_poly1_avg_80","fit_lm_poly2_avg_80","fit_lm_poly3_avg_80")

linearmodels_avg_train_rmse_80 <- c(min(fit_loess_avg_80$results$RMSE),min(fit_glm_avg_80$results$RMSE),RMSE((predict(fit_lm_poly1_avg_80,train_set_80)),train_set_80$stress),RMSE((predict(fit_lm_poly2_avg_80,train_set_80)),train_set_80$stress),RMSE((predict(fit_lm_poly3_avg_80,train_set_80)),train_set_80$stress))

pred_loess_avg_80 <- predict(fit_loess_avg_80,newdata = test_set_80)
pred_glm_avg_80 <- predict(fit_glm_avg_80,newdata = test_set_80)
pred_lm_poly1_avg_80 <- predict(fit_lm_poly1_avg_80,newdata = test_set_80)
pred_lm_poly2_avg_80 <- predict(fit_lm_poly2_avg_80,newdata = test_set_80)
pred_lm_poly3_avg_80 <- predict(fit_lm_poly3_avg_80,newdata = test_set_80)

linearmodels_avg_test_rmse_80 <- c(RMSE(pred_loess_avg_80,test_set_80$stress),RMSE(pred_glm_avg_80,test_set_80$stress),RMSE(pred_lm_poly1_avg_80,test_set_80$stress),RMSE(pred_lm_poly2_avg_80,test_set_80$stress),RMSE(pred_lm_poly3_avg_80,test_set_80$stress))
```

```
linearmodels_avg_methods_rmse_summary_80 <-
data.frame(method=linearmodels_avg_methods_80,train_rmse=linearmodels_avg_train_rmse_80,test_
rmse=linearmodels_avg_test_rmse_80)

linearmodels_avg_methods_rmse_summary_80

```

Evaluate RMSE on test data linear models avg

```{r Evaluate RMSE on test data linear models avg, echo=FALSE, fig.height=8, fig.width=16,
message=FALSE, warning=FALSE}

plot(pred_glm_avg_80,test_set_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_loess_avg_80,test_set_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_lm_poly1_avg_80,test_set_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_lm_poly2_avg_80,test_set_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_lm_poly3_avg_80,test_set_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)

```

Generating the predictive vector for the expanded matrix linear models avg

```{r Generating the predictive vector for the expanded matrix linear models avg, echo=FALSE,
fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysYhat_glm_avg_80 <- predict(fit_glm_avg_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_glm_avg_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_glm_avg_80)
FSixDaysTrialResults_glm_avg_80 <- FSixDaysTrialResults_glm_avg_80 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_loess_avg_80 <- predict(fit_loess_avg_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_loess_avg_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_loess_avg_80)
FSixDaysTrialResults_loess_avg_80 <- FSixDaysTrialResults_loess_avg_80 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_lm_poly1_avg_80 <- predict(fit_lm_poly1_avg_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_lm_poly1_avg_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_lm_poly1_avg_80)
FSixDaysTrialResults_lm_poly1_avg_80 <- FSixDaysTrialResults_lm_poly1_avg_80 %>% as.data.frame()
%>% rename(Expected_Stress_MPa=y_hat)
```

```
FSixDaysYhat_lm_poly2_avg_80 <- predict(fit_lm_poly2_avg_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_lm_poly2_avg_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_lm_poly2_avg_80)
FSixDaysTrialResults_lm_poly2_avg_80 <- FSixDaysTrialResults_lm_poly2_avg_80 %>% as.data.frame()
%>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_lm_poly3_avg_80 <- predict(fit_lm_poly3_avg_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_lm_poly3_avg_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_lm_poly3_avg_80)
FSixDaysTrialResults_lm_poly3_avg_80 <- FSixDaysTrialResults_lm_poly3_avg_80 %>% as.data.frame()
%>% rename(Expected_Stress_MPa=y_hat)
```

Performance Density Diagram - faceted by FlyAsh Content linear models avg

```{r Performance Density Diagram - faceted by FlyAsh Content linear models avg, echo=FALSE,
fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_glm_avg_80 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,140))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_loess_avg_80 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,140))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_lm_poly1_avg_80 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,140))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_lm_poly2_avg_80 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,140))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_lm_poly3_avg_80 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,140))+facet_wrap(.~FlyAsh,labeller =
label_both)
```

```

###Optimum mix linear models avg

```{r Optimum mix linear models avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

optimum_glm_avg_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_glm_avg_80$Expected_Stress_MPa),])
optimum_loess_avg_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_loess_avg_80$Expected_Stress_MPa),])
optimum_lm_poly1_avg_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_lm_poly1_avg_80$Expected_Stress_MPa),])
optimum_lm_poly2_avg_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_lm_poly2_avg_80$Expected_Stress_MPa),])
optimum_lm_poly3_avg_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_lm_poly3_avg_80$Expected_Stress_MPa),])

optimum_linearmodels_avg_mixtures_80 <-
rbind(optimum_glm_avg_80,optimum_loess_avg_80,optimum_lm_poly1_avg_80,optimum_lm_poly2_avg_80,optimum_lm_poly3_avg_80) %>% as.data.frame()

optimum_linearmodels_avg_max_80 <-
c(max(FSixDaysTrialResults_glm_avg_80),max(FSixDaysTrialResults_loess_avg_80),max(FSixDaysTrialResults_lm_poly1_avg_80),max(FSixDaysTrialResults_lm_poly2_avg_80),max(FSixDaysTrialResults_lm_poly3_avg_80))

optimum_linearmodels_avg_summary_80 <-
data.frame(methods=linearmodels_avg_methods_80,optimum_linearmodels_avg_mixtures_80,predicted_stress_MPa=optimum_linearmodels_avg_max_80)
optimum_linearmodels_avg_summary_80

Experimental_mix_rank

```

linear models - optimize mtry parameter to fit matrix with individual specimens

```{r linear models to fit matrix with individual specimens, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

set.seed(1997, sample.kind = "Rounding")
grid <- expand.grid(span = 5, degree = 2)

```
fit_loess_sp_80 <- train(stress ~. ,data = train_set_sp_80,method="gamLoess",tuneGrid=grid,
preProc=c('center','scale'))

fit_glm_sp_80 <- train(stress ~. ,data = train_set_sp_80,method="glm",preProc=c('center','scale'))

fit_lm_poly1_sp_80 <- train_set_sp_80 %>% lm(stress~poly(Slag+Microsilica+FlyAsh, degree=1,
raw=TRUE), data = .)
fit_lm_poly2_sp_80 <- train_set_sp_80 %>% lm(stress~poly(Slag+Microsilica+FlyAsh, degree=2,
raw=TRUE), data = .)
fit_lm_poly3_sp_80 <- train_set_sp_80 %>% lm(stress~poly(Slag+Microsilica+FlyAsh, degree=3,
raw=TRUE), data = .)

```

### Evaluate linear models generated by fitting matrix with individual specimens

```{r Evaluate linear models generated by fitting matrix with individual specimens, echo=FALSE,
fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

options(digits = 3)
linearmodels_sp_methods_80 <-
c("fit_loess_sp_80","fit_glm_sp_80","fit_lm_poly1_sp_80","fit_lm_poly2_sp_80","fit_lm_poly3_sp_80")

linearmodels_sp_train_rmse_80 <-
c(min(fit_loess_sp_80$results$RMSE),min(fit_glm_sp_80$results$RMSE),RMSE((predict(fit_lm_poly1_sp
_80,train_set_sp_80)),train_set_sp_80$stress),RMSE((predict(fit_lm_poly2_sp_80,train_set_sp_80)),trai
n_set_sp_80$stress),RMSE((predict(fit_lm_poly3_sp_80,train_set_sp_80)),train_set_sp_80$stress))

pred_loess_sp_80 <- predict(fit_loess_sp_80,newdata = test_set_sp_80)
pred_glm_sp_80 <- predict(fit_glm_sp_80,newdata = test_set_sp_80)
pred_lm_poly1_sp_80 <- predict(fit_lm_poly1_sp_80,newdata = test_set_sp_80)
pred_lm_poly2_sp_80 <- predict(fit_lm_poly2_sp_80,newdata = test_set_sp_80)
pred_lm_poly3_sp_80 <- predict(fit_lm_poly3_sp_80,newdata = test_set_sp_80)

linearmodels_sp_test_rmse_80 <-
c(RMSE(pred_loess_sp_80,test_set_sp_80$stress),RMSE(pred_glm_sp_80,test_set_sp_80$stress),RMSE
(pred_lm_poly1_sp_80,test_set_sp_80$stress),RMSE(pred_lm_poly2_sp_80,test_set_sp_80$stress),RM
SE(pred_lm_poly3_sp_80,test_set_sp_80$stress))

linearmodels_sp_methods_rmse_summary_80 <-
data.frame(method=linearmodels_sp_methods_80,train_rmse=linearmodels_sp_train_rmse_80,test_r
mse=linearmodels_sp_test_rmse_80)

linearmodels_sp_methods_rmse_summary_80

```

Evaluate RMSE on test data linear sp

```{r Evaluate RMSE on test data linear sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

plot(pred_glm_sp_80,test_set_sp_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_loess_sp_80,test_set_sp_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_lm_poly1_sp_80,test_set_sp_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_lm_poly2_sp_80,test_set_sp_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)
plot(pred_lm_poly3_sp_80,test_set_sp_80$stress,ylim = c(0,160),xlim = c(0,160))
abline(0,1)

```

Generating the predictive vector for the expanded matrix linear sp

```{r Generating the predictive vector for the expanded matrix linear sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysYhat_glm_sp_80 <- predict(fit_glm_sp_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_glm_sp_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_glm_sp_80)
FSixDaysTrialResults_glm_sp_80 <- FSixDaysTrialResults_glm_sp_80 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_loess_sp_80 <- predict(fit_loess_sp_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_loess_sp_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_loess_sp_80)
FSixDaysTrialResults_loess_sp_80 <- FSixDaysTrialResults_loess_sp_80 %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_lm_poly1_sp_80 <- predict(fit_lm_poly1_sp_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_lm_poly1_sp_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_lm_poly1_sp_80)
FSixDaysTrialResults_lm_poly1_sp_80 <- FSixDaysTrialResults_lm_poly1_sp_80 %>% as.data.frame()
%>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_lm_poly2_sp_80 <- predict(fit_lm_poly2_sp_80,ExpandedPredictorsMatrix)
FSixDaysTrialResults_lm_poly2_sp_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_lm_poly2_sp_80)
FSixDaysTrialResults_lm_poly2_sp_80 <- FSixDaysTrialResults_lm_poly2_sp_80 %>% as.data.frame()
%>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_lm_poly3_sp_80 <- predict(fit_lm_poly3_sp_80,ExpandedPredictorsMatrix)
```

```
FSixDaysTrialResults_lm_poly3_sp_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_lm_poly3_sp_80)
FSixDaysTrialResults_lm_poly3_sp_80 <- FSixDaysTrialResults_lm_poly3_sp_80 %>% as.data.frame()
%>% rename(Expected_Stress_MPa=y_hat)

```

Performance Density Diagram - faceted by Slag Content linear models sp

```{r Performance Density Diagram - faceted by Slag Content linear models sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_glm_sp_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_loess_sp_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_lm_poly1_sp_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_lm_poly2_sp_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_lm_poly3_sp_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

```

###Optimum mix linear models sp

```{r Optimum mix linear models sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

optimum_glm_sp_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_glm_sp_80$Expected_Stress_MPa),])
optimum_loess_sp_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_loess_sp_80$Expected_Stress_MPa),])
optimum_lm_poly1_sp_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_lm_poly1_sp_80$Expected_Stress_MPa),])
optimum_lm_poly2_sp_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_lm_poly2_sp_80$Expected_Stress_MPa),])
optimum_lm_poly3_sp_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_lm_poly3_sp_80$Expected_Stress_MPa),])

optimum_linearmodels_sp_mixtures_80 <-
rbind(optimum_glm_sp_80,optimum_loess_sp_80,optimum_lm_poly1_sp_80,optimum_lm_poly2_sp_80,optimum_lm_poly3_sp_80) %>% as.data.frame()

optimum_linearmodels_sp_max_80 <-
c(max(FSixDaysTrialResults_glm_sp_80),max(FSixDaysTrialResults_loess_sp_80),max(FSixDaysTrialResults_lm_poly1_sp_80),max(FSixDaysTrialResults_lm_poly2_sp_80),max(FSixDaysTrialResults_lm_poly3_sp_80))

optimum_linearmodels_sp_summary_80 <-
data.frame(methods=linearmodels_sp_methods_80,optimum_linearmodels_sp_mixtures_80,predicted_stress_MPa=optimum_linearmodels_sp_max_80)
optimum_linearmodels_sp_summary_80

Experimental_mix_rank

```

Generating the predictive vector for the expanded matrix ENSEMBLE (knn+rf) avg

```{r Generating the predictive vector for the expanded matrix ENSEMBLE, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysYhat_ensemble_avg_80 <-
data.frame(KNN=FSixDaysYhat_knn_avg_80,RF=FSixDaysYhat_rf_avg_80)
a <- rowMeans(FSixDaysYhat_ensemble_avg_80)
FSixDaysYhat_ensemble_avg_80 <- FSixDaysYhat_ensemble_avg_80 %>% mutate(ENSEMBLE=a)
```

```
FSixDaysTrialResults_ensemble_avg_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_avg_80$ENSEMBLE)
FSixDaysTrialResults_ensemble_avg_80 <- FSixDaysTrialResults_ensemble_avg_80 %>% as.data.frame()
%>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_ensemble_avg_100 <-
data.frame(KNN=FSixDaysYhat_knn_avg_100,RF=FSixDaysYhat_rf_avg_100)
a <- rowMeans(FSixDaysYhat_ensemble_avg_100)
FSixDaysYhat_ensemble_avg_100 <- FSixDaysYhat_ensemble_avg_100 %>% mutate(ENSEMBLE=a)
FSixDaysTrialResults_ensemble_avg_100 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_avg_100$ENSEMBLE)
FSixDaysTrialResults_ensemble_avg_100 <- FSixDaysTrialResults_ensemble_avg_100 %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_ensemble_avg_80_combined <-
data.frame(KNN=FSixDaysYhat_knn_avg_80_combined,RF=FSixDaysYhat_rf_avg_80_combined)
a <- rowMeans(FSixDaysYhat_ensemble_avg_80_combined)
FSixDaysYhat_ensemble_avg_80_combined <- FSixDaysYhat_ensemble_avg_80_combined %>%
mutate(ENSEMBLE=a)
FSixDaysTrialResults_ensemble_avg_80_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_avg_80_combined$ENSEMBLE)
FSixDaysTrialResults_ensemble_avg_80_combined <- FSixDaysTrialResults_ensemble_avg_80_combined
%>% as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_ensemble_avg_final <-
data.frame(KNN=FSixDaysYhat_knn_avg_final,RF=FSixDaysYhat_rf_avg_final)
a <- rowMeans(FSixDaysYhat_ensemble_avg_final)
FSixDaysYhat_ensemble_avg_final <- FSixDaysYhat_ensemble_avg_final %>% mutate(ENSEMBLE=a)
FSixDaysTrialResults_ensemble_avg_final <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_avg_final$ENSEMBLE)
FSixDaysTrialResults_ensemble_avg_final <- FSixDaysTrialResults_ensemble_avg_final %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_ensemble_avg_100_combined <-
data.frame(KNN=FSixDaysYhat_knn_avg_100_combined,RF=FSixDaysYhat_rf_avg_100_combined)
a <- rowMeans(FSixDaysYhat_ensemble_avg_100_combined)
FSixDaysYhat_ensemble_avg_100_combined <- FSixDaysYhat_ensemble_avg_100_combined %>%
mutate(ENSEMBLE=a)
FSixDaysTrialResults_ensemble_avg_100_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_avg_100_combined$ENSEMBLE)
FSixDaysTrialResults_ensemble_avg_100_combined <-
FSixDaysTrialResults_ensemble_avg_100_combined %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_ensemble_optimum <-
data.frame(KNN=FSixDaysYhat_knn_sp_100_combined,RF=FSixDaysYhat_rf_avg_100_combined)
a <- rowMeans(FSixDaysYhat_ensemble_optimum)
FSixDaysYhat_ensemble_optimum <- FSixDaysYhat_ensemble_optimum %>% mutate(ENSEMBLE=a)
```

```
FSixDaysTrialResults_ensemble_optimum <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_optimum$ENSEMBLE)
FSixDaysTrialResults_ensemble_optimum <- FSixDaysTrialResults_ensemble_optimum %>%
as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

```

Performance Density Diagram - faceted by Slag Content ENSEMBLE avg

```{r Performance Density Diagram - faceted by Slag Content ENSEMBLE avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_ensemble_avg_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_ensemble_avg_100 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_ensemble_avg_80_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_ensemble_avg_final %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_ensemble_avg_100_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

```

Performance Density Diagram - faceted by Fly Ash Content ENSEMBLE avg

```{r Performance Density Diagram - faceted by Fly Ash Content ENSEMBLE avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_ensemble_avg_80 %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_ensemble_avg_100 %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_ensemble_avg_80_combined %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_ensemble_avg_final %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_ensemble_avg_100_combined %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

FSixDaysTrialResults_ensemble_optimum %>% filter(FlyAsh %in% c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>% ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller = label_both)

```

Performance Density Diagram - faceted by Microsilica Content ENSEMBLE avg

```{r Performance Density Diagram - faceted by Microsilica Content ENSEMBLE avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}
```

```
FSixDaysTrialResults_ensemble_avg_80 %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_ensemble_avg_100 %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_ensemble_avg_80_combined %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_ensemble_avg_final %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_ensemble_avg_100_combined %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)
```

Optimum mix ENSEMBLE avg

```{r Optimum mix ENSEMBLE avg, echo=FALSE, fig.height=8, fig.width=16, message=FALSE,
warning=FALSE}

optimum_ensemble_avg_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_ensemble_avg_80$Expected
_Stress_MPa),])
optimum_ensemble_avg_100 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_ensemble_avg_100$Expecte
d_Stress_MPa),])
```

```
optimum_ensemble_avg_80_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_ensemble_avg_80_combined$Expected_Str
ess_MPa),])
optimum_ensemble_avg_final <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_ensemble_avg_final$Expecte
d_Stress_MPa),])
optimum_ensemble_avg_100_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_ensemble_avg_100_combined$Expected_St
ress_MPa),])

optimum_ensemble_avg_mixtures <-
rbind(optimum_ensemble_avg_80,optimum_ensemble_avg_100,optimum_ensemble_avg_80_combine
d,optimum_ensemble_avg_final,optimum_ensemble_avg_100_combined) %>% as.data.frame()

optimum_ensemble_avg_max <-
c(max(FSixDaysTrialResults_ensemble_avg_80),max(FSixDaysTrialResults_ensemble_avg_100),max(FSix
DaysTrialResults_ensemble_avg_80_combined),max(FSixDaysTrialResults_ensemble_avg_final),max(FSi
xDaysTrialResults_ensemble_avg_100_combined))

ensemble_avg_methods <-
c("fit_ensemble_avg_80","fit_ensemble_avg_100","fit_ensemble_avg_80_combined","fit_ensemble_av
g_final","fit_ensemble_avg_100_combined")

optimum_ensemble_avg_summary <-
data.frame(methods=ensemble_avg_methods,optimum_ensemble_avg_mixtures,predicted_stress_MP
a=optimum_ensemble_avg_max)
optimum_ensemble_avg_summary

Experimental_mix_rank <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear
mixing")%>% group_by(Material_number)%>%
summarize(Slag=mean(Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAs
h_percent_wt),Stress=mean(Stress_56D_MPa)) %>% arrange(desc(Stress)) %>% as.data.frame() %>%
head(n=10)
Experimental_mix_rank

```

Generating the predictive vector for the expanded matrix ENSEMBLE (knn+rf)sp

```{r Generating the predictive vector for the expanded matrix ENSEMBLE sp, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysYhat_ensemble_sp_80 <-
data.frame(KNN=FSixDaysYhat_knn_sp_80,RF=FSixDaysYhat_rf_sp_80)
a <- rowMeans(FSixDaysYhat_ensemble_sp_80)
FSixDaysYhat_ensemble_sp_80 <- FSixDaysYhat_ensemble_sp_80 %>% mutate(ENSEMBLE=a)
FSixDaysTrialResults_ensemble_sp_80 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_sp_80$ENSEMBLE)
```

```
FSixDaysTrialResults_ensemble_sp_80 <- FSixDaysTrialResults_ensemble_sp_80 %>% as.data.frame()
%>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_ensemble_sp_100 <-
data.frame(KNN=FSixDaysYhat_knn_sp_100,RF=FSixDaysYhat_rf_sp_100)
a <- rowMeans(FSixDaysYhat_ensemble_sp_100)
FSixDaysYhat_ensemble_sp_100 <- FSixDaysYhat_ensemble_sp_100 %>% mutate(ENSEMBLE=a)
FSixDaysTrialResults_ensemble_sp_100 <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_sp_100$ENSEMBLE)
FSixDaysTrialResults_ensemble_sp_100 <- FSixDaysTrialResults_ensemble_sp_100 %>% as.data.frame()
%>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_ensemble_sp_80_combined <-
data.frame(KNN=FSixDaysYhat_knn_sp_80_combined,RF=FSixDaysYhat_rf_sp_80_combined)
a <- rowMeans(FSixDaysYhat_ensemble_sp_80_combined)
FSixDaysYhat_ensemble_sp_80_combined <- FSixDaysYhat_ensemble_sp_80_combined %>%
mutate(ENSEMBLE=a)
FSixDaysTrialResults_ensemble_sp_80_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_sp_80_combined$ENSEMBLE)
FSixDaysTrialResults_ensemble_sp_80_combined <- FSixDaysTrialResults_ensemble_sp_80_combined
%>% as.data.frame() %>% rename(Expected_Stress_MPa=y_hat)

FSixDaysYhat_ensemble_sp_100_combined <-
data.frame(KNN=FSixDaysYhat_knn_sp_100_combined,RF=FSixDaysYhat_rf_sp_100_combined)
a <- rowMeans(FSixDaysYhat_ensemble_sp_100_combined)
FSixDaysYhat_ensemble_sp_100_combined <- FSixDaysYhat_ensemble_sp_100_combined %>%
mutate(ENSEMBLE=a)
FSixDaysTrialResults_ensemble_sp_100_combined <- ExpandedPredictorsMatrix %>%
mutate(y_hat=FSixDaysYhat_ensemble_sp_100_combined$ENSEMBLE)
FSixDaysTrialResults_ensemble_sp_100_combined <-
FSixDaysTrialResults_ensemble_sp_100_combined %>% as.data.frame() %>%
rename(Expected_Stress_MPa=y_hat)

```
```

Performance Density Diagram - faceted by Slag Content ENSEMBLE sp

```{r Performance Density Diagram - faceted by Slag Content ENSEMBLE sp, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_ensemble_sp_80 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)
```
```

```
FSixDaysTrialResults_ensemble_sp_100 %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_ensemble_sp_80_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

FSixDaysTrialResults_ensemble_sp_100_combined %>% filter(Slag %in%
c(0,ExpandedPredictorsMatrix$Slag[26],ExpandedPredictorsMatrix$Slag[51],ExpandedPredictorsMatrix$
Slag[76],60)) %>%
ggplot(aes(Microsilica,FlyAsh,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale
_fill_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Slag,labeller =
label_both)

```

Performance Density Diagram - faceted by Fly Ash Content ENSEMBLE sp

```{r Performance Density Diagram - faceted by Fly Ash Content ENSEMBLE sp, echo=FALSE, fig.height=8,
fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_ensemble_sp_80 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_ensemble_sp_100 %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_ensemble_sp_80_combined %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

FSixDaysTrialResults_ensemble_sp_100_combined %>% filter(FlyAsh %in%
c(0,FlyAshPredictor[14,],FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],15)) %>%
```

```
ggplot(aes(Microsilica,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fil
l_gradientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~FlyAsh,labeller =
label_both)

```

Performance Density Diagram - faceted by Microsilica Content ENSEMBLE sp

```{r Performance Density Diagram - faceted by Microsilica Content ENSEMBLE sp, echo=FALSE,
fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_ensemble_sp_80 %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_ensemble_sp_100 %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_ensemble_sp_80_combined %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

FSixDaysTrialResults_ensemble_sp_100_combined %>% filter(Microsilica %in%
c(0,ExpandedPredictorsMatrix$Microsilica[2510],ExpandedPredictorsMatrix$Microsilica[5010],Expande
dPredictorsMatrix$Microsilica[7510],ExpandedPredictorsMatrix$Microsilica[10000])) %>%
ggplot(aes(FlyAsh,Slag,z=Expected_Stress_MPa,fill=Expected_Stress_MPa))+geom_raster()+scale_fill_gr
adientn(colors=c("#F8766D","white","#00BFC4"),limits=c(70,130))+facet_wrap(.~Microsilica,labeller =
label_both)

```

###Optimum mix ENSEMBLE sp

```{r Optimum mix ENSEMBLE sp, echo=FALSE, fig.height=8, fig.width=16, message=FALSE,
warning=FALSE}
```

```
optimum_ensemble_sp_80 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_ensemble_sp_80$Expected_S
tress_MPa),])
optimum_ensemble_sp_100 <-
as.data.frame(ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_ensemble_sp_100$Expected
_Stress_MPa),])
optimum_ensemble_sp_80_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_ensemble_sp_80_combined$Expected_Stre
ss_MPa),])
optimum_ensemble_sp_100_combined <- as.data.frame(
ExpandedPredictorsMatrix[which.max(FSixDaysTrialResults_ensemble_sp_100_combined$Expected_Str
ess_MPa),])

optimum_ensemble_sp_mixtures <-
rbind(optimum_ensemble_sp_80,optimum_ensemble_sp_100,optimum_ensemble_sp_80_combined,o
ptimum_ensemble_sp_100_combined) %>% as.data.frame()

optimum_ensemble_sp_max <-
c(max(FSixDaysTrialResults_ensemble_sp_80),max(FSixDaysTrialResults_ensemble_sp_100),max(FSixDa
ysTrialResults_ensemble_sp_80_combined),max(FSixDaysTrialResults_ensemble_sp_100_combined))

ensemble_sp_methods <-
c("fit_ensemble_sp_80","fit_ensemble_sp_100","fit_ensemble_sp_80_combined","fit_ensemble_sp_10
0_combined")

optimum_ensemble_sp_summary <-
data.frame(methods=ensemble_sp_methods,optimum_ensemble_sp_mixtures,predicted_stress_MPa=
optimum_ensemble_sp_max)
optimum_ensemble_sp_summary

Experimental_mix_rank <- Phase_A_Results_combined %>% filter(Mixing_Process=="High shear
mixing")%>% group_by(Material_number)%>%
summarize(Slag=mean(Slag_percent_wt),Microsilica=mean(Microsilica_percent_wt),FlyAsh=mean(FlyAs
h_percent_wt),Stress=mean(Stress_56D_MPa)) %>% arrange(desc(Stress)) %>% as.data.frame() %>%
head(n=10)

Optimum_mixtures_summary <-
rbind(optimum_knn_avg_summary,optimum_rf_avg_summary,optimum_ensemble_avg_summary,opti
mum_knn_sp_summary,optimum_rf_sp_summary,optimum_ensemble_sp_summary)

Optimum_mixtures_summary
Experimental_mix_rank

```

3D Evaluation of fitting models (individual specimens)

```{r 3D Evaluation of knn model fitting the averaged results vs fitting the individual specimens, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

FSixDaysTrialResults_knn_avg_100_combined_3D_FA0 <- FSixDaysTrialResults_knn_avg_100_combined %>% filter(FlyAsh==0)
FSixDaysTrialResults_knn_sp_100_combined_3D_FA0 <- FSixDaysTrialResults_knn_sp_100_combined %>% filter(FlyAsh==0)
FSixDaysTrialResults_knn_avg_100_combined_3D_FA5 <- FSixDaysTrialResults_knn_avg_100_combined %>% filter(FlyAsh==5)
FSixDaysTrialResults_knn_sp_100_combined_3D_FA5 <- FSixDaysTrialResults_knn_sp_100_combined %>% filter(FlyAsh==5)
FSixDaysTrialResults_knn_avg_100_combined_3D_FA10 <- FSixDaysTrialResults_knn_avg_100_combined %>% filter(FlyAsh==10)
FSixDaysTrialResults_knn_sp_100_combined_3D_FA10 <- FSixDaysTrialResults_knn_sp_100_combined %>% filter(FlyAsh==10)
FSixDaysTrialResults_knn_avg_100_combined_3D_FA15 <- FSixDaysTrialResults_knn_avg_100_combined %>% filter(FlyAsh==15)
FSixDaysTrialResults_knn_sp_100_combined_3D_FA15 <- FSixDaysTrialResults_knn_sp_100_combined %>% filter(FlyAsh==15)

FSixDaysTrialResults_rf_avg_100_combined_3D_FA0 <- FSixDaysTrialResults_rf_avg_100_combined %>% filter(FlyAsh==0)
FSixDaysTrialResults_rf_sp_100_combined_3D_FA0 <- FSixDaysTrialResults_rf_sp_100_combined %>% filter(FlyAsh==0)
FSixDaysTrialResults_rf_avg_100_combined_3D_FA5 <- FSixDaysTrialResults_rf_avg_100_combined %>% filter(FlyAsh==5)
FSixDaysTrialResults_rf_sp_100_combined_3D_FA5 <- FSixDaysTrialResults_rf_sp_100_combined %>% filter(FlyAsh==5)
FSixDaysTrialResults_rf_avg_100_combined_3D_FA10 <- FSixDaysTrialResults_rf_avg_100_combined %>% filter(FlyAsh==10)
FSixDaysTrialResults_rf_sp_100_combined_3D_FA10 <- FSixDaysTrialResults_rf_sp_100_combined %>% filter(FlyAsh==10)
FSixDaysTrialResults_rf_avg_100_combined_3D_FA15 <- FSixDaysTrialResults_rf_avg_100_combined %>% filter(FlyAsh==15)
FSixDaysTrialResults_rf_sp_100_combined_3D_FA15 <- FSixDaysTrialResults_rf_sp_100_combined %>% filter(FlyAsh==15)

FSixDaysTrialResults_ensemble_optimum_3D_FA0 <- FSixDaysTrialResults_ensemble_optimum %>% filter(FlyAsh==0)
FSixDaysTrialResults_ensemble_optimum_3D_FA3 <- FSixDaysTrialResults_ensemble_optimum %>% filter(FlyAsh==FSixDaysTrialResults_ensemble_optimum[24,])
FSixDaysTrialResults_knn_sp_100_combined_3D_FA3 <- FSixDaysTrialResults_knn_sp_100_combined %>% filter(FlyAsh==FSixDaysTrialResults_knn_sp_100_combined[24,])
FSixDaysTrialResults_rf_avg_100_combined_3D_FA3 <- FSixDaysTrialResults_rf_avg_100_combined %>% filter(FlyAsh==FSixDaysTrialResults_knn_sp_100_combined[24,])
```

```
#KNN
scatter3D(FSixDaysTrialResults_knn_sp_100_combined_3D_FA3$Microsilica,
FSixDaysTrialResults_knn_sp_100_combined_3D_FA3$Slag,
      FSixDaysTrialResults_knn_sp_100_combined_3D_FA3$Expected_Stress_MPa, clab = "Predicted
Stress (MPa)",
      pch = 200, cex = 0.2, bty = "b2",
      col = ramp.col(c("red", "orange", "green")), theta = 155, phi = 10,
      main = "Fly Ash = 3.5 %", xlab = "% Microsilica",
      ylab ="% Slag", zlab = "Stress (MPa)", ticktype = "detailed",zlim=c(70,130),
      colkey = list(side = 1, length = 0.25))

scatter3D(FSixDaysTrialResults_knn_sp_100_combined_3D_FA0$Microsilica,
FSixDaysTrialResults_knn_sp_100_combined_3D_FA0$Slag,
FSixDaysTrialResults_knn_sp_100_combined_3D_FA0$Expected_Stress_MPa, clab = "Predicted Stress
(MPa)",
       pch = 200, cex = 0.5, bty = "b2",type="h",
       col = ramp.col(c("red", "orange", "green")), theta = 45, phi = 10,
       main = "Fly Ash = 0 %", xlab = "% Microsilica",
       ylab ="% Slag", zlab = "Stress (MPa)", ticktype = "detailed",zlim=c(70,130),
       colkey = list(side = 1, length = 0.25))

#RF
scatter3D(FSixDaysTrialResults_rf_avg_100_combined_3D_FA3$Microsilica,
FSixDaysTrialResults_rf_avg_100_combined_3D_FA3$Slag,
      FSixDaysTrialResults_rf_avg_100_combined_3D_FA3$Expected_Stress_MPa, clab = "Predicted
Stress (MPa)",
      pch = 200, cex = 0.2, bty = "b2",
      col = ramp.col(c("red", "orange", "green")), theta = 135, phi = 10,
      main = "Fly Ash = 3.5 %", xlab = "% Microsilica",
      ylab ="% Slag", zlab = "Stress (MPa)", ticktype = "detailed",zlim=c(70,130),
      colkey = list(side = 1, length = 0.25))

scatter3D(FSixDaysTrialResults_rf_sp_100_combined_3D_FA0$Microsilica,
FSixDaysTrialResults_rf_sp_100_combined_3D_FA0$Slag,
FSixDaysTrialResults_rf_sp_100_combined_3D_FA0$Expected_Stress_MPa, clab = "Predicted Stress
(MPa)",
       pch = 200, cex = 0.5, bty = "b2",type="h",
       col = ramp.col(c("red", "orange", "green")), theta = 45, phi = 10,
       main = "Fly Ash = 0 %", xlab = "% Microsilica",
       ylab ="% Slag", zlab = "Stress (MPa)", ticktype = "detailed",zlim=c(70,130),
       colkey = list(side = 1, length = 0.25))

#ensemble

scatter3D(FSixDaysTrialResults_ensemble_optimum_3D_FA3$Microsilica,
FSixDaysTrialResults_ensemble_optimum_3D_FA3$Slag,
      FSixDaysTrialResults_ensemble_optimum_3D_FA3$Expected_Stress_MPa, clab = "Predicted Stress
(MPa)",
```

```
        pch = 200, cex = 0.2, bty = "b2",
        col = ramp.col(c("red", "orange", "green")), theta = 135, phi = 10,
        main = "Fly Ash = 3.5 %", xlab = "% Microsilica",
        ylab ="% Slag", zlab = "Stress (MPa)", ticktype = "detailed",zlim=c(70,130),
        colkey = list(side = 1, length = 0.25))

```

###CEDDs

```{r load data and plot CEDDs, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

Phase_A_CDD <- read_excel("D:/PhD - papers in progress/Emission Density Diagrams Paper/computations/phase A - CDD values.xlsx")
Phase_A_CDD <- data.frame(Phase_A_CDD,stringsAsFactors = TRUE) %>% rename(TotalCost=Total_Cost_per_m3)

index_CDD <- which(FSixDaysTrialResults_ensemble_optimum$FlyAsh %in% c(0,FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],FlyAshPredictor[84,],15))

Phase_A_CDD[index_CDD,] %>% ggplot(aes(Microsilica,Slag,z=TotalCost,fill=TotalCost))+geom_raster()+scale_fill_gradientn(colors=c("#00BFC4","white","#F8766D"),limits=c(min(Phase_A_CDD$TotalCost),max(Phase_A_CDD$TotalCost)))+facet_wrap(.~FlyAsh,labeller = label_both)

#p_axial

Phase_A_CDD[index_CDD,] %>% ggplot(aes(Microsilica,Slag,z=p_axial,fill=p_axial))+geom_raster()+scale_fill_gradientn(colors=c("#00BFC4","white","#F8766D"),limits=c(min(Phase_A_CDD$p_axial),max(Phase_A_CDD$p_axial)))+facet_wrap(.~FlyAsh,labeller = label_both)

Phase_A_CDD[which.min(Phase_A_CDD$p_axial),]
Phase_A_CDD[which.min(Phase_A_CDD$TotalCost),]
Phase_A_CDD %>% filter (p_axial<Phase_A_CDD[which.min(Phase_A_CDD$TotalCost),]$p_axial) %>%
  arrange(desc(TotalCost))%>% as.data.frame() %>% head(n=1)
Phase_A_CDD[which.min(FSixDaysTrialResults_ensemble_optimum$Expected_Stress_MPa),]

#p_bending

Phase_A_CDD[index_CDD,] %>% ggplot(aes(Microsilica,Slag,z=p_bending,fill=p_bending))+geom_raster()+scale_fill_gradientn(colors=c("#00BFC4","white","#F8766D"),limits=c(min(Phase_A_CDD$p_bending),max(Phase_A_CDD$p_bending)))+facet_wrap(.~FlyAsh,labeller = label_both)

Phase_A_CDD[which.min(Phase_A_CDD$p_bending),]
Phase_A_CDD[which.min(Phase_A_CDD$TotalCost),]
```

```
Phase_A_CDD %>% filter
(p_bending<(Phase_A_CDD[which.min(Phase_A_CDD$TotalCost),]$p_bending)+0.1) %>%
  arrange(desc(TotalCost))%>% as.data.frame() %>% head(n=1)
Phase_A_CDD[which.min(FSixDaysTrialResults_ensemble_optimum$Expected_Stress_MPa),]

#p_deflection

Phase_A_CDD[index_CDD,] %>%
ggplot(aes(Microsilica,Slag,z=p_deflection,fill=p_deflection))+geom_raster()+scale_fill_gradientn(colors=
c("#00BFC4","white","#F8766D"),limits=c(min(Phase_A_CDD$p_deflection),max(Phase_A_CDD$p_defle
ction)))+facet_wrap(.~FlyAsh,labeller = label_both)

Phase_A_CDD[which.min(Phase_A_CDD$p_deflection),]
Phase_A_CDD[which.min(Phase_A_CDD$TotalCost),]
Phase_A_CDD %>% filter
(p_deflection<(Phase_A_CDD[which.min(Phase_A_CDD$TotalCost),]$p_deflection)+0.1) %>%
  arrange(desc(TotalCost))%>% as.data.frame() %>% head(n=1)
Phase_A_CDD[which.min(FSixDaysTrialResults_ensemble_optimum$Expected_Stress_MPa),]

```

###EEDDs

```{r load data and plot EEDDs, echo=FALSE, fig.height=8, fig.width=16, message=FALSE, warning=FALSE}

Phase_A_EDD <- read_excel("D:/PhD - papers in progress/Emission Density Diagrams
Paper/computations/phase A - GWP values.xlsx")
Phase_A_EDD <- data.frame(Phase_A_EDD,stringsAsFactors = TRUE)%>%
rename(TotalGWP=Total_GWP_per_m3)

index_EDD <- which(FSixDaysTrialResults_ensemble_optimum$FlyAsh %in%
c(0,FlyAshPredictor[26,],FlyAshPredictor[51,],FlyAshPredictor[62,],FlyAshPredictor[84,],15))

Phase_A_EDD[index_EDD,] %>%
ggplot(aes(Microsilica,Slag,z=TotalGWP,fill=TotalGWP))+geom_raster()+scale_fill_gradientn(colors=c("#
009966","white","#F8766D"),limits=c(min(Phase_A_EDD$TotalGWP),max(Phase_A_EDD$TotalGWP)))+f
acet_wrap(.~FlyAsh,labeller = label_both)

#X_axial

Phase_A_EDD[index_EDD,] %>%
ggplot(aes(Microsilica,Slag,z=X_axial,fill=X_axial))+geom_raster()+scale_fill_gradientn(colors=c("#00996
6","white","#F8766D"),limits=c(min(Phase_A_EDD$X_axial),max(Phase_A_EDD$X_axial)))+facet_wrap(.
~FlyAsh,labeller = label_both)

Phase_A_EDD[which.min(Phase_A_EDD$X_axial),]
Phase_A_EDD[which.min(Phase_A_EDD$TotalGWP),]
```

```
Phase_A_EDD %>% filter (X_axial<Phase_A_EDD[which.min(Phase_A_EDD$TotalGWP),]$X_axial+0.01)
%>%
  arrange(desc(TotalGWP))%>% as.data.frame() %>% head(n=1)
Phase_A_EDD[which.min(FSixDaysTrialResults_ensemble_optimum$Expected_Stress_MPa),]

#X_bending

Phase_A_EDD[index_EDD,] %>%
ggplot(aes(Microsilica,Slag,z=X_bending,fill=X_bending))+geom_raster()+scale_fill_gradientn(colors=c("#
009966","white","#F8766D"),limits=c(min(Phase_A_EDD$X_bending),max(Phase_A_EDD$X_bending)))+
facet_wrap(.~FlyAsh,labeller = label_both)

Phase_A_EDD[which.min(Phase_A_EDD$X_bending),]
Phase_A_EDD[which.min(Phase_A_EDD$TotalGWP),]
Phase_A_EDD %>% filter
(X_bending<(Phase_A_EDD[which.min(Phase_A_EDD$TotalGWP),]$X_bending)+0.1) %>%
  arrange(desc(TotalGWP))%>% as.data.frame() %>% head(n=1)
Phase_A_EDD[which.min(FSixDaysTrialResults_ensemble_optimum$Expected_Stress_MPa),]
```

APPENDIX E

Author's Biography

Cesario Sarmento Goncalves Martins e Tavares was born in Lisbon, Portugal, on January 20th, 1989. He graduated with an integrated master's degree from University of Porto, Faculty of Engineering (FEUP) in February 2014 and immediately started working as a civil engineer for Constructora SanJose for nearly 3 years, in the role of assistant site manager. In August 2017, Cesario joins Texas A&M University as a graduate student to pursue his Ph.D. in the civil engineering program. He worked as a graduate research assistant in an NSF project for 3 years, and later worked as a graduate teaching assistant for a junior level course (Mechanics of Materials) for 2 years. During his graduate studies, Cesario published 3 journal articles, presented his research in numerous conferences and received a Graduate Teaching Fellowship to be the main instructor teaching a sophomore level course (Eng. Mechanics: Statics). Following the completion of his Ph.D., Cesario is starting a postdoctoral appointment at Texas A&M University.