


2018

A computational framework for data-driven infrastructure engineering using advanced statistical learning, prediction, and curing

Ikkyun Song
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Civil Engineering Commons](#), and the [Statistics and Probability Commons](#)

Recommended Citation

Song, Ikkyun, "A computational framework for data-driven infrastructure engineering using advanced statistical learning, prediction, and curing" (2018). *Graduate Theses and Dissertations*. 16671.
<https://lib.dr.iastate.edu/etd/16671>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**A computational framework for data-driven infrastructure engineering using
advanced statistical learning, prediction, and curing**

by

Ikkyun Song

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Civil Engineering (Intelligent Infrastructure Engineering)

Program of Study Committee:
In-Ho Cho, Major Professor
Halil Ceylan
Kristen Cetin
An Chen
Jae-Kwang Kim

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2018

Copyright © Ikkyun Song, 2018. All rights reserved.

DEDICATION

To my wife, Hwahyun

For your love, patience, and overwhelming support

To my son, Seonu and my daughter, Ellie

For making me happier and stronger

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES	viii
ACKNOWLEDGMENTS	xiv
ABSTRACT	xvi
CHAPTER 1. INTRODUCTION	1
1.1 General Background	1
1.1.1 Overview of efforts to prevent runway incursions in the US airports	1
1.1.2 Overview of prediction of earthquake engineering data	2
1.1.3 Overview of prediction methods for pavement response and performance	2
1.1.4 Overview of an investigation into the impact of imputation on prediction	3
1.1.5 Overview of bridge health monitoring systems and efforts to utilize the data from these systems	4
1.2 Research Objectives	4
1.3 Research Contributions	5
1.4 Dissertation Organization	6
Bibliography	7
CHAPTER 2. DATA-DRIVEN PREDICTION OF RUNWAY INCURSIONS WITH UN- CERTAINTY QUANTIFICATION	10
2.1 Introduction	10
2.2 Methodology	14
2.2.1 Data collection	14
2.2.2 Data extraction and transformation	16
2.2.3 Advanced statistical model, GAM	17
2.2.4 Metrics for prediction accuracy	19
2.3 Selection of Best GAM Model Using Parallel Computing	20
2.4 Prediction Results	23
2.5 Uncertainty Quantification Procedure	28
2.6 Conclusions	31
2.A Appendix I. Dataset for Current Study	33
2.B Appendix II. Comparison of GAM and Artificial Neural Network	33
2.7 Acknowledgments	33
Bibliography	37

CHAPTER 3. AN ADVANCED STATISTICAL APPROACH TO DATA-DRIVEN EARTH- QUAKE ENGINEERING	39
3.1 Introduction	39
3.2 Summary of the Generalized Additive Model	43
3.3 Metrics for Prediction Comparisons	46
3.4 Prediction with GAM	47
3.5 Constructing a Best GAM with a Given Number of Variables	49
3.6 Statistical Prediction VS. High-Precision Computer Simulations	52
3.7 Uncertainty Estimation	55
3.8 Difference from Traditional Statistical Methods	57
3.9 Limitation of Statistical Prediction	59
3.10 <i>R</i> Code for Constructing GAM by Cross-Validation	62
3.11 Remarks on Parallel Processing of <i>R</i> & <i>Rmpi</i> Code	65
3.12 Conclusions	65
3.A Appendix	67
3.13 Acknowledgments	67
Bibliography	70
CHAPTER 4. EFFICIENT VARIABLE SELECTION METHODS FOR ADVANCED STA- TISTICAL LEARNING AND PREDICTION OF RIGID PAVEMENT SYSTEMS	72
4.1 Introduction	73
4.2 Overview of Generalized Additive Model	75
4.3 Description of Pavement Databases for Model Development	76
4.4 Best Predictor Variables for GAM Prediction	78
4.5 Relative Importance of Predictor Variables in GAM Prediction	79
4.6 Efficient Variable Selection Strategy	81
4.6.1 Forward variable selection procedure descriptions	81
4.6.2 Backward variable selection procedure descriptions	84
4.6.3 Comparison of variable selection methods	85
4.7 Impact of Distribution Family on Prediction Performance	86
4.8 Parameter Study: Impact of Spline Base	86
4.9 Conclusions	87
4.10 Acknowledgments	89
Bibliography	90
CHAPTER 5. IMPACTS OF FRACTIONAL HOT-DECK IMPUTATION ON LEARNING AND PREDICTION OF ENGINEERING DATA	92
5.1 Introduction	93
5.2 Theory: Fractional Hot-Deck Imputation	95
5.3 Theory: Statistical Learning and Machine Learning Methods	96
5.3.1 Statistical learning: GAM	96
5.3.2 Recap and settings of the adopted machine learning methods	98
5.4 Materials	102
5.5 Imputation	103

5.6	Impact of FHDI on Statistical and Machine Learning-Based Regression	104
5.6.1	Positive role of FHDI on prediction accuracy improvement	105
5.6.2	Impact of the categorization number	107
5.6.3	Impact of donor numbers	109
5.6.4	Impact of extreme data missing	114
5.7	Conclusions	115
5.8	Acknowledgments	116
	Bibliography	117
CHAPTER 6. A COMPUTATIONAL FRAMEWORK FOR STATISTICAL DATA-CURING AND PREDICTION OF BRIDGE AND TRAFFIC BIG DATA		120
6.1	Introduction	120
6.2	Methodology	122
6.2.1	Data collection	122
6.2.2	Data extraction and transformation	123
6.2.3	Data merging with traffic data	125
6.2.4	Data curing: FHDI	125
6.3	Statistical Learning and Prediction	129
6.3.1	Summary of generalized additive model	129
6.3.2	Excellent performance of GAM compared to SVM and ERT	130
6.3.3	Direct search versus correlation-based predictor selection	131
6.3.4	Prediction of traffic data using bridge sensor data	134
6.4	Remarks on Various Impacts on Prediction Accuracy	134
6.4.1	Impact of data curing on prediction	134
6.4.2	Impact of traffic information on prediction performance	138
6.5	Parallelization Strategy	140
6.6	Conclusions	142
6.7	Acknowledgments	143
	Bibliography	145
CHAPTER 7. CONCLUSIONS		147

LIST OF TABLES

	Page
Table 2.1	Summary of datasets used in the current study 15
Table 2.2	Potential impact (mile) visibility criteria based on METAR board 17
Table 2.3	Comparison of prediction performance between direct search algorithm (proposed herein) and PCA-guided variables (all values are CVE_b/CVE) 22
Table 2.4	Metrics used for best combination of predictor variables (GAM-CRS) 26
Table A2.1	Dataset used in statistical learning and prediction 34
Table A2.2	Dataset used in statistical learning and prediction(<i>Continued</i>) 35
Table B2.1	ANN prediction summary using 10 independent variables 36
Table 3.1	Selection of the best combination of variables for GAM using CRS (p-values in parentheses) 50
Table 3.2	Selection of the best combination of variables for GAM using TPRS (p-values in parentheses) 51
Table 3.3	Predictions <i>without</i> WSH series (F_{max} is normalized by that from experiment) 61
Table 3.4	Predictions <i>with</i> WSH series (F_{max} is normalized by that from experiment) 62
Table 3.5	Description of the stand-alone R code (see Table 3.1 in Appendix) 63
Table 3.6	Description of the parallel version of <i>R&Rmpi</i> code (see Table 3.2 in Appendix) 64
Table A3.1	<i>R</i> code for constructing a best GAM using TPRS (3-variable combination) 68
Table A3.2	<i>Rmpi&R</i> code for constructing a best GAM using TPRS (3-variable combination and 3 slaves) 69
Table 4.1	Variable description of concrete overlay and rigid airport pavements data . . 77

Table 4.2	Selection of the best combination of variables for GAM using CRS (p-values in parentheses)	82
Table 5.1	Summary of datasets used in the current study	103
Table 5.2	Four key steps for FHDI method	105
Table 5.3	Expectation ratio (i.e., expectation $E[.]$ of each attribute in the original full data set divided by that of cured data set by FHDI) with different missing rates (10, 30 and 50%). The appliance energy dataset is used.	106
Table 5.4	Comparison of RMSE values from predictions using datasets that were pre-cured by FHDI or a Naive method	108
Table 5.5	Impact of donor numbers on prediction using the weather dataset with 50% response and GAM	109
Table 6.1	Summary of datasets during the transformation process from the raw data to the final dataset	127
Table 6.2	Summary of predictor and response variable for GAM model	132
Table 6.3	Correlation among variables	133
Table 6.4	Best predictors selected by the direct search method	137

LIST OF FIGURES

		Page
Figure 2.1	Scatter plot of variables: (a) runway incursion versus general aviation operation; (b) runway incursion versus general aviation operation and high visibility	12
Figure 2.2	Workflow of runway incursion (RI) prediction using GAM: raw data is collected from various databases and transformed into suitable forms of dataset, with which GAM learns and predicts future RI on high performance computing (HPC)	13
Figure 2.3	Example of thin plate spline basis function using 2 covariates	19
Figure 2.4	Cyclic allocation of the proposed parallel code of <i>R</i> & <i>Rmpi</i> ; two-variable case is shown with "nv" meaning the total number of variables. Height of box corresponds computation load	21
Figure 2.5	Parallel computing performance of <i>R</i> & <i>Rmpi</i> code for finding the 7-variable combination out of 3,432 total combinations	23
Figure 2.6	Biplot from principle component analysis (PCA): (a) entire biplot; (b) part of biplot denoted by dashed box "b"; (c) by "c"; and (d) by "d"	24
Figure 2.7	Pseudo code for finding the best combination of predictor variables	25
Figure 2.8	Comparison of performance between CRS and TPRS on this study: (a) ratio of $CVE_b = CVE$; (b) Pearson correlation; (c) coefficient of determination	26
Figure 2.9	Illustration of cross validation: (a) shows that the first airport's data is omitted, a GAM is constructed by learning all other airport data; (b) shows the same procedure by omitting the second airport data	27

Figure 2.10	(a) Q-Q plot of real-world measured data and predicted data; (b) residuals plot showing that residuals are evenly scattered	27
Figure 2.11	GCV score with varying smoothing parameter. (λ^* = automatically optimized value)	28
Figure 2.12	Confidence interval of smoothing functions of five predictors: (a) the number of taxi operations; (b) the number of general aviation operations; (c) hour of high visibility impact; (d) hour of slight visibility impact; (e) hour of sum of visibility impacts	29
Figure 2.13	Confidence interval for GAM prediction points of 36 airports using (a) GAM and (b) multivariate linear regression: vertical bar represents 95% confidence interval, circle represents measured (real) RI number, and "x" mark represents a median value of bootstrap samples; horizontal axis means airport index; table of 36 airport indexes and generated data is presented in Appendixes 2.A and 2.B	31
Figure 3.1	Sparseness and biasness revealed from 470 real experiments of RC shear wall database (collected from <i>NEESHUB</i> , international reports, and literature) . .	40
Figure 3.2	Number of specimens of each type of RCSW (R: rectangular; T: T-shaped; B: Barbell-shaped; I: I-shaped; B-O: Barbell-shaped with opening; etc.: all other types)	41
Figure 3.3	Change in the interpretability of database with increasing dimensionality: (a) two-dimensional (2D) scatter plot of standardized f_y (steel yield strength of longitudinal bars) and F_{max} (maximum shear force); (b) 3D plot of F_{max} , the standardized f_y , and the standardized f'_c (concrete strength). Some axes are unitless due to the standardized values	42
Figure 3.4	Example of one-dimensional regressions of 470 real RC wall data: (a) hb (thickness of boundary element) versus F_{max} ; (b) wall height versus F_{max} .	44
Figure 3.5	Example of thin plate spline basis function using 2 covariates	45

Figure 3.6 Illustration of cross validation: left figure represents that first specimen’s data is omitted. A GAM is constructed by learning all other wall data; right figure shows the same procedure by omitting the second wall data . . . 47

Figure 3.7 Q-Q plot of real experimental data and the predicted value using (a) GAM(CRS); (b) GAM(TPRS). Both axes are unitless owing to the standardized values . . . 49

Figure 3.8 Illustration of cross validation: left figure represents that first specimen’s data is omitted. A GAM is constructed by learning all other wall data; right figure shows the same procedure by omitting the second wall data . . . 53

Figure 3.9 Normalized maximum shear force of experiment (E), VEEL (V), and GAM using TPRS (GT) and CRS (GC) of RW1 and RW2 (Vulcano et al., 1988), and WSH1 through WSH6 (Orakcal and Wallace, 2006). (Note: The value of vertical axis represents the maximum shear force normalized by experimental result; thus, "E" has always one) 54

Figure 3.10 Prediction accuracy comparison between high-precision computational simulation (VEEL) and statistical prediction (GAM) result using WSH series: (Top 6 panels) experimental results cited from Orakcal and Wallace (2006); (Bottom 6 panels) prediction results from VEEL, GAM-TPRS, and GAM-CRS. Note that the maximum force is the comparison target 56

Figure 3.11 95% confidence interval of WSH wall series’ F_{max} estimated from GAM prediction using bootstrap method. Circle and "x" mark represents measured F_{max} and a median value of bootstrap samples, respectively 58

Figure 3.12 Prediction power comparison of GAM against other popular prediction methods 59

Figure 3.13 Scatter plot of rectangular RCSW specimens showing the ranges of database. WSH wall series occupy the boundary of the database 60

Figure 3.14	Cyclic allocation of the proposed parallel code of <i>R</i> & <i>Rmpi</i> . Two-variable case is shown with <i>nv</i> meaning the total number of variables. Height of box corresponds computation load	66
Figure 3.15	Parallel computing performance of <i>R</i> & <i>Rmpi</i> code for finding the best 5-variable combination out of 252 total combinations. "User code" means the time spent on execution of user-defined codes while "Total" means the total elapsed wall clock time of the parallel code (attained from <i>proc.time()</i> of <i>R</i> .	67
Figure 4.1	The number of predictor variables selected by direct search for the most accurate prediction of (a) <i>case 1</i> , (b) <i>case 2</i> , (c) <i>case 3</i> , and (d) <i>case 4</i> of <i>concrete overlay</i> data	79
Figure 4.2	The number of predictor variables selected by direct search for the most accurate prediction of (a) σ_{xx_top} , (b) σ_{xx_bot} , (c) σ_{yy_top} , and (d) σ_{yy_bot} for <i>case M</i> ; (e) σ_{xx_top} , (f) σ_{xx_bot} , (g) σ_{yy_top} , and (h) σ_{yy_bot} for <i>case TM</i> of <i>rigid airport pavements</i> data	80
Figure 4.3	Prediction performances using different variable selection methods: (a) <i>concrete overlay</i> ; (b) <i>rigid airport pavements</i> . DS stands for <i>direct search</i> , AIC(b) for backward selection using AIC, and $p(f,0.05)$ for forward selection using p-value of 0.05, etc	85
Figure 4.4	Prediction performance depending on different distribution families (Gamma, Gaussian, and Poisson), in which m_{xx_top} stands for the maximum tensile stress in the x direction on the top of the slab with the mechanical loading condition and tm_{yy_bot} stands for the maximum tensile stress in the y direction on the bottom of the slab with thermal and mechanical loading, etc	87
Figure 4.5	Impact of number of TPRS base (<i>k</i>) on GAM prediction: (a) <i>case 1</i> ; (b) <i>case 2</i> ; (c) <i>case 3</i> ; (d) <i>case 4</i> of the <i>concrete overlay</i> data	88

Figure 5.1	Impact of categorization numbers on prediction (appliance energy data set is used). 10 to 50% response rates are investigated.	110
Figure 5.2	Impact of categorization numbers on prediction (air quality data set is used). 10 to 50% response rates are investigated.	110
Figure 5.3	Impact of categorization numbers on prediction (phenotype data set is used). 10 to 50% response rates are investigated.	111
Figure 5.4	Impact of categorization numbers on prediction (weather data set is used). 10 to 50% response rates are investigated.	111
Figure 5.5	Impact of donor numbers on prediction (appliance energy data set is used). 10 to 50% response rates are investigated.	112
Figure 5.6	Impact of donor numbers on prediction (air quality data set is used). 10 to 50% response rates are investigated.	112
Figure 5.7	Impact of donor numbers on prediction (phenotype data set is used). 10 to 50% response rates are investigated.	113
Figure 5.8	Impact of donor numbers on prediction (weather data set is used). 10 to 50% response rates are investigated.	113
Figure 5.9	Relationship between coefficient of variance (CV) of RMSE and normalized RMSE from (a) <i>10%-dataset</i> and (b) <i>50%-dataset</i>	114
Figure 5.10	Impact of extreme missing rates on prediction (appliance energy data set is used). 10 to 50% missing rates are investigated.	115
Figure 6.1	Instrumentation plan of sensors of the target bridge	123
Figure 6.2	Flow chart showing data-transformation from raw bridge and traffic data to the final hybrid data set	124
Figure 6.3	Strain history over (a) 10 minutes and (b) 1 minute. <i>Top</i> peak and <i>bottom</i> peak strains are selected outside the range between $+5\mu$ and -5μ from the median strain value	125
Figure 6.4	Histogram of peak strains	126

Figure 6.5 Example of key procedures for FHDI: (a) entire flow chart; (b) original dataset in which the NA stands for a missing value; (c) categorized dataset; (d) cured dataset 128

Figure 6.6 Comparison of prediction performance between GAM and other methods. In vertical axes, the higher value indicates the higher prediction accuracy. (cited from Song et al. (2018b)) 130

Figure 6.7 The comparison of the best predictor selection between the algorithm used in this study and correlation: (a) mean of top peak strains; (b) mean of bottom peak strains; (c) standard deviation of median strain; (d) minimum strain value of bottom peak; (e) maximum strain value of top peak; (f) area 135

Figure 6.8 The number of the best predictors of traffic data prediction: traffic of (a) small car, (b) medium car and (c) large car 136

Figure 6.9 GAM prediction vs. measured value of traffic: (a) small car, (b) medium car and (c) large car 136

Figure 6.10 Comparison of GAM prediction performances using the dataset with and without imputation 139

Figure 6.11 Impact of missing rates on prediction accuracy (cited from Song et al. (2018a)) 139

Figure 6.12 Comparison of GAM prediction performances using the dataset with and without traffic data 140

Figure 6.13 The impact of the inclusion of traffic data on prediction of the *strainMean-Comp* depending on different missing rates 141

Figure 6.14 Job distribution scheme in the parallel computing system. Jobs are evenly distributed to slaves and then the master collects results from slaves and finds the best predictors 141

Figure 6.15 Pseudo code for algorithm of the parallel computing to find the best predictor combination 142

Figure 6.16 Speed-up test using parallel computing 143

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this dissertation. First and foremost, I am very grateful to my advisor, Dr. In-Ho Cho for his guidance, patience and support throughout this research and the writing of this dissertation. His insights and words of encouragement have often inspired me for completing my doctoral program.

My sincere appreciation also goes to my committee members for their efforts and contributions to this work: Dr. Halil Ceylan, Dr. Kristen Cetin, Dr. An Chen, and Dr. Jae-Kwang Kim. Their constructive comments and advices on my research significantly helped me broaden my insights. Additionally, without their reviews of my dissertation, I would not be able to reach the current stage. Thanks also go to Dr. Sunghwan Kim for his sincere advice on my research.

I would like to acknowledge the financial supports from the Partnership to Enhance General Aviation Safety, Accessibility and Sustainability (PEGASAS) Center of Excellence (COE) fellowship program of the Federal Aviation Administration (FAA), the Iowa Highway Research Board (IHRB), and the Iowa Department of Transportation (IA DOT), the Iowa Department of Transportation, Midwest Transportation Center, U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology, and the research funding of the Department of Civil, Construction, and Environmental Engineering of Iowa State University (ISU). The work regarding parallel computing is partially supported by the HPC@ISU equipment at ISU, some of which has been purchased through funding provided by NSF.

I also would like to express my gratitude to Dr. Halil Ceylan, Dr. Kristen Cetin, Dr. Brent Phares, Dr. Anuj Sharma, and Dr. Carolyn J Lawrence-Dill for sharing their valuable data.

Lastly, I would like to thank my brother Jaekyun, his wife Sunae, my nephews Sophie and Kevin, my father Youngchan, and my mother Seokjeom Kang. Without their lovely supports, I would not have been able to complete this work.

ABSTRACT

Over the past few decades, in most science and engineering fields, data-driven research has been becoming a promising next-generation research paradigm due to noticeable advances in computing power and accumulation of valuable databases. Despite this valuable accomplishment, the leveraging of these databases is still in its infancy. To address this issue, this dissertation investigates the following studies that use advanced statistical methods.

The first study aims to develop a computational framework for collecting and transforming data obtained from heterogeneous databases in the Federal Aviation Administration and build a flexible predictive model using a generalized additive model (GAM) to predict runway incursions for 15 years in the top major US 36 airports. Results show that GAM is a powerful method for RI prediction with a high prediction accuracy. A direct search for finding the best predictor variables appears to be superior over the variable selection approach based on a principal component analysis. The prediction power of GAM turns out to be comparable to that of an artificial neural network (ANN).

The second study is to build an accurate predictive model based on earthquake engineering databases. As with the previous study, GAM is adopted as a predictive model. The result shows a promising predictive power of GAM with application to existing reinforced concrete shear wall databases.

The primary objective of the third study is to suggest an efficient predictor variable selection method and provide relative importance among predictor variables using field survey pavement and simulated airport pavement data. Results show that the direct search method always finds the best predictor model, but the method takes a long time depending on the size of data and the variables' dimensions. The results also depict that all variables are not necessary for the best prediction and identify the relative importance of variables selected for the GAM model.

The fourth study deals with the impact of fractional hot-deck imputation (FHDI) on statistical and machine learning and prediction using practical engineering databases. Multiple response rates and internal parameters (i.e., category number and donor number) are investigated regarding the behavior and impacts of FHDI on prediction models. GAM, ANN, support vector machine, and extremely randomized trees are adopted as predictive models. Results show that the FHDI holds a positive impact on the prediction for engineering-based databases. The optimal internal parameters are also suggested to achieve a better prediction accuracy.

The last study aims to offer a systematic computational framework including data collection, transformation, and squashing to develop a prediction model for the structural behavior of the target bridge. Missing values in the bridge data are cured by using the FHDI method to avoid an inaccurate data analysis due to biasness and sparseness of data. Results show that the application of FHDI improves prediction performances.

This dissertation is expected to provide a notable computational framework for data processing, suggest a seamless data curing method, and offer an advanced statistical predictive model based on multiple projects. This novel research approach will help researchers to investigate their databases with a better understanding and build a statistical model with high accuracy according to their knowledge about the data.

CHAPTER 1. INTRODUCTION

The primary goal of this study is to develop a computational framework for data-driven infrastructure engineering using advanced statistical methods. The research suggests a novel data-driven approach to several data from survey and simulation results in the infrastructure-related domain. It also includes the application of an advanced imputation method on engineering databases. This study is expected to help researchers better explore their databases and build an efficient computational framework for data analysis. The following sections address the general background, research objectives, and research contribution.

1.1 General Background

This section briefly addresses previous research on data-driven approach in the civil infrastructure engineering domain. The following subsections present the research efforts and their limitation for each research topic. More details can be found in the next manuscript-based chapters.

1.1.1 Overview of efforts to prevent runway incursions in the US airports

A runway incursion (RI) is a major concern in airports because it can cause severe runway collisions. Much research has been conducted to resolve this issue by developing detection and alert systems (Ludwig, 2007; Schwab and Rost, 1985; Watnick and Ianniello, 1992; Singh and Meier, 2004; Jones et al., 2001; Eggert et al., 2006; Squire et al., 2010; Schnefeld and Miller, 2012). Additionally, some statistical methods have been investigated for RI studies. Wilke et al. (2015) and Johnson et al. (2016) investigated the impact of the geometry of airports on RI occurrence using the best regression model to find the optimal variable combination among geometric-related variables.

Despite the efforts of developing such detection and warning systems, RI occurrence is reported to increase continuously (FAA, 2015). This might be because those attempts provide a practical solution for a specific airport, but such systems seldom identify other possible factors that can cause RI occurrence and have difficulty finding hidden relationships among the factors. The previous statistical approaches also deliver meaningful results, but they still lack general applicability and flexibility. Comprehensive investigations using advanced statistical models are needed to find a novel RI prediction.

1.1.2 Overview of prediction of earthquake engineering data

Due to dramatic advances in computing power, researchers can obtain valuable knowledge from data (Fishman, 1995; Caffisch, 1998; Kamdar et al., 2016). NSF has been constructing comprehensive community-level earthquake databases (Hacker et al., 2011; Rathje et al., 2017). However, the databases have not been actively utilized to improve the predictive ability of earthquake engineering fields. Moreover, the earthquake engineering community learned about hidden issues they were previously unaware after severe earthquake disasters (Song et al., 2012; Park and Chen, 2012).

Real world experiments are indispensable because they can provide in-depth quantitative knowledge about factors for earthquake occurrences, but limited financial resources prohibit researchers from conducting such experiments and elucidating the underlying relationship between salient variables. Furthermore, after successful experiments, there remain substantial uncertainties, and it might be infeasible to cover a full range of structural variables. Therefore, the need of a notable data-driven approach to the existing earthquake databases appears to be indispensable.

1.1.3 Overview of prediction methods for pavement response and performance

The prediction of pavement response and performance is important to establish efficient pavement designs and maintenance plans. Several research efforts have been conducted to find salient factors for the prediction of pavement response and performance using a variety of methods. Salama et al. (2006) and Heba and Assaf (2017) used linear regression models to investigate the complex in-

terplay among variables. The use of machine learning (ML) is also noteworthy. Ceylan et al. (1998, 1999) used an artificial neural network to build a predictive model for the response of a jointed concrete airport pavement. Gopalakrishnan and Kim (2011) adopted a support vector machine to predict hot mix asphalt stiffness. Tabatabaee et al. (2013) developed a two-stage predictive strategy using both neural network and a support vector machine to better predict pavement responses.

Even though ML's prediction performance is considerable, the causal pathway from input to output is unclear, prohibiting researchers from clearly interpreting prediction results. On the other hand, a statistical method can provide a better understanding of the interplay between the input and output because statistical learning and prediction are based on statistical theories and knowledge. This advantage helps researchers to clearly interpret prediction results and better build a predictive model depending on their knowledge about the data. Meanwhile, though a simple linear regression is handy to use, it is not suitable for complex non-linear data. Therefore, the use of an advanced and flexible statistical model is needed to improve prediction accuracies and clearly elucidate the relationship between variables.

1.1.4 Overview of an investigation into the impact of imputation on prediction

Missing data is commonly observed in surveys and experiments. It prohibits researchers from obtaining a trustworthy conclusion from data analysis due to biasness and sparseness of the data. Brown and Kros (2003); Roth (1994) showed that the missing data causes an inaccurate data analysis. An imputation is a popular method to cure missing data. There have been several efforts to investigate the impact of imputation methods on ML regression and classification (Farhangfar et al., 2008; Batista and Monard, 2003; Heltshe et al., 2012; Lin et al., 2017; Wang et al., 2016; Su et al., 2008; Yoo et al., 2017).

However, the impact of fractional hot-deck imputation (FHDI), which is an advanced repeated imputation method comparable to multiple imputation (Rubin, 1987), on statistical and ML regression has been rarely investigated (see detailed advantages of the FHDI in the section 5.1). This investigation is strongly needed for the general application of FHDI in the engineering domain.

1.1.5 Overview of bridge health monitoring systems and efforts to utilize the data from these systems

Due to advances in bridge health monitoring system (BHM), a variety of sensor-measured data has been accumulated (Jang et al., 2010; Ko and Ni, 2005; Li et al., 2004; Ntotsios et al., 2009). Despite this active data collection, the databases have not been actively used to build predictive models for a better bridge management. Li et al. (2003) used a linear regression model to assess the fatigue for a specific bridge, but its general application is challenging.

The issue of missing data is also important. Since bridge data is measured by sensors, the likelihood of missing data appears inevitable due to various causes including sensors malfunctioning and human-induced mistakes.

Therefore, an advanced data curing method and a predictive model are rigorously needed for building an accurate predictive model and its general application for the community-level research.

1.2 Research Objectives

The overall objective of this study is to develop a computational framework for infrastructure databases using advanced statistical methods and parallel computing. The following are specific objectives of this research.

- **Objective 1:** Develop a systematic framework for gathering data from various databases and leverage the generalized additive model (GAM) using parallel computing to predict runway incursions.
- **Objective 2:** Build a novel statistical learning and prediction framework using GAM to predict the maximum shear forces of a rectangular wall database and compare the prediction performance between simulation, GAM, and other ML methods.
- **Objective 3:** Find the best predictor variables in pavement databases and their relative importance in a GAM prediction and offer an alternative efficient approach to find appropriate predictor variables.

- **Objective 4:** Introduce a relatively new imputation method, the fractional hot-deck imputation (FHDI), to a wide range of engineering community, elucidate the impact of FHDI on statistical and ML regressions, and provide an optimal setting for general application.
- **Objective 5:** Develop a computational framework for collecting and transforming bridge sensor and traffic big data, cure the missing data using the FHDI, and investigate the impact of FHDI on the improvement of GAM prediction accuracy.

1.3 Research Contributions

The primary contributions of this study are systematic computational frameworks for data-driven infrastructure engineering using advanced statistical methods including GAM and FHDI. These novel frameworks can help researchers to better investigate their databases and build appropriate predictive models, as well as help stakeholders to make an appropriate decision based on prediction results. This dissertation provides the following contributions.

First, this study develops a computational framework to leverage an advanced statistical model, GAM, to resolve runway incursion (RI) issues. The data used are collected from heterogeneous databases in the Federal Aviation Administration and squeezed into a compact dataset for RI prediction. The parallel computing to find the best predictor variables can help researchers and engineers to obtain prediction results and make a decision quickly. This approach can be generally applicable for the major airports in the United States (US) because the predictive model is built upon a flexible statistical method and the data covers almost all major airports in the US.

Second, the study develops a novel statistical learning and prediction model using GAM for reinforced concrete shear wall databases. Results identify the best predictor variables and reveal the relevant importance of predictor variables in GAM predictions. The prediction performance comparison between high-precision simulation, GAM, and ML can help researchers to choose a suitable prediction method depending on their knowledge, data quality, and expectation.

Third, the research identifies the most significant variables of pavement databases for GAM predictions, which can help the pavement engineering community to understand the complex interplay

among explanatory and response variables, better carry out pavement designs, and make a maintenance plan. The study also offers an efficient variable selection method, enabling stakeholders to choose a suitable method according to their need.

Next, the study introduces an advanced data curing method, the fractional hot-deck imputation (FHDI) to infrastructure engineering communities. Missing data can be cured by the FHDI, enabling researchers to conduct data analysis without biasness and sparseness of data. Optimal settings, for improving the prediction accuracies after FHDI applications, are investigated using several databases. As a result, optimum parameters for FHDI implementations are suggested.

Last, the research provides a computational framework for collecting, transforming, and merging bridge sensor and traffic big data using a parallel computing. Bridge responses are predicted by using the GAM with a direct search method. The study reveals the prediction power of GAM for the bridge sensor database. Since this approach is developed in a systematic manner, other researchers and engineers can utilize this approach for the long-term decision making and strategic planning.

1.4 Dissertation Organization

This dissertation is organized as follows: Chapter 1 summarizes the background of research including literature reviews and their limitation. Chapters 2 through 6 present five published and submitted journal papers. Chapter 7 concludes this dissertation with major findings from each study and possible future research topics.

Bibliography

- Batista, G. E. A. P. A. and Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533.
- Brown, M. L. and Kros, J. F. (2003). Data mining and the impact of missing data. *Industrial Management & Data Systems*, 103(8):611–621.
- Caffisch, R. E. (1998). Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49.
- Ceylan, H., Tutumluer, E., and Barenberg, E. (1999). Artificial neural networks for analyzing concrete airfield pavements serving the boeing b-777 aircraft. *Transportation Research Record: Journal of the Transportation Research Board*, (1684):110–117.
- Ceylan, H., Tutumluer, E., and Barenberg, E. J. (1998). Artificial neural networks as design tools in concrete airfield pavement design. In *Airport Facilities: Innovations for the Next Century. Proceedings of the 25th International Air Transportation Conference. American Society of Civil Engineers*.
- Eggert, J. R., Howes, B. R., Kuffner, M. P., Wilhelmssen, H., and Bernays, D. J. (2006). Operational evaluation of runway status lights. *Lincoln Laboratory Journal*, 16(1):123.
- FAA (2015). National runway safety report 2013-2014.
- Farhangfar, A., Kurgan, L., and Dy, J. (2008). Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41(12):3692–3705.
- Fishman, G. (1995). *Monte Carlo: concepts, algorithms, and applications*. Springer Science & Business Media, New York.
- Gopalakrishnan, K. and Kim, S. (2011). Support vector machines approach to hma stiffness prediction. *Journal of Engineering Mechanics*, 137(2).
- Hacker, T. J., Eigenmann, R., Bagchi, S., Irfanoglu, A., Pujol, S., Catlin, A., and Rathje, E. (2011). The neeshub cyberinfrastructure for earthquake engineering. *Computing in Science & Engineering*, 13(4):67–78.
- Heba, A. and Assaf, G. J. (2017). Road performance prediction model for the libyan road network depending on experts knowledge and current road condition using bayes linear regression. In *International Congress and Exhibition "Sustainable Civil Infrastructures: Innovative Infrastructure Geotechnology"*, pages 153–167. Springer.
- Heltshe, S. L., Lubin, J. H., Koutros, S., Coble, J. B., Ji, B.-T., Alavanja, M. C. R., Blair, A., Sandler, D. P., Hines, C. J., Thomas, K. W., Barker, J., Andreotti, G., Hoppin, J. A., and Beane Freeman, L. E. (2012). Using multiple imputation to assign pesticide use for non-responders in the follow-up questionnaire in the agricultural health study. *J Expos Sci Environ Epidemiol*, 22(4):409–416.

- Jang, S., Jo, H., Cho, S., Mechitov, K., Rice, J. A., Sim, S.-H., Jung, H.-J., Yun, C. B., Spencer Jr, B. F., and Agha, G. (2010). Structural health monitoring of a cable-stayed bridge using smart sensor technology: deployment and evaluation. *Smart Structures and Systems*, 6(5-6):439–459.
- Johnson, M. E., Zhao, X., Faulkner, B., and Young, J. P. (2016). Statistical models of runway incursions based on runway intersections and taxiways. *Journal of Aviation Technology and Engineering*, 5(2):3.
- Jones, D. R., Quach, C. C., and Young, S. D. (2001). Runway incursion prevention system-demonstration and testing at the dallas/fort worth international airport. In *Digital Avionics Systems, 2001. DASC. 20th Conference*, volume 1, pages 2D2/1–2D2/11 vol. 1. IEEE.
- Kamdar, H., Turk, M., and Brunner, R. (2016). Machine learning and cosmological simulations ii. hydrodynamical simulations. *Monthly Notices of the Royal Astronomical Society*, 457(2):11621179.
- Ko, J. and Ni, Y. (2005). Technology developments in structural health monitoring of large-scale bridges. *Engineering structures*, 27(12):1715–1725.
- Li, H.-N., Li, D.-S., and Song, G.-B. (2004). Recent applications of fiber optic sensors to health monitoring in civil engineering. *Engineering structures*, 26(11):1647–1657.
- Li, Z., Chan, T. H., and Zheng, R. (2003). Statistical analysis of online strain response and its application in fatigue assessment of a long-span steel bridge. *Engineering structures*, 25(14):1731–1741.
- Lin, C.-C., Li, C.-I., Liu, C.-S., Lin, W.-Y., Lin, C.-H., Yang, S.-Y., and Li, T.-C. (2017). Development and validation of a risk prediction model for end-stage renal disease in patients with type 2 diabetes. *Scientific Reports*, 7(1):10177.
- Ludwig, D. (2007). Direct alerting to the cockpit for runway incursions. In *Digital Avionics Systems Conference, 2007. DASC'07. IEEE/AIAA 26th*, pages 5. A. 6–1–5. A. 6–10. IEEE.
- Ntotsios, E., Papadimitriou, C., Panetsos, P., Karaiskos, G., Perros, K., and Perdikaris, P. C. (2009). Bridge health monitoring system based on vibration measurements. *Bulletin of Earthquake Engineering*, 7(2):469.
- Park, J. and Chen, Y. (2012). Understanding and improving the seismic design of shear walls. final year projects. Technical report, Dept. of Civil and Natural Resources Engineering, University of Canterbury.
- Rathje, E. M., Dawson, C., Padgett, J. E., Pinelli, J.-P., Stanzione, D., Adair, A., Arduino, P., Brandenberg, S. J., Cockerill, T., and Dey, C. (2017). Designsafes: New cyberinfrastructure for natural hazards engineering. *Natural Hazards Review*, 18(3):06017001.
- Roth, P. L. (1994). Missing data: A conceptual review for applied psychologists. *Personnel Psychology*, 47(3):537–560.
- Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. John Wiley & Sons.

- Salama, H. K., Chatti, K., and Lyles, R. W. (2006). Effect of heavy multiple axle trucks on flexible pavement damage using in-service pavement performance data. *Journal of transportation engineering*, 132(10):763–770.
- Schnefeld, J. and Miller, D. (2012). Runway incursion prevention systems: A review of runway incursion avoidance and alerting system approaches. *Progress in Aerospace Sciences*, 51:31–49.
- Schwab, C. and Rost, D. (1985). Airport surface detection equipment. *Proceedings of the IEEE*, 73(2):290–300.
- Singh, G. and Meier, C. (2004). Preventing runway incursions and conflicts. *Aerospace science and technology*, 8(7):653–670.
- Song, C., Pujol, S., and Lepage, A. (2012). The collapse of the alto ro building during the 27 february 2010 maule, chile, earthquake. *Earthquake Spectra*, 28(S1):S301–S334.
- Squire, P. N., Barrow, J. H., Durkee, K. T., Smith, C. M., Moore, J. C., and Parasuraman, R. (2010). Rimdas: A proposed system for reducing runway incursions. *Ergonomics in Design*, 18(2):10–17.
- Su, X., Khoshgoftaar, T. M., Zhu, X., and Greiner, R. (2008). Imputation-boosted collaborative filtering using machine learning classifiers. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 949–950. ACM.
- Tabatabaee, N., Ziyadi, M., and Shafahi, Y. (2013). Two-stage support vector classifier and recurrent neural network predictor for pavement performance modeling. *Journal of Infrastructure Systems*, 19(3):266–274.
- Wang, J., Tsapakis, I., and Zhong, C. (2016). A spacetime delay neural network model for travel time prediction. *Engineering Applications of Artificial Intelligence*, 52(Supplement C):145–160.
- Watnick, M. and Ianniello, J. W. (1992). Airport movement area safety system. In *11th Digital Avionics Systems Conf., IEEE/AIAA*, pages 549–552. IEEE.
- Wilke, S., Majumdar, A., and Ochieng, W. Y. (2015). The impact of airport characteristics on airport surface accidents and incidents. *Journal of safety research*, 53:63–75.
- Yoo, K. D., Noh, J., Lee, H., Kim, D. K., Lim, C. S., Kim, Y. H., Lee, J. P., Kim, G., and Kim, Y. S. (2017). A machine learning approach using survival statistics to predict graft survival in kidney transplant recipients: A multicenter cohort study. *Scientific Reports*, 7(1):8904.

CHAPTER 2. DATA-DRIVEN PREDICTION OF RUNWAY INCURSIONS WITH UNCERTAINTY QUANTIFICATION

A paper published in *Journal of Computing in Civil Engineering, ASCE*, (2018)

Ikkyun Song, In-Ho Cho, Tom Tessitore, Tony Gurcsik, and Halil Ceylan

Abstract

In 2015 only, more than 1,500 runway incursions (RIs) occurred at US airports, which could result in serious runway collisions. Nonlinear interactions among many factors and complex data structures pose challenges to RI prevention, and reportedly, the annual RI occurrence is gradually increasing. This study seeks to offer a data-driven solution of advanced statistical learning and prediction by leveraging the generalized additive model (GAM). The GAM holds a powerful flexibility with little restriction to many variables over a broad range of modeling distributions. This study proposes a method to systematically obtain, parse, and transform various factors from diverse databases to give rise to interpretable datasets. It also presents high-performance computational procedures to automatically select out salient factors to achieve the best GAM with a strong predictive power. Practical applications to RI of US airports show promising performance. A combination of GAM and bootstrapping method to build confidence intervals is expounded upon as a means to quantify underlying uncertainties.

2.1 Introduction

In 2015 alone, 1,507 runway incursions (RIs) happened at airports in the United States (FAA), which can lead to a runway collision. To resolve this problem, there have been practical efforts to solve the RI issue: e.g., airport movement area safety system (AMASS) (Watnick and Ianniello, 1992), RI alert system (Jones et al., 2001), and RI prevention system (Schnefeld and Miller, 2012).

However, despite the efforts, the occurrence of runway incursion is reported to increase almost every year (FAA, 2015). The attempts in the above literatures provide suitable remedy for the specific airport, but other possible factors causing RI occurrence and hidden relationship among them are not presented and the suggested solutions are not easy to applied in other airports since each airport has many different conditions (e.g., geometry, traffic, weather, and so on) which means a general prediction model needs to be established. To resolve this issue and reduce RI occurrence, reliable solution of future prediction of RIs is imperative.

Wilke et al. (2015) investigated the impact of geometry of airport and causal factors on runway incursion occurrence. Johnson et al. (2016) reported the relationship between geometry of airport and runway incursion occurrence at 63 airports in the United States. They used best subset regression to find the best combination among four geometry variables to predict RI and reported two of them are suitable predictor variables for runway incursion prediction. Their prediction accuracy was, however, not practical for general application as they stated in the paper. For better prediction performance, the deployment of several predictor variables on the more flexible and accurate prediction model is likely to be inevitable.

Several factors have been identified by the previous studies and Federal Aviation Administration (FAA, 2008): poor weather, low visibility, time of day, miscommunication with air traffic control (ATC), and so on. Yet, it is hard to elucidate the quantitative relevance of the factors to RI and their relative importance. Prediction of future RI occurrence is more difficult because of the complex interrelations among the contributing factors. Simple statistical methods such as linear regression would be an immediate solution, but typical regression methods appear to be unsuccessful in view of complex nonlinearity of factors. As shown in Figure 2.1, finding standard relationships among variables and RI appears considerably difficult. On the other hand, the machine learning-based approaches would be an eventual candidate for a successful remedy as in other domains (Karlaftis and Vlahogianni, 2011). However, database of RI is still in its early development phase in terms of size, data consistency, and quality. It should be noted that if one is interested in individual airport's RI at a specific time, a machine learning-based "classification" will be also successful.

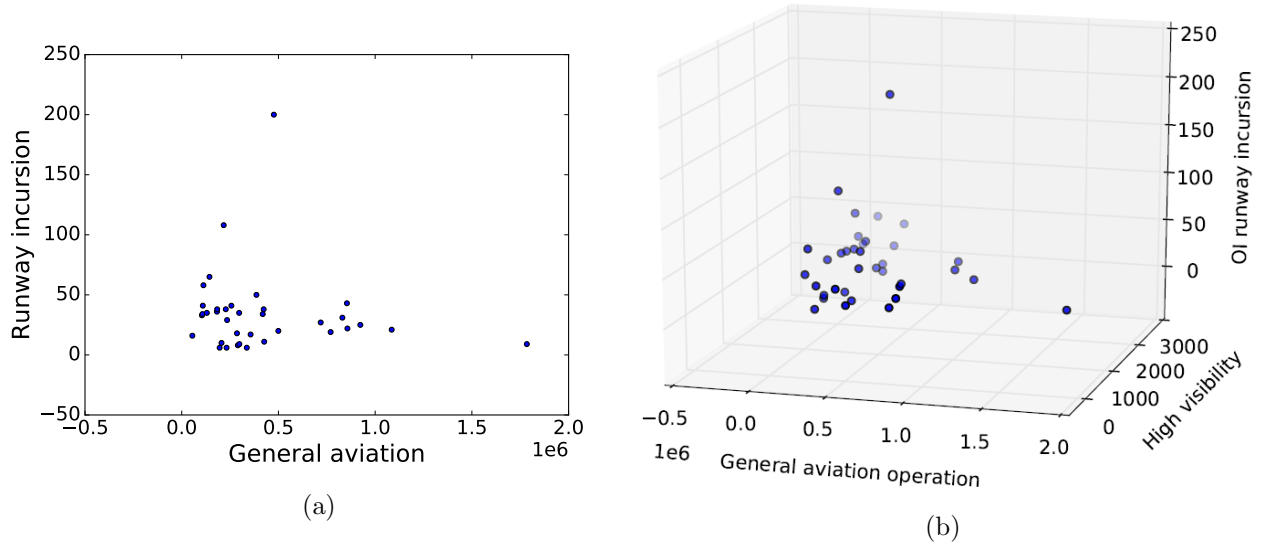


Figure 2.1: Scatter plot of variables: (a) runway incursion versus general aviation operation; (b) runway incursion versus general aviation operation and high visibility

But, this study's focus is on the total number of RI of an airport for 15 years and statistical investigation, learning, and prediction. Hence, such classification approaches will be another future work, when sufficient databases become available (i.e., large enough for training, validation and testing (Baesens, 2014)).

One of the implementational challenges is tied to the dispersed locations of RI databases. Key data pertaining to primary factors of RI are not located in the same location. We collected the data from different databases, developed programs to extract required information from raw data and transform into a suitable form of data for the dataset. Another issue was the computation cost attributed to a number of factors. Basically, not all of the factors are needed for accurate prediction. To find the best combination of factors contributing to RI prediction, multiple loop simulation should be done, leading to expensive computing time. We utilized the parallel strategy to solve this issue, which shall be addressed in the later section. The overall workflow is shown in Figure 2.2.

Objectives of this study are to (1) develop a systematic framework for gathering and processing various databases, (2) leverage the generalized additive model (GAM) to conduct statistical learn-

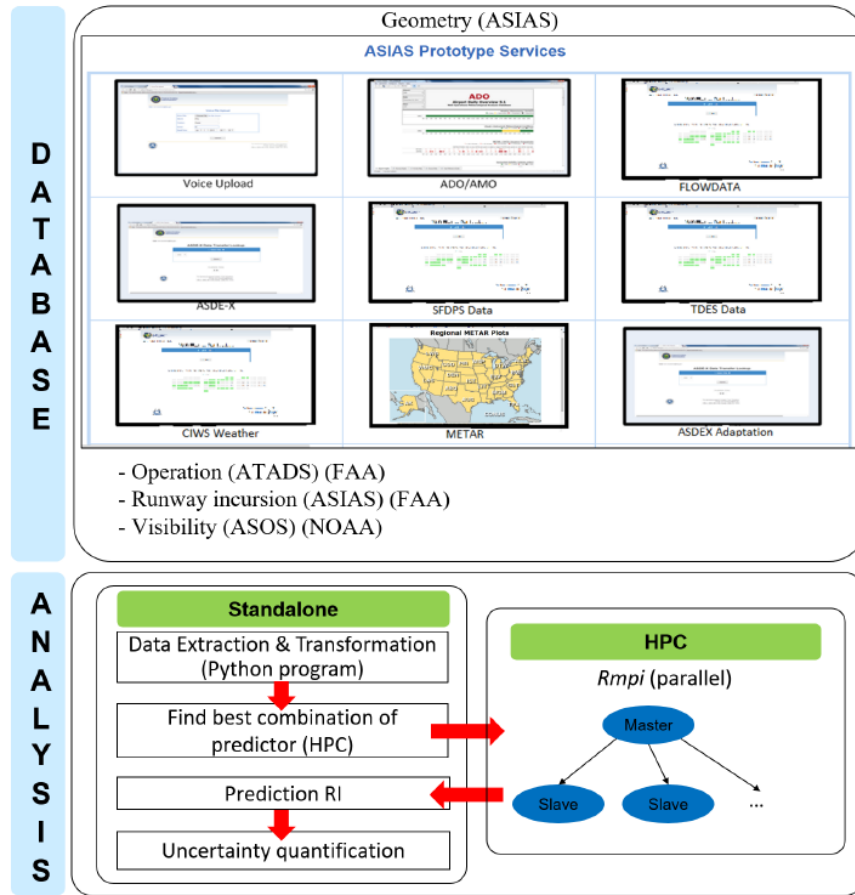


Figure 2.2: Workflow of runway incursion (RI) prediction using GAM: raw data is collected from various databases and transformed into suitable forms of dataset, with which GAM learns and predicts future RI on high performance computing (HPC)

ing and prediction, (3) introduce and apply GAM to RI prediction, (4) parallelize the suggested methods, and (5) provide an uncertainty quantification method for the GAM. If interest lies in an individual airport’s RI at a specific time frame, advanced machine learning algorithms may be helpful, e.g., a machine learning-based ”classification.” But, the goal of this study is to add a new dimension by providing an advanced statistical learning and prediction method. Statistical methods will be helpful to improve interpretability of the prediction model, thereby complementing machine learning approaches in the future. Such synergistic combinations of statistical and machine learning will be beneficial to research community. Hence, this study’s focus is on statistical prediction of the total number of RIs of 36 airports for 15 years.

As shall be elaborated later, GAM is a nonparametric statistical model developed by Hastie and Tibshirani (1990) and is highly flexible, being capable of embracing a large number of variables with substantial nonlinearity. The GAM can cover a wide range of statistical distributions, and these favorable attributes of the GAM enable us to learn and predict RI database and to make RI prediction procedure more comprehensible.

The outline of the paper is as follows: data structures used for building the dataset of RI and GAM-based predictions are addressed. The central algorithms regarding how to collect, extract, and transform the raw data tailored for GAM are presented. Cross validationbased procedure for finding the best combination of predictors (i.e., variables used for learning and prediction) is presented. A remark on a parallel strategy for the proposed algorithms is summarized. Importantly, the procedure for uncertainty quantification using bootstrapping and GAM is presented. All the processed data and prediction results of 36 airports are presented in Appendix [2.A](#).

2.2 Methodology

2.2.1 Data collection

To facilitate prediction of RIs, the primary data are classified to three categories: (1) geometric information, (2) operational data, and (3) visibility data. Airport runway is a long stretch of pavement on which an aircraft can take off and land in airport. The FAA aviation safety information analysis and sharing (ASIAS) system, developed by FAA provides a wide range of data regarding safety. In this study, spatial and geometric information of the 36 airports was obtained from the ASIAS system. Using the spatial and geometry data, numbers of runway, intersection between runways, and intersection between runway and taxiway were obtained by parsing a XML data. Operation data of aircraft in airport are also important. To collect and extract operation data, we leveraged the air traffic activity system (ATADS). ATADS provides all activity information related to air traffic, including airport operation, tower operation, terminal operation, and so on. The data obtained includes airport name and operation history of air carrier, air taxi, general aviation, and military for 15 years (from 2001 to 2015) of the major 36 airports. Visibility data was obtained

from automated surface observing system (ASOS) developed by a joint work done by the National Weather Service (NWS) that is a component of the National Oceanic and Atmospheric Administration (NOAA), the FAA, and the Department of Defense (DoD). The NOAA is a government agency which provides extensive information about weather, climate, and ocean. ASOS provides meteorological and climatological observation measure in more than 900 ASOS sites. They cover all major airports in the United States. Data can be obtained in the form of 1-minute, 5-minute, 1-hour. Visibility has three potential impact factors (i.e., slight, moderate, and high). Hours of factors were counted for 15 years.

Last, the RI data are obtained from ASIAs. They provide comprehensive information about RI in the most airports. There are three different types of runway incursion: (1) pilot deviation (PD), (2) operational incident (OI), and (3) vehicle (driver) deviation (VD). PD is defined by the incursion committed by a pilot of aircraft (e.g., landing or taking off without clearance from ATC); OI by ATC (e.g., clearance of an aircraft onto a runway while another aircraft is on the runway); VD by passing a runway holding mark without ATC clearance (FAA, 2008). Summary of data is shown in Table 2.1. Here, "predictors" mean the observed factors (or variables) that are used for the GAM-based learning and predictions of RI occurrence.

Table 2.1: Summary of datasets used in the current study

Data	Predictors	Types	Sources
Geometric information	Runway, intersection between runways, and intersection between runway and taxiway	Count (integer)	ASIAs
Operation data	Air carrier, air taxi, general aviation, military aviation, and total aviation	Count (integer)	ATADS
Visibility	High impact, moderate impact, and slight impact	Hour (integer)	ASOS

2.2.2 Data extraction and transformation

Because of the dispersed database locations, this study first investigated multiple heterogeneous databases to obtain data that are required to build GAM model. We collected several raw data from different databases, extracted only the required parts, and transformed them into a form suitable for the GAM.

First, the geometric data is obtained in the form of XML from AVIAS. The XML file contains polygon information of runway, taxiway, and other structures in an airport and the file consists of coordinates of points (i.e., x and y coordinates) which are connected each other to make polygon lines. We counted number of runway, intersection between runways, and intersection between runway and taxiway based on the number of a keyword in a tag. For example, a tag with `<Runway name="35L" id="8">` in the XML file means the polygon information of a new runway would be within the tag. We searched the keyword Runway and counted it as number of runway whenever our program found it. Second, the operation information was downloaded from the ATADS in the form of spread sheet. We directly downloaded operation information of 36 airports for 15 years, and thanks to various download options of ATADS further parsing process was not necessary.

Third, the visibility information was the most difficult to obtain because it requires multiple processing steps. A number of raw data files were downloaded from the NOAA file transfer protocol (FTP) (FAA) server, and then they were transformed into more interpretable form by using the JAVA program provided by NOAA. It should be noted that the same time frame of the weather data from NOAA is used for each incursion incidents (the generated dataset will be available upon request). The transformed data includes the United States Air Force (UASF) codes so that airports can be identified by the code. The data contains 1-hour information including the visibility presented in the unit of mile. The program counts hours of slight, moderate, and high visibility of 36 airports for 15 years based on the meteorological terminal aviation routine weather report (METAR) board (FAA) (Table 2.2).

2.2.3 Advanced statistical model, GAM

Compared to traditional regression methods, a generalized additive model (GAM) (Hastie and Tibshirani, 1990) is relatively new and rarely used in these fields. Thus, it is instructive to touch upon the key notions of GAM and its strengths, which are important for resolving our target RI problem. GAM is a sort of generalized linear model, but holds strong flexibility and general applicability. Rather than relying on pre-defined distributions or parameters, GAM harnesses unspecified smoothing functions. By virtue of the unspecified smoothing functions, covariates do not need to have a set of parameters. For predicting RI occurrence of i^{th} airport (denoted by $Y_i \in \mathbb{R}$) with n predictors (denoted by $\mathbf{x}_i \in \mathbb{R}^n$), the general form of GAM can be represented as:

$$g(\mu_i) = f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + \dots, \quad (2.1)$$

where g is a smooth link function; the expectation of Y_i given \mathbf{x}_i is denoted by $\mu_i \equiv \mathbb{E}(Y_i | \mathbf{x}_i)$; Y_i is a target response from an exponential family of distribution (e.g., normal, binomial, or gamma distribution); f_j are smooth functions of covariates x_{ji} (Wood, 2006). In our study, Y_i would mean the number of RI of i^{th} airport and \mathbf{x}_i consists of many factors of the airport including the number of runways, visibility, etc. In essence, the GAM has non-specified smoothing function per each predictor, and this fact imparts substantial flexibility to GAM. For brevity of explanation, the following description involves a normally distributed single variable, but generalization to multiple variables is straightforward (Wood, 2006). Now, let GAM be $\mathbb{E}(Y | x) = f(x)$, and the smoothing

Table 2.2: Potential impact (mile) visibility criteria based on METAR board

Threat	Visibility
None	≥ 5.1
Slight	$5.1 > X \geq 3$
Moderate	$3 > X \geq 1$
High	< 1

function f can be represented as:

$$f(x) = \sum_{j=1}^k b_j(x)\beta_j \quad (2.2)$$

where $b_j(x)$ is the j basis function and β_j is an unknown parameter. Model fitting can be done by maximizing the corresponding likelihood with a penalty term which is given as $\lambda \int [f''(x)]^2 dx$ where λ is smoothing parameter. Too large λ leads to an over-smoothed estimate while too small λ results in an under-smoothed estimate. The error becomes the largest in the both extreme cases. The optimized λ value can be chosen in such a way that model can fit accurately by minimizing generalized cross validation (GCV) score (see (Golub et al., 1979) for detail of GCV). λ is selected appropriately via the relevant GAM library; thus, in general, there is no need to manually adjust λ .

A spline basis should be selected for GAM building. There are two types of basis which are commonly used in GAM: (a) thin plate regression splines (TPRS) (Wood, 2003) and (b) cubic regression spline (CRS) (Wood, 2006). Cubic spline is a curve formed by connecting a number of cubic polynomial sections. Those sections are connected each other at "knot", a certain point of which location should be selected in advance for the cubic spline basis. The cubic polynomial sections are joined in a such way that the entire spline should be continuous up to second derivative. Although there are many ways to get a basis for cubic spline, a simple basis is offered by (Gu, 2013), which is given by

$$b_1(x) = 1, b_2(x) = x, \text{ and } b_{i+2} = R(x, x_i^*) \text{ for } i = 1, 2, \dots, p-2 \quad (2.3)$$

where p is number of rank for basis, x_i^* is knot location, and,

$$\begin{aligned} R(x, x_i^*) = & \left[(x_i^* - 1/2)^2 - 1/12 \right] \left[(x - 1/2)^2 - 1/12 \right] / 4 \\ & - \left[(|x - x_i^*| - 1/2)^4 - 1/2(|x - x_i^*| - 1/2)^2 + 7/240 \right] / 24 \end{aligned} \quad (2.4)$$

Thin plate spline (Duchon, 1977) can be used for multiple covariates. Thin plate spline function, \mathbf{f} , can be determined by minimizing $\|\mathbf{y} - \mathbf{f}\|^2 + \lambda J_{md}(f)$, where \mathbf{y} is the vector of y_i data and $\mathbf{f} = [f(x_1), f(x_2), \dots, f(x_n)]^T$. $J_{md}(f)$ is a penalty functional measuring the wiggleness of f , and

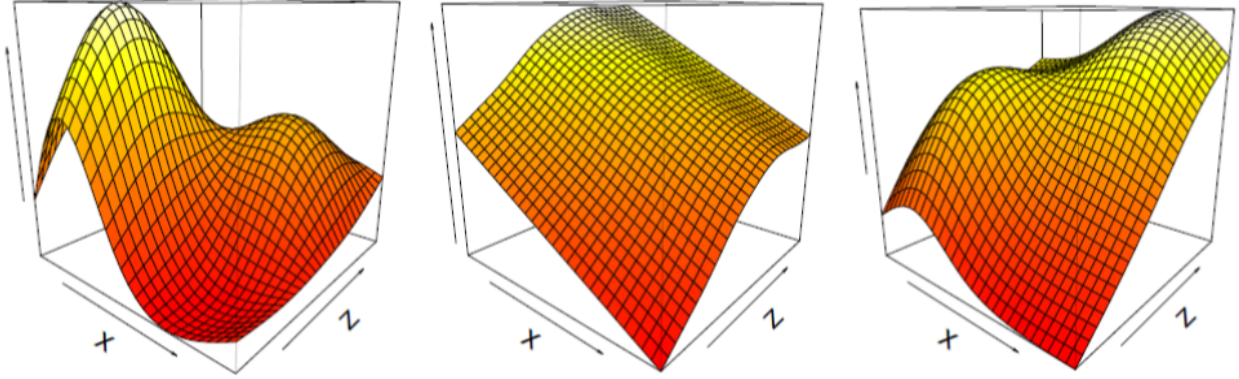


Figure 2.3: Example of thin plate spline basis function using 2 covariates

λ is a smoothing parameter, controlling the tradeoff between data fitting and smoothness of f . The wiggleness penalty is defined as

$$J_{md} = \int \cdots \int_{R_d} \sum_{v_1 + \cdots + v_d} \frac{m!}{v_1! + \cdots + v_d!} \left(\frac{\partial^m f}{\partial x_1^{v_1}, \cdots, \partial x_d^{v_d}} \right)^2 dx_1, \cdots, dx_d. \quad (2.5)$$

One example of thin plate spline basis function with 2 covariates is shown in Figure 2.3.

In sum, GAM requires no prejudice on relations among parameters and holds little restriction to the number of variables and nonlinear distribution of variables. Importantly, GAM's internal setting always seeks to balance the fitting accuracy and smoothness, in which the generality and flexibility of GAM is rooted.

2.2.4 Metrics for prediction accuracy

In this study, three metrics are used to compare the GAM-based prediction performance: (1) CVE_b/CVE = the ratio between base cross validation error (CVE_b) and cross validation error (CVE); (2) the Pearson correlation, ρ ; (3) the coefficient of determination, R^2 . The CVE and CVE_b are defined as

$$CVE = \frac{1}{N} \sum_{i=1}^N (y_{ex}^i - y_{pr}^i)^2; \quad CVE_b = \frac{1}{N} \sum_{i=1}^N (y_{ex}^i - y_{mean,pr})^2, \quad (2.6)$$

where N is number of data, y_{ex}^i is the i_{th} real-world measured response, y_{pr}^i is i_{th} predicted response in the cross-validation procedure, and $y_{mean,pr}$ is the mean of predicted values. ρ and R^2 are defined

as

$$\rho = \frac{COV(y_{pr}, y_{ex})}{\sigma_{y_{pr}} \times \sigma_{y_{ex}}}; R^2 = 1 - \frac{\sum_{i=1}^N (y_{ex}^i - y_{mean,pr})^2}{\sum_{i=1}^N (y_{ex}^i - y_{pr}^i)^2} \quad (2.7)$$

The CVE_b/CVE is an indicator of the goodness of model fitting: i.e., CVE_b is a naive prediction using the mean of predicted values, and thus a high ratio means the good prediction performance. We use this ratio as auxiliary metric for accuracy of fit following (Kamdar et al., 2016). ρ and R^2 are our primary metrics to measure accuracy of fitted model. ρ indicates the linear correlation between real-measured and predicted values, and when the model is fitted well, becomes closer to 1. R^2 represents proportion of variance in the response variables that can be predictable from predictor variables. R^2 will be close to 1 if the model fits well. This choice has been made following the comparable study on machine learning comparisons of (Kamdar et al., 2016). In essence, the higher metrics, the more accurate predictions.

2.3 Selection of Best GAM Model Using Parallel Computing

The GAM can be built upon arbitrary combinations of many predictors. Among many possibilities, a prudent choice of predictors is critical for accurate GAM prediction. To avoid artificial bias in the selection of predictors and automatic processing, this framework objectively compares the aforementioned three metrics of prediction performance (i.e., $CVE_b = CVE$, ρ and R^2) to determine the best combination of predictors. In total, 14 variables are taken from raw data without any prejudices on relations or a priori knowledge on the relative significance of predictors. The 14 variables are: runway number, intersection number for runways, intersection number of runway and taxiway, air carrier operation, air taxi operation, general aviation operation, military operation, total operation, average visibility, low visibility hours, moderate visibility hours, high visibility hours, sum of high and moderate visibility hours, and sum of all visibility hours (Tables 2.1 and 2.2). The prediction target response is the total number of RI occurrence per airport. The proposed approach for the search of the best predictor combination is straightforward, yet computationally expensive: e.g., total combination of seven variables selected from 14 total variables = $14!/(7!(14-7)!) = 3,432$. In a future extension, if dozens or hundreds of predictors are used,

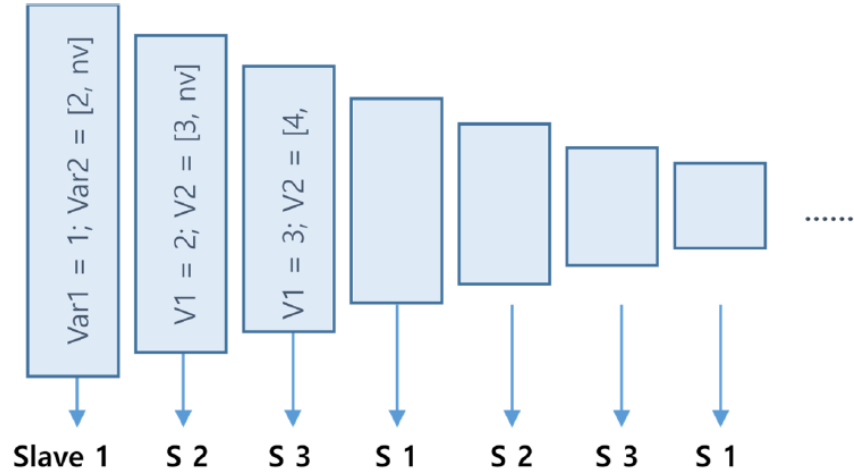


Figure 2.4: Cyclic allocation of the proposed parallel code of R & $Rmpi$; two-variable case is shown with "nv" meaning the total number of variables. Height of box corresponds computation load

parallel computing is essential for practical efficiency. For instance, the serial version's running time for the seven-variable case was 217 min on a desktop computer (2.8 GHz dual cores, 8GB memory), and the statistical library uses R. Such a long running time may be attributed to the expensive computation cost of the GAM library as mentioned in (Wood, 2006). In particular, the total run time of 217 min is composed of 1 min for making the dataset, 207 min for estimating the GAM, and 9 min for predictions and metric calculations. For scalability, this study distributes the combination search task to available processors. Particularly, a parallel computing algorithm was developed using $Rmpi$ (Yu, 2002). The $Rmpi$ is controlled by only one master, and a number of slaves can be spawned. Because the computation load decreases as the size of interwoven loops decreases (Figure 2.4), the so-called cyclic allocation of tasks is used to ensure load balance on the slave processors. A successful parallelization can be achieved by cyclically allocating jobs to available slaves. As the problem size increases, the cyclic allocation scheme approaches the optimal parallel load balancing (Kam et al., 2011). The parallel algorithm was tested and results are summarized in Figure 2.5. The best speed-up was achieved with 56 slaves.

It is instructive to touch upon other methods that can be alternatives because there exist efficient methods different from the direct parallel comparison of all possible cases. For instance,

principal component analysis (PCA) (Jolliffe, 2002) would be a good candidate because PCA is helpful to identify predictors’ relative contribution to the total variability of raw data and to reveal intervariable correlations. To briefly show this nature, Figure 2.6 summarizes a biplot from PCA using the 14 predictor variables of this study using the R package. Indeed, Figure 2.6 confirms that the positive correlation of visibility variables and their large contribution to the total variability, which is consistent with the finding from this study (see the section ”Uncertainty Quantification Procedure”). To examine whether the PCA-guided variable selection is helpful, three cases were considered according to the PCA results in Figure 2.6: (1) prediction using three variables in the positive direction, (2) prediction using six variables in the negative direction, and (3) prediction using nine variables in both directions along the principal axis. With those cases, additional analyses of GAM and a multiple regression were performed. The PCA-guided variable combinations led to relatively less prediction accuracy than the direct search algorithm proposed herein (Table 2.3). This result suggests that, although PCA is helpful in understanding variability, the PCA-guided set of variables may be different from the optimal combination with the highest predictive power. Hence, this study continues to seek a computationally straightforward framework that can explicitly select out salient predictors from arbitrarily many real-world variables. Such a straightforward framework is easily made autonomous and parallelizable. Also, the proposed pair of GAM and the bootstrapping method appears to work well to deal with uncertainty quantification. Hence, the use of another alternative such as PCA will be a future extension topic.

Table 2.3: Comparison of prediction performance between direct search algorithm (proposed herein) and PCA-guided variables (all values are CVE_b/CVE)

Number of variables	5 (direct search proposed herein)	3 in the positive direction(PCA)	6 in the negative direction(PCA)	9 in both directions (PCA)
Multiple regression	1.31	0.89	0.87	0.78
GAM	3.34	0.03	0.12	0.07

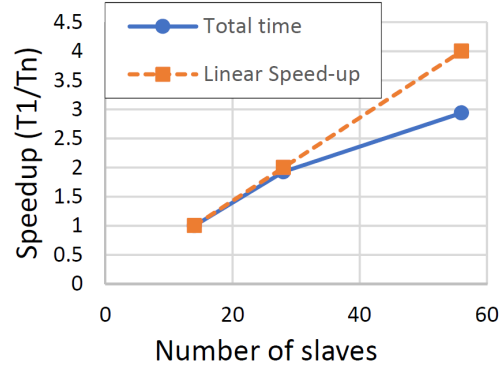


Figure 2.5: Parallel computing performance of *R* & *Rmpi* code for finding the 7-variable combination out of 3,432 total combinations

As addressed before, a parallel computing algorithm enables consideration of all possible variable combinations and comparison of the prediction metrics to obtain the "best" set of predictors (see Figure 2.7 for algorithm). The best combination identified consists of five predictors: (1) the number of taxi operations, (2) the number of general operations, (3) hours of high impact visibility, (4) hours of slight impact visibility, and (5) sum of hours of high, moderate, and slight impact visibility (Table 2.4). Figure 2.8 shows the prediction quality results using two smoothers of GAM, i.e., CRS and TPRS. In essence, larger metrics mean better prediction in both summaries. Both CRS and TPRS identify the same conclusion for the best combination. Particularly for the current data and the RI prediction problem, CRS appears to perform better than TPRS (Figure 2.8), which may be attributable to the unique characteristics of RI data. This relative prediction performance of GAM-CRS and GAM-TPRS may be changed as data quantity and complexity are added in future extension of this work. Still, the proposed procedure and methodology will be meaningful because the framework is independent of data-related changes and is readily expandable for more variables.

2.4 Prediction Results

Per the least requirements of GAM, a logarithmic link function was chosen that can easily incorporate multiplicative relations of engineering variables. Because all the quantified predictors

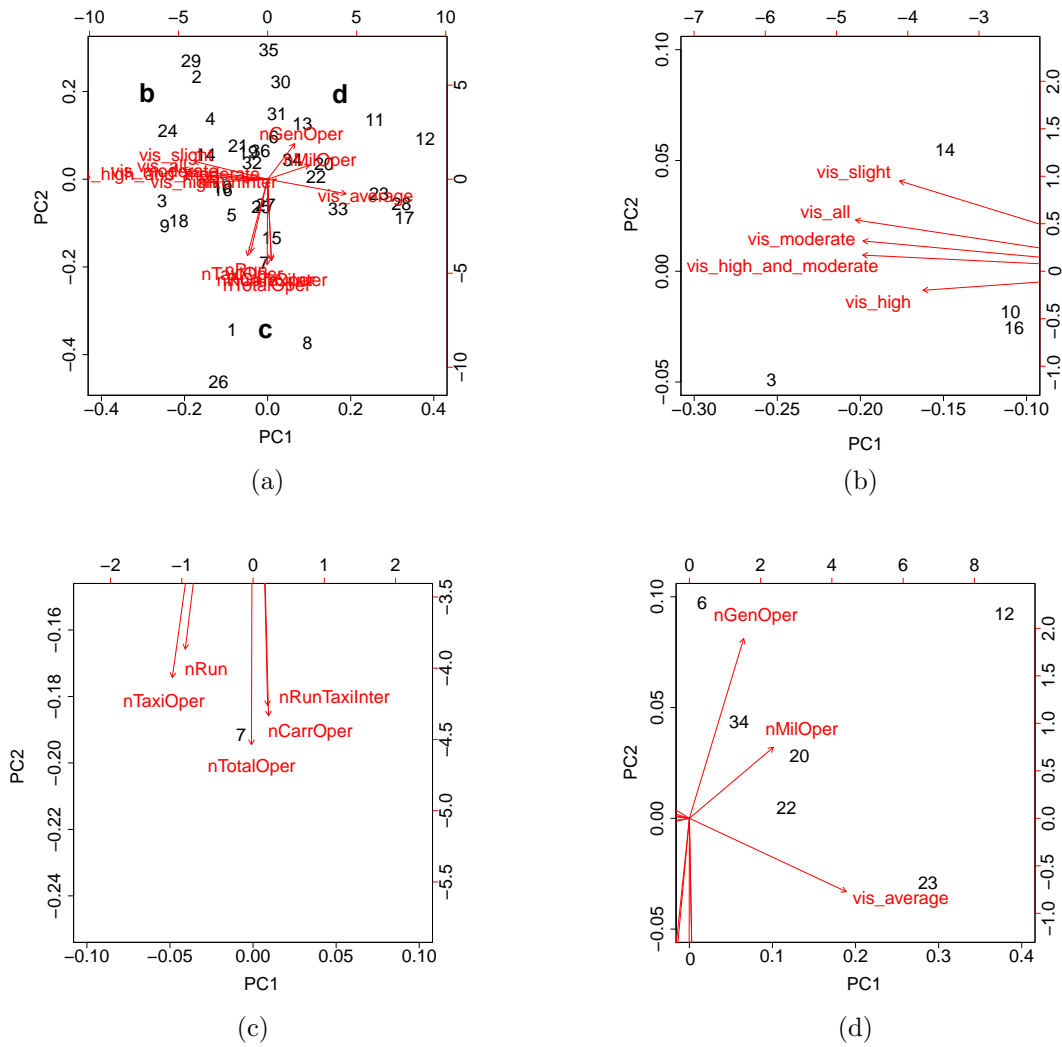


Figure 2.6: Biplot from principle component analysis (PCA): (a) entire biplot; (b) part of biplot denoted by dashed box "b"; (c) by "c"; and (d) by "d"

Algorithm <i>Rmpi</i> for best variable combination selection	
Input: dataset \mathbf{D}	
Output: $[\mathbf{V}]^*$, the best variable combination with the highest prediction accuracy	
Task	Description
1: [on P_0]	
2: $\mathbf{D} \leftarrow$ dataset	- \mathbf{D} , data matrix
3: Spawn n slaves	- spawn n slave to assign a task
4: [on $P_1 \sim P_{p-1}$]	
5: $m \leftarrow$ MPI_Comm_rank()	- m , CPI id
6: $p \leftarrow$ MPI_Comm_size()	- p , total number of CPU
7: $n_combi_local = n_combi / (p-1)$	- n_combi , number of all combinations
8: main loop: $t = 1, \dots, n_combi_local$	- n_combi_local , number of combinations assigned to each slave
9: $[\mathbf{V}]_t \leftarrow$ predictors combination	- n_rows , number of row in dataset
10: for $i=1$ to n_rows	- $\mathbf{D}_{(i)}$, dataset of i th row
11: $\mathbf{D}_{(i)}$ and $\mathbf{D}_{(-i)} \leftarrow \mathbf{D}$	- $\mathbf{D}_{(-i)}$, dataset excluding i th row
12: $\mathbf{F}_i \leftarrow$ GAM($\mathbf{D}_{(-i)}$)	- \mathbf{F}_i , fitted model using $\mathbf{D}_{(-i)}$
13: $[\mathbf{P}_i] \leftarrow$ GAM($\mathbf{F}_i, \mathbf{D}_{(i)}$)	- $[\mathbf{P}_i]$, prediction from the fitted model, \mathbf{F}_i using $\mathbf{D}_{(i)}$
14: end for	
15: $\mathbf{M}_t^m \leftarrow$ CVE _b /CVE(\mathbf{P}_i), $\rho(\mathbf{P}_i)$, $R^2(\mathbf{P}_i)$	- \mathbf{M}_t^m , prediction performance metrics
16: end main loop	
17: Send \mathbf{M}_t^m to P_0	
18: [on P_0]	
19: Receive $\mathbf{M}_t^m, m = 1, \dots, p-1$	
20: $[\mathbf{V}]^* = [\mathbf{V}]_{t^*}$ where $t^* = \operatorname{argmax}_t \{\mathbf{M}_t^m; t = 1 \sim n_combi\}$	- Select the best combination of predictors for high prediction accuracy

Figure 2.7: Pseudo code for finding the best combination of predictor variables

and response are positive and considered as countable (i.e., integers), the Poisson distribution was assumed. As recommended by Wood (2006), the parameter k (i.e., the number of basis dimensions in smooth functions) was set to 6; k is not the number of predictors but is related to how many bases are used in each smooth function.

For a small data sample as in this case, cross validation can be used instead of using train and test set separately (Baesens, 2014). To evaluate the prediction capability systematically, the cross-validation method was applied: (1) exclusion of an airport, (2) construction of a GAM by learning the remaining airport data, and (3) prediction of the runway incursion of the omitted airport. To construct the GAM, one of the airports was excluded whereas learning samples (i.e., other airport data) were used in the cross validation (Figure 2.9). Thereafter, a series of runway incursions of the excluded airport was predicted using the GAM. These steps were repeated throughout all airport

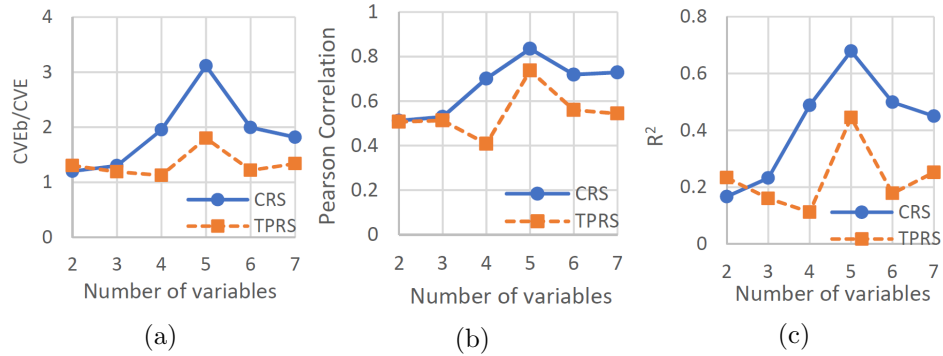


Figure 2.8: Comparison of performance between CRS and TPRS on this study: (a) ratio of $CVE_b = CVE$; (b) Pearson correlation; (c) coefficient of determination

Table 2.4: Metrics used for best combination of predictor variables (GAM-CRS)

Number of variables	CVE_b/CVE	Pearson correlation	Coefficient of determination
2	1.2	0.512	0.1667
3	1.302	0.529	0.232
4	1.952	0.701	0.488
5	3.115	0.835	0.679
6	1.995	0.719	0.499
7	1.818	0.729	0.45

Note: The bold values show the largest value.

data. The difference between the predicted number of RIs from GAM and the original value of RIs for the excluded airport directly represents the GAM's prediction quality.

To demonstrate the prediction results, so-called Q-Q plots were drawn to correlate the scaled response of real measured and predicted value [Figure 2.10a; a linear line means accurate prediction]. Note that all statistical predictions in Figure 2.10 are drawn from the best GAM model that only uses 5 predictors. Remarkably, the predicted responses exhibit good accuracy compared to the real-world data even though there was no bias used for statistical learning and prediction. Figure 2.10b shows that residuals are scattered evenly, and thus, the developed statistical model appears

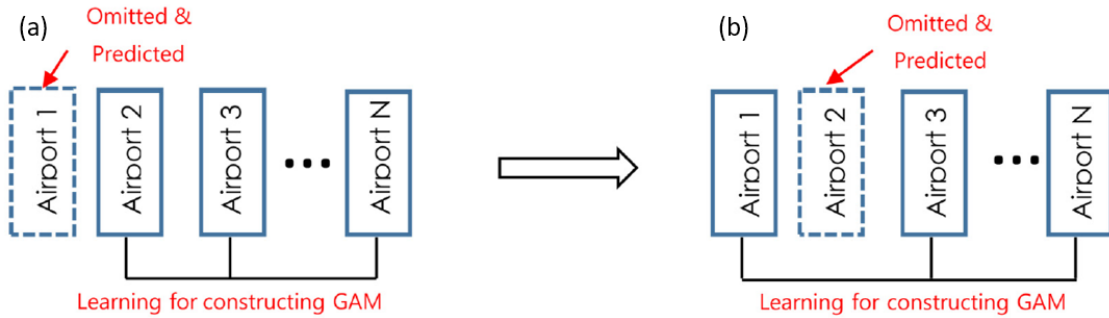


Figure 2.9: Illustration of cross validation: (a) shows that the first airport's data is omitted, a GAM is constructed by learning all other airport data; (b) shows the same procedure by omitting the second airport data

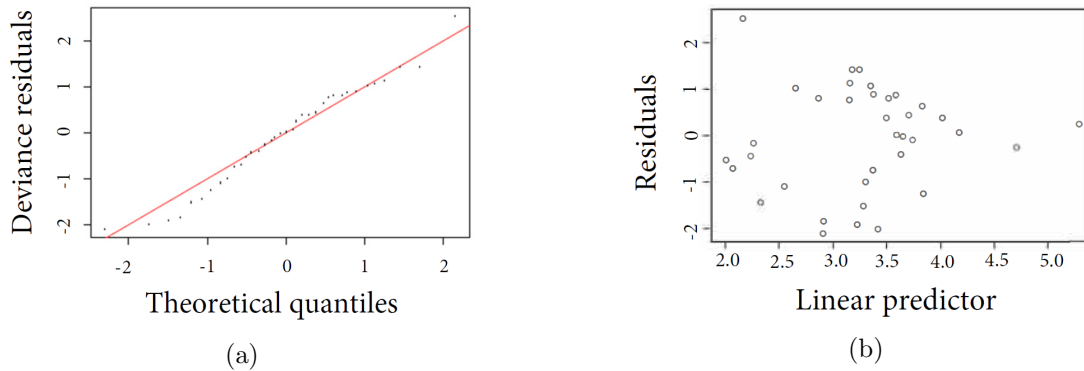


Figure 2.10: (a) Q-Q plot of real-world measured data and predicted data; (b) residuals plot showing that residuals are evenly scattered

to be acceptable. Additionally, Appendix 2.B presents a comparison between an artificial neural network and GAM, which confirms the GAM's promising predictive power.

A parameter study on the smoothing parameter, λ , was also conducted to examine whether the automatically optimized lambda (denoted as λ^*) guarantees the highest prediction accuracy. Figure 2.11 summarizes GCV scores with varying λ . As expected, the optimal λ (denoted as $\lambda^* = 7.384$) leads to the lowest GCV score, and the GCV score increases as the λ deviates from the optimal value; thus, this study recommends the automatic optimization of λ . The λ^* is automatically optimized by the GAM library. To demonstrate this optimization, this study manually changed

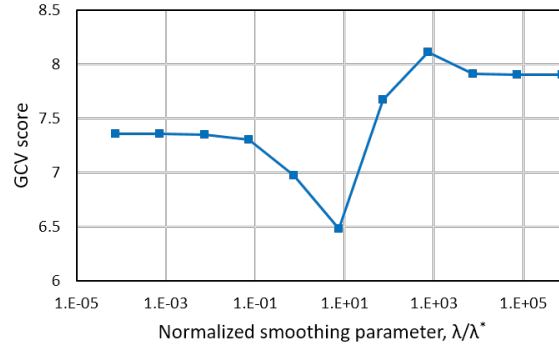


Figure 2.11: GCV score with varying smoothing parameter. (λ^* = automatically optimized value)

the lambda values, which can be done by overriding the GAM library using a command such as `"gam(response s(predictor,sp=7.384))"` in *R* terminal.

2.5 Uncertainty Quantification Procedure

Of particular interest is variability of individual predictors. Hence, the confidence bands of each of five predictors used for the smoothing function of the best GAM were summarized. As shown in Figure 2.12, the vertical axis (y-axis) represents the centered smoothing function, i.e., $\sum_{i=1}^n s(x_i) = 0$ for each predictor, where $s(x_i)$ is a smoothing function and x_i is the predictor under consideration. The solid line indicates a centered smoothing function fitted, and the dotted line shows the 95% confidence band. A wider width of the confidence band implies more variability of the predictor. Hence, comparing confidence bands offers a relative order of variability of predictors. The variability of the hours of slight visibility impact appears relatively higher than the operation number based on the range of the confidence bands (e.g., the distance of two dotted line in Figure 2.12a is larger than for the other four cases), which means visibility variables have more influence on uncertainty in prediction than operation variables.

The prediction interval was investigated because a statistical prediction essentially entails uncertainty for various reasons. To quantitatively assess the uncertainty of the GAM prediction, the confidence interval of the predicted RIs of 36 airports is shown. The confidence interval in RI

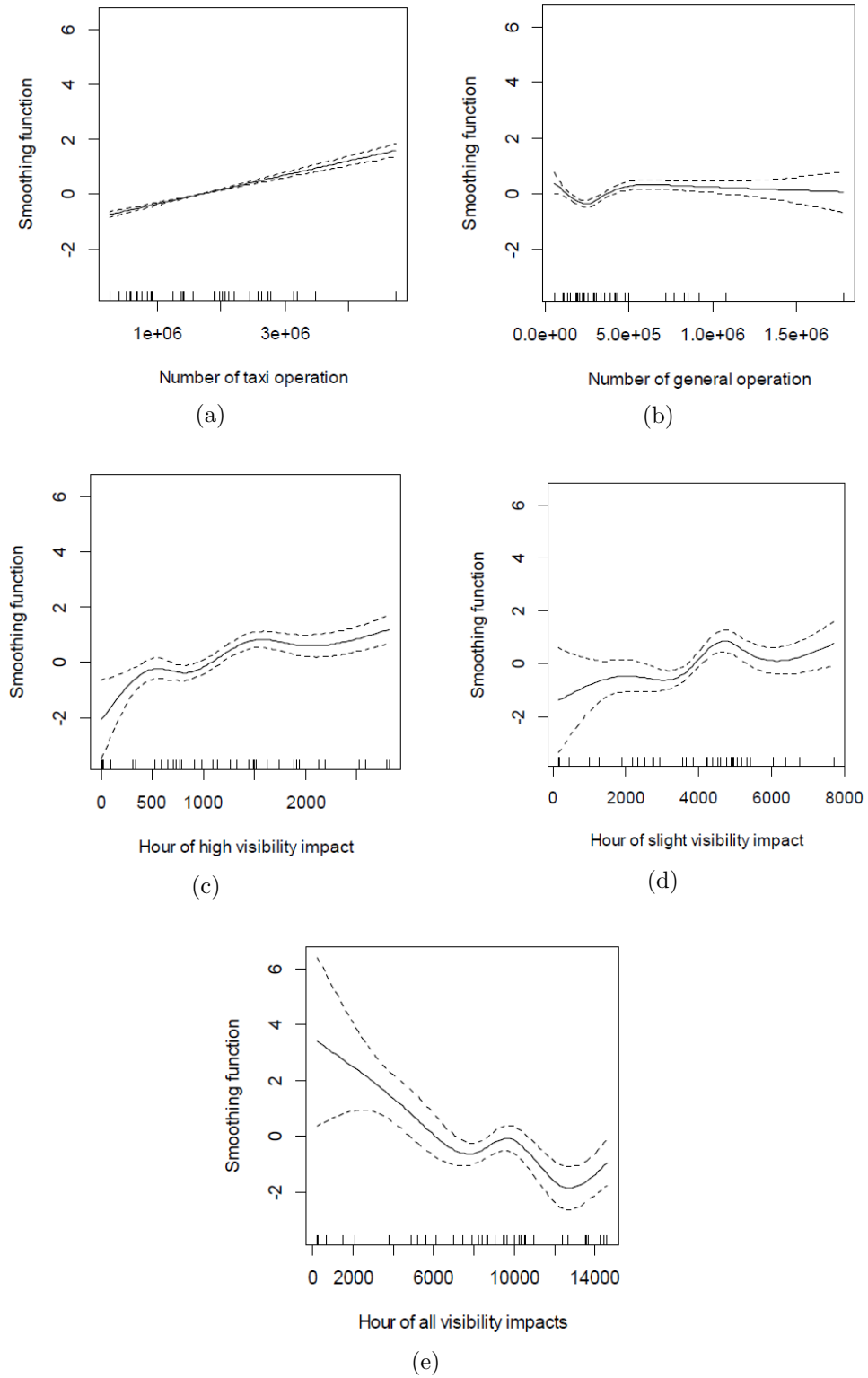


Figure 2.12: Confidence interval of smoothing functions of five predictors: (a) the number of taxi operations; (b) the number of general aviation operations; (c) hour of high visibility impact; (d) hour of slight visibility impact; (e) hour of sum of visibility impacts

prediction for 36 airports was generated by the percentile method using bootstrapping (Efron and Tibshirani, 1994). The detailed procedure to obtain a bootstrapping sample is as follows:

1. Fit a GAM model and obtain the fitted responses \hat{y}_i and calculate residuals $\varepsilon_i = y_i - \hat{y}_i$ ($i = 1, \dots, n$), where y_i is measured response and n is the sample size.
2. Generate a synthetic sample set $y_i^* = \hat{y}_i + \hat{\varepsilon}_{j,centred}$ by resampling centred residuals, where $\hat{\varepsilon}_{j,centred}$ is generated by:
 - (a) randomly selecting a residual, $\hat{\varepsilon}_j$ from the set of residuals of step 1 (j is randomly selected from $i = 1, \dots, n$ with replacement);
 - (b) subtracting mean value of ε_j from $\hat{\varepsilon}_j$ for every i .
3. Re-fit the regression model using the synthetic sample set y_i^* .
4. Repeat steps (2) and (3) B times, then B bootstrapped predicted response samples, $\hat{y}_1^*, \hat{y}_2^*, \dots, \hat{y}_B^*$ are generated.

These B bootstrapped samples are used to develop the confidence interval for the RI prediction in 36 airports by adopting the percentile method (Efron and Hastie, 2016). For example, when we have B bootstrap samples, the cumulative distribution function of bootstrap samples less than b can be written as

$$\hat{G}(b) = \mathcal{F}\{\hat{y}_i^* \leq b\} / B, \quad i = 1, \dots, B, \quad (2.8)$$

where \mathcal{F} represents frequencies of y_i^* . We can find a point with a specific percentile (α) using the inverse function of \hat{G} which is given by

$$\hat{y}^{*(\alpha)} = \hat{G}^{-1}(\alpha). \quad (2.9)$$

And then, the 95% confidence interval is obtained as

$$(\hat{y}^{*(0.025)}, \hat{y}^{*(0.975)}). \quad (2.10)$$

The physical meaning of the confidence interval is that the predicted number of RIs of a specific airport may fall into the range with 95% probability. To provide a sense of how different

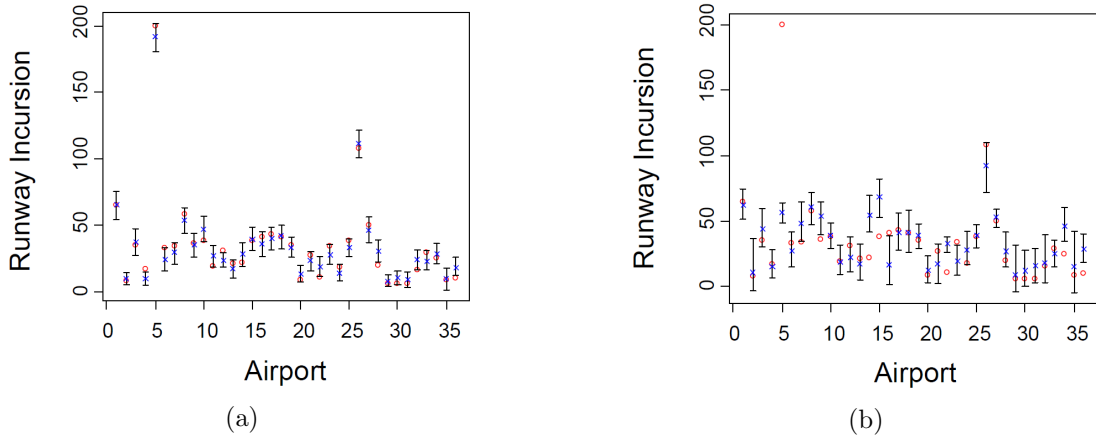


Figure 2.13: Confidence interval for GAM prediction points of 36 airports using (a) GAM and (b) multivariate linear regression: vertical bar represents 95% confidence interval, circle represents measured (real) RI number, and "x" mark represents a median value of bootstrap samples; horizontal axis means airport index; table of 36 airport indexes and generated data is presented in Appendixes 2.A and 2.B

statistical models influence the confidence interval, two case studies using GAM and a multivariate linear regression (MLR) with the same predictors are juxtaposed in Figure 2.13 (*R* package was used). The measured RI number and the median of bootstrap samples are marked by "o" and "x," respectively. As shown in Figure 2.13, MLR appears to exhibit low predictive power and wider confidence intervals compared to GAM. In future extensions of this research, more predictors from diverse databases such as human factors related to the pilots or sophisticated weather information would help improve the predictive power and shorten the confidence interval. Overall, GAM prediction entails narrow confidence intervals, strengthening the authors' confidence in these statistical predictions.

2.6 Conclusions

This study developed a computational framework to leverage an advanced statistical learning and prediction method to resolve runway incursion (RI) problems in the United States airports. A systematic procedure for gathering, processing, and creating interpretable datasets from various data sources has been documented. The framework adopts GAM, which has notable strengths

in flexibility and expandability. By virtue of the "additive" nature of GAM's formulation, GAM can accommodate any number of predictors in the future extension of this work, which will facilitate future application or sophistication to airports having comprehensive databases. Practical applications with GAM to data from the major 36 US airports show a promising predictive power. Notably, the predictions were made without any prejudice on relations or a priori knowledge of the raw data. Results suggest that all factors are not always necessary for the best prediction of RI, and rather, there appear to exist significant relations among a few manageable factors that may govern RIs. The identified five factors include (1) the number of taxi operations, (2) the number of general operations, (3) hours of high impact visibility, (4) hours of slight impact visibility, and (5) sum of hours of high, moderate, and slight impact visibility.

With persistent efforts, researchers will have increasing accessibility to the growing aviation databases. Thus, in future extensions, the proposed framework will complement the new data-driven discovery in the aviation field and also facilitate machine learning-based approaches. Some of the generated datasets are provided herein, and all other relevant data of the 36 target airports will be available upon request. An airport-specific learning and prediction would be helpful for improving predictive power. However, because the focus of this work lies in development of a general-purpose framework that can help investigate the total number of 15 years' of RIs, the authors started from common databases that were measured in a consistent manner by relevant agencies. Such an airport-specific approach will be a future research topic. Furthermore, airplane-specific classification using machine learning-based or PCA-based approaches would be another future research topic. Last but not least, the contribution of the proposed statistical prediction method to the machine learning mainstream is noteworthy. The statistical learning can provide clear causal pathways between descriptors and targets as well as the relative importance of descriptors for a given target. This leads to a sort of "glass box" prediction. Such a statistical glass box will complement machine learning by facilitating the selection of salient attributes, and will help stakeholders to devise practical decisions with the clear causal pathways.

2.A Appendix I. Dataset for Current Study

Note that the target response of this study is the operational incident (OI) incursion shown in the last column. Dataset used in statistical learning and prediction is summarized in Tables A2.1 and A2.2.

2.B Appendix II. Comparison of GAM and Artificial Neural Network

Although it is not the main scope of this study, to briefly compare the relative performance trends, an artificial neural network (ANN) was run using the Levenberg-Marquardt algorithm as the main learning algorithm. In general, on function approximation problems, for networks that contain up to a few hundred weights, the Levenberg-Marquardt algorithm will have the fastest convergence. This advantage is especially noticeable if very accurate training is required. In many cases, the Levenberg-Marquardt algorithm is able to obtain lower mean square errors than any of the other algorithms tested. In this exploratory study, one hidden layer with a lower number of hidden neurons (310) has been used, and a learning rate of 0.1 was used.

As shown below (Table B2.1), the case of ANN prediction using 36 airports (30 training and 6 validation) exhibits the best predictive performance, having R^2 of 0.66. This confirms that the proposed statistical method has comparable or slightly better predictive power than ANN because the GAM-CRS produces R^2 of 0.679 (Table 2.4). Still, a generalization of this exploratory comparison needs in-depth investigation because other sophisticated machine learning approaches may outperform the current statistical framework (e.g., by larger data inclusion, data-oriented tuning, outlier removal, etc.).

2.7 Acknowledgments

This study is supported by the Partnership to Enhance General Aviation Safety, Accessibility and Sustainability (PEGASAS) Center of Excellence (COE) fellowship program of the Federal Aviation Administration (FAA). Regarding the data acquisition and working environment, the

Table A2.1: Dataset used in statistical learning and prediction

Index	Air port	Runway	Runway intersection	Run-taxi intersection	Air carrier	Air taxi	General aviation	Military	Total operation
1	ATL	5	0	81	10,663,103	3,192,083	143,416	14,470	14,013,072
2	BDL	3	0	21	934,479	516,253	290,022	51,665	1,792,419
3	BOS	6	7	57	3,356,134	2,202,001	296,893	10,327	5,865,355
4	BWI	3	1	34	3,150,569	696,324	356,096	19,225	4,222,214
5	CLT	4	1	71	4,436,253	2,774,368	476,256	29,574	7,716,451
6	DCA	3	3	34	2,598,898	1,401,885	104,480	23,499	4,128,762
7	DEN	6	0	65	6,157,278	2,453,254	107,223	8,749	8,726,504
8	DFW	7	0	124	7,262,435	3,120,579	112,231	4,857	10,500,102
9	DTW	6	4	74	3,995,086	2,733,246	182,195	3,260	6,913,787
10	EWR	3	2	65	4,210,659	1,982,908	183,228	4,225	6,381,020
11	FLL	2	0	30	2,728,828	758,871	769,841	8,392	4,265,932
12	HNL	4	2	45	2,476,190	930,783	830,621	237,506	4,475,100
13	HOU	4	4	53	1,635,594	573,261	1,085,411	21,859	3,316,125
14	IAD	4	0	42	2,392,296	2,576,821	856,588	19,905	5,845,610
15	IAH	5	0	55	4,207,466	3,476,742	227,781	3,972	7,915,961
16	JFK	4	2	68	4,888,803	842,417	109,035	5,574	5,845,829
17	LAS	4	2	62	5,332,958	1,887,286	854,019	83,182	8,157,445
18	LAX	4	0	63	7,155,959	2,058,158	256,701	37,201	9,508,019
19	LGA	2	1	33	3,562,992	2,007,874	129,959	5,057	5,705,882
20	MCO	4	0	47	3,947,149	578,731	297,938	17,120	4,840,938
21	MDW	5	6	56	2,735,601	682,937	718,436	21,108	4,158,082
22	MEM	4	0	63	3,008,710	1,563,710	426,365	28,734	5,027,519
23	MIA	4	1	62	4,634,269	893,108	418,863	29,526	5,975,766
24	MKE	5	6	48	960,681	1,369,844	285,610	38,528	2,654,663
25	MSP	4	3	78	4,569,670	1,909,646	423,698	41,425	6,944,439
26	ORD	9	3	100	8,659,436	4,725,974	217,020	3,317	13,605,747
27	PHL	4	1	61	3,936,716	2,647,590	386,299	16,930	6,987,535
28	PHX	3	0	59	5,623,358	1,253,271	499,468	45,674	7,421,771
29	PVD	2	1	20	655,003	394,361	336,449	11,174	1,396,987
30	SAN	1	0	15	2,289,793	573,584	196,089	15,128	3,074,594
31	SDF	3	1	51	1,389,627	761,079	231,981	47,200	2,429,887
32	SEA	3	0	47	4,171,612	918,411	54,910	1,471	5,146,404
33	SFO	4	4	86	4,265,157	1,245,092	234,651	42,421	5,787,321
34	SLC	4	0	46	2,503,005	2,113,580	923,224	55,476	5,595,285
35	SNA	2	0	22	1,294,103	265,226	1,784,108	3,625	3,347,062
36	STL	4	1	57	2,333,033	1,426,229	205,691	62,735	4,027,688

Table A2.2: Dataset used in statistical learning and prediction(*Continued*)

Index	Air-port	Average visibility	High		Moderate		Slight		Sum of high and moderate		Sum of all visibility	OI incursion
			visibility	impact	visibility	impact	visibility	impact	visibility	impact		
1	ATL	8.9	1,935	4,102	3,568	6,037	9,605	65				
2	BDL	8.9	2,124	6,531	4,877	8,655	13,532	8				
3	BOS	8.9	2,794	6,428	4,400	9,222	13,622	35				
4	BWI	8.8	1,617	5,338	5,419	6,955	12,374	17				
5	CLT	8.9	1,505	4,506	4,530	6,011	10,541	200				
6	DCA	9.1	594	2,640	4,966	3,234	8,200	33				
7	DEN	9.4	2,195	3,347	1,899	5,542	7,441	34				
8	DFW	9.5	649	2,359	2,171	3,008	5,179	58				
9	DTW	8.6	1,490	6,145	6,764	7,635	14,399	36				
10	EWR	8.9	1,322	5,034	4,600	6,356	10,956	38				
11	FLL	9.7	88	696	1,283	784	2,067	19				
12	HNL	9.9	10	189	445	199	644	31				
13	HOU	9.3	1,264	2,285	2,538	3,549	6,087	21				
14	IAD	8.8	1,888	5,376	5,340	7,264	12,604	22				
15	IAH	9.2	1,515	2,760	2,731	4,275	7,006	38				
16	JFK	9	2,812	3,772	3,651	6,584	10,235	41				
17	LAS	9.9	18	69	164	87	251	43				
18	LAX	8.4	1,743	5,124	7,716	6,867	14,583	41				
19	LGA	9.1	1,144	4,460	3,857	5,604	9,461	35				
20	MCO	9.5	910	1,766	2,177	2,676	4,853	9				
21	MDW	8.9	704	4,566	5,064	5,270	10,334	27				
22	MEM	9.4	534	2,305	2,778	2,839	5,617	11				
23	MIA	9.8	91	425	1,010	516	1,526	34				
24	MKE	8.7	2,514	5,632	6,081	8,146	14,227	18				
25	MSP	9.1	769	4,066	4,222	4,835	9,057	38				
26	ORD	9	1,097	4,240	5,191	5,337	10,528	108				
27	PHL	9.2	987	3,408	4,254	4,395	8,649	50				
28	PHX	9.9	28	72	183	100	283	20				
29	PVD	8.8	1,910	6,335	5,345	8,245	13,590	6				
30	SAN	8.9	732	2,379	4,777	3,111	7,888	6				
31	SDF	8.9	311	3,139	4,951	3,450	8,401	6				
32	SEA	9.2	2,578	3,101	2,941	5,679	8,620	16				
33	SFO	9.5	344	1,113	2,335	1,457	3,792	29				
34	SLC	9.4	1,443	3,232	2,778	4,675	7,453	25				
35	SNA	8.7	795	2,802	6,395	3,597	9,992	9				
36	STL	8.9	700	3,462	5,346	4,162	9,508	10				

Table B2.1: ANN prediction summary using 10 independent variables

Cases	R^2	
	Training	Validation
ANN with all dataset (24 training, 12 validation)	0.41	0.64
ANN with all dataset (30 training, 6 validation)	0.31	0.66
ANN without three least significant input parameters (30 training, 6 validation)	0.27	0.38

Note: The bold value represent the highest value from ANN analysis. It was used to compare the prediction performance between GAM and ANN.

warm support of the FAA technical center is appreciated. This research is also supported by the research funding of the Department of Civil, Construction, and Environmental Engineering of Iowa State University. Research funding from Black & Veatch is appreciated. The parallel computing research reported herein is partially supported by the HPC@ISU equipment at ISU, some of which has been purchased through funding provided by NSF under MRI Grant Nos. CNS 1229081 and CRI 1205413.

Bibliography

- Baesens, B. (2014). *Analytics in a big data world: The essential guide to data science and its applications*. John Wiley & Sons.
- Duchon, J. (1977). *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, pages 85–100. Springer, Berlin, Heidelberg.
- Efron, B. and Hastie, T. (2016). *Computer Age Statistical Inference*, volume 5. Cambridge University Press, New York, NY, USA.
- Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press, Boca Raton, FL, USA.
- FAA. Metar board. <https://www.aviationweather.gov/metar/help?page=board>. [Online; accessed 15-July-2016].
- FAA. Noaa ftp server. <ftp://ftp.ncdc.noaa.gov/pub/data/noaa/>. [Online; accessed 15-July-2016].
- FAA. Runway incursion totals by quarter fy2016 vs. fy2015. https://www.faa.gov/airports/runway_safety/statistics/year/?fy1=2016&fy2=2015. [Online; accessed 15-July-2016].
- FAA (2008). *Pilots Handbook of Aeronautical Knowledge*. US Department of Transportation-Federal Aviation Administration-Flight Standards Service, Oklahoma City, OK, USA.
- FAA (2015). National runway safety report 2013-2014.
- Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223.
- Gu, C. (2013). Smoothing spline anova models. *Springer Science and Business Media*.
- Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC Press, Boca Raton, FL, USA.
- Johnson, M. E., Zhao, X., Faulkner, B., and Young, J. P. (2016). Statistical models of runway incursions based on runway intersections and taxiways. *Journal of Aviation Technology and Engineering*, 5(2):3.
- Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library.
- Jones, D. R., Quach, C. C., and Young, S. D. (2001). Runway incursion prevention system-demonstration and testing at the dallas/fort worth international airport. In *Digital Avionics Systems, 2001. DASC. 20th Conference*, volume 1, pages 2D2/1–2D2/11 vol. 1. IEEE.

- Kam, W. Y., Pampanin, S., and Elwood, K. (2011). Seismic performance of reinforced concrete buildings in the 22 february christchurch (lyttelton) earthquake. *Bulletin of the New Zealand Society for Earthquake Engineering*, 44(4):239–278.
- Kamdar, H., Turk, M., and Brunner, R. (2016). Machine learning and cosmological simulations ii. hydrodynamical simulations. *Monthly Notices of the Royal Astronomical Society*, 457(2):11621179.
- Karlaftis, M. G. and Vlahogianni, E. I. (2011). Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, 19(3):387–399.
- Schnefeld, J. and Miller, D. (2012). Runway incursion prevention systems: A review of runway incursion avoidance and alerting system approaches. *Progress in Aerospace Sciences*, 51:31–49.
- Watnick, M. and Ianniello, J. W. (1992). Airport movement area safety system. In *11th Digital Avionics Systems Conf., IEEE/AIAA*, pages 549–552. IEEE.
- Wilke, S., Majumdar, A., and Ochieng, W. Y. (2015). The impact of airport characteristics on airport surface accidents and incidents. *Journal of safety research*, 53:63–75.
- Wood, S. (2006). *Generalized additive models: an introduction with R*. CRC Press.
- Wood, S. N. (2003). Thin plate regression splines. *the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114.
- Yu, H. (2002). Rmpi: parallel statistical computing in r. *R News*, 2(2):10–14.

CHAPTER 3. AN ADVANCED STATISTICAL APPROACH TO DATA-DRIVEN EARTHQUAKE ENGINEERING

A paper published in *Journal of Earthquake Engineering*, (2018)

Ikkyun Song, In-Ho Cho, and Raymond K. W. Wong

Abstract

Decades-long experimental databases become accessible in global earthquake engineering community. Yet, complex interactions of a multitude of variables pose formidable challenges to data-driven research. We embarked upon developing an advanced statistical learning and prediction framework with the generalized additive model (GAM). We showed promising performance of GAM with applications to existing RC shear wall databases. Without any prejudice, GAM can predict structural responses accurately using raw databases, and also can identify salient attributes. This study addresses computational implementation and parallel processing, and all codes are made publicly available to promote data-driven research of earthquake engineering community.

3.1 Introduction

In a broad spectrum of scientific and engineering fields, data-driven research is becoming a promising next-generation research paradigm. Notable advances in computing power enable researchers to draw valuable knowledge from data (e.g., drug design (Fishman, 1995), seismology (Caflich, 1998), and even cosmology (Kamdar et al., 2016b)). In relation to natural hazards, NSF has been persistent to construct community-level data nexus (e.g., Network for Earthquake Engineering Simulation hub (NEEShub) (Hacker et al., 2011)) and the new NSF Cyber-infrastructure for natural hazards engineering research (Rathje et al., 2017) that will be important foundations for future data-driven discovery, particularly in earthquake engineering community.

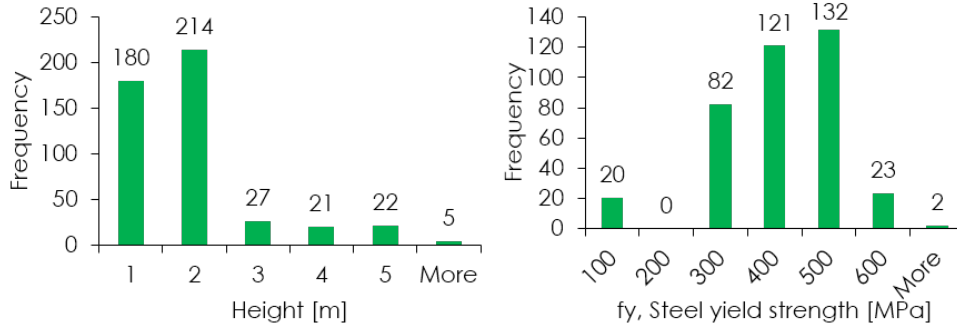


Figure 3.1: Sparseness and biasness revealed from 470 real experiments of RC shear wall database (collected from *NEESHub*, international reports, and literature)

Hitherto, data have not been actively used to improve the predictive and preventive ability of the earthquake engineering fields. Often, identification of problematic structural issues occurs at the post-disaster phase. Indeed, the earthquake engineering communities learned about hidden issues they were previously unaware after a natural hazard caused severe societal damage and claimed many lives. Two apt examples would be the surge of research on brittle steel structures after the 1994 Northridge earthquake and on the weak performance of reinforced concrete shear wall (RCSW) structures after the 2010 Chile and 2011 Christchurch earthquakes (Park and Chen, 2012). Real experiments are indispensable since they offer in-depth quantitative understandings of complex interplay among structural variables (geometric dimensions, materials and mechanical properties, etc.) and performance variables (load-carrying capacities, crack sizes, degree of crushing and buckling, etc.). But, limited financial resources prohibit real test-based approaches from unraveling the interdependency among salient variables. After successful real experiments, there remain considerable uncertainties, and more important, it is nearly infeasible to completely cover a full range of structural variables. Database quality raises another significant issue. Substantial biasness, sparseness, and missing values of real tests data pose a formidable challenge (e.g., Figure 3.1).

In the emerging era of data, this paper seeks to aid research community by finding a new remedy that is driven by and based on database. The novel objective of this paper is to apply an advanced and flexible non-parametric statistical technique to the well-established earthquake engineering

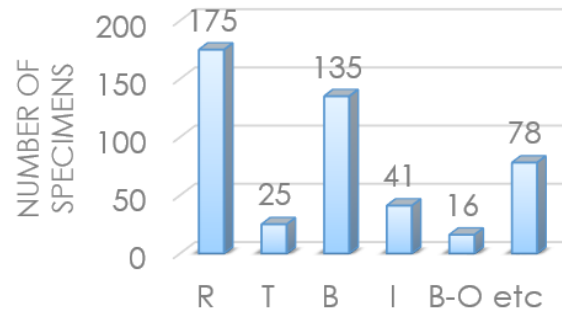


Figure 3.2: Number of specimens of each type of RCSW (R: rectangular; T: T-shaped; B: Barbell-shaped; I: I-shaped; B-O: Barbell-shaped with opening; etc.: all other types)

database: in particular, RC shear wall data. Our target wall type is rectangular walls since they constitute the majority of existing database (Figure 3.2). But it should be noted that the proposed methodology can be applied to other types of walls, which will be a future extension of this study. This paper demonstrates the promising capability of the data-driven approach and how we can rigorously predict untested structures' responses and hidden significance of some variables, notably directly from data. In particular, this paper expounds upon a non-parametric technique called generalized additive model (GAM), (e.g. (Hastie and Tibshirani, 1990)). As shall be addressed, the adopted GAM holds excellent accuracy and efficiency, and at the same time, allows remarkable flexibility in terms of the distributions of the response variable and its relationship to the predictor variables. Indeed, to tackle the complexity of multiple predictor variables is one of the key objectives of this study. As shown in Figure 3.3a, a variable may not show clear relationship with a target response of structures. But as we increase the number variables (Figure 3.3b), there may be a relationship in a form of curved surface or so. We regard this as the increasing interpretability. But, as we will show later, the more variables do not necessarily guarantee the increasing interpretability and/or predictability.

This paper holds notable contributions to research community. First, the use of GAM for data-driven prediction in earthquake engineering is novel since it is a glass-box approach. Second, it is of practical importance to systematically document how to harness GAM for selecting salient

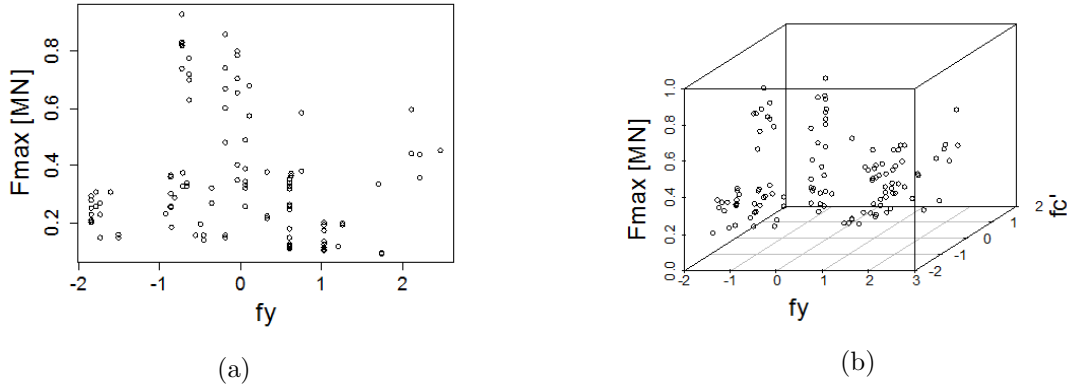


Figure 3.3: Change in the interpretability of database with increasing dimensionality: (a) two-dimensional (2D) scatter plot of standardized f_y (steel yield strength of longitudinal bars) and F_{max} (maximum shear force); (b) 3D plot of F_{max} , the standardized f_y , and the standardized f'_c (concrete strength). Some axes are unitless due to the standardized values

variables, learning raw data sets of earthquake engineering, and predicting practically important responses. Third, an introduction to a robust uncertainty estimation method for GAM prediction is noteworthy. All the developed statistical codes are shared via authors' research website (Cho, I., 2017). The earthquake engineering research community will benefit from the novel capabilities of the proposed statistical approach.

This paper is organized as follows: we introduce GAM and address its strength and theory. Three metrics for measuring the prediction power of GAM are presented. A cross-validation-based algorithm for finding the best predictor combination is presented. The prediction accuracy comparison between statistical prediction and high-precision computer simulation is addressed. The confidence interval of the response value predicted is presented for uncertainty estimation using a bootstrap method. A brief comparison of prediction performance of GAM and other statistical and machine learning methods is presented. Limitations of the proposed statistical prediction are summarized. The pseudo code is presented to explain the algorithm for the best predictor combination selection, followed by remarks on parallel computing. Full codes are presented in Appendix.

3.2 Summary of the Generalized Additive Model

A generalized additive model (GAM) (Hastie and Tibshirani, 1990) is a non-parametric extension of the well-known generalized linear model in which covariates enter the model through unspecified smooth functions. The general form of GAM can be given by:

$$g(\mu_i) = f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + \dots, \quad (3.1)$$

where g is a smooth link function; $\mu_i \equiv \mathbb{E}(Y_i \mid \mathbf{x}_i)$; Y_i is a response variable and from some exponential distribution family (e.g., normal, binomial, or gamma distribution); \mathbf{x}_i is i^{th} vector of data points comprising multiple variables, $\mathbf{x}_i = \{x_{1i}, x_{2i}, \dots\}$; f_j are smooth functions of covariates x_{ji} (Wood, 2006). For instance, Y_i would be the maximum shear force of i^{th} RC shear wall specimen and $\mathbf{x}_i = \{\text{length}_i, \text{height}_i, \text{AxialForce}_i, \dots\}$.

The GAM is a non-parametric regression model, which depends on sum of unspecified smooth functions rather than pre-specified forms of x_i . This leads to the flexible nature of GAM, and distinguishes GAM from the commonly used linear models. GAM can be constructed to predict complex data accurately whereas linear models can be only used for data with linear relationship. To effectively glean the central notion of the GAM, the following descriptions involve one variable and normal distribution case. Extension to multiple variables and other distributions are straightforward, and details can be found in (Wood, 2006). Now, the GAM becomes $\mathbb{E}(Y \mid x)$, and the smooth function f can be approximated as follows:

$$f(x) = \sum_{j=1}^k b_j(x)\beta_j \quad (3.2)$$

where $b_j(x)$ is the j th basis function and β_j is an unknown parameter. Fitting of the model can be accomplished by maximizing the corresponding likelihood with a penalty term represented by $\lambda \int [f''(x)]^2 dx$ where λ is *smoothing parameter*. An over smoothed estimate is attributed to too large λ value while an under smoothed estimate is done by too small λ value. The error between spline estimate \hat{f} and true function f is large in the both extreme cases. We can choose λ value which enables to fit model appropriately by minimizing generalized cross validation (GCV) score

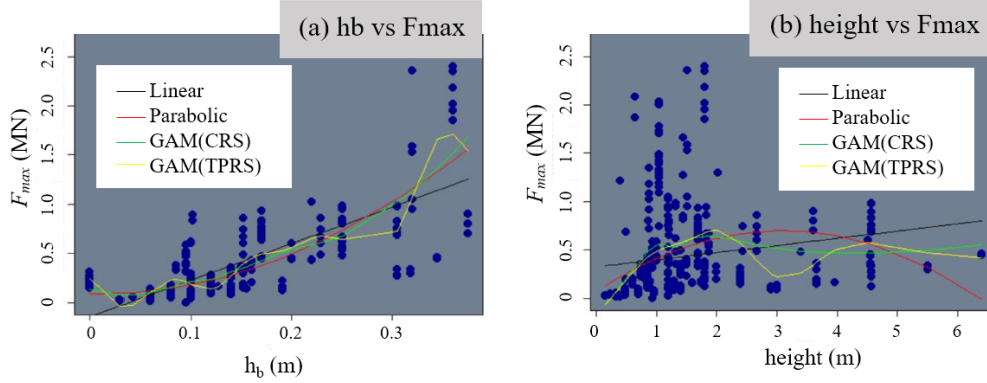


Figure 3.4: Example of one-dimensional regressions of 470 real RC wall data: (a) h_b (thickness of boundary element) versus F_{max} ; (b) wall height versus F_{max}

(Golub et al., 1979). The smallest GCV score is achieved by selecting an optimum λ value via the relevant GAM library (Wood, 2001).

A basis for spline should be chosen to construct a GAM. There are two popular types of basis used in GAM: (a) thin plate regression splines (TPRS) (Wood, 2003) and (b) Cubic regression spline (CRS) (Wood, 2006). TPRS is suitable for any number of covariates and notably, "knot-free" (i.e. requiring no knot location selection). Yet, CRS requires knot location selection and is restricted to a single variable. In general, TPRS requires more computational cost than CRS. As an illustrative example, Figure 3.4 compares four regression models (i.e., Black = Linear; Red = Parabolic; Green = GAM(CRS); Yellow = GAM(TPRS)) with 470 real RC shear wall data. Figure 3.4 presents a good example of the flexibility of GAM when applied to the complex real-world database.

On one hand, cubic spline is a curve constructed by combining a number of cubic polynomial sections. Those sections join at a certain point, called knot, of which location should be pre-selected for the cubic spline basis. The cubic polynomial sections are joined such that the entire spline becomes continuous up to second derivative. Although somewhat different from the practical regression splines (see (Wood, 2006)), to help grasp a sense of relevant mathematical forms, some cubic spline functions (Gu, 2013) are given by

$$b_1(x) = 1, b_2(x) = x, \text{ and } b_{i+2} = R(x, x_i^*) \text{ for } i = 1, 2, \dots, p - 2 \quad (3.3)$$

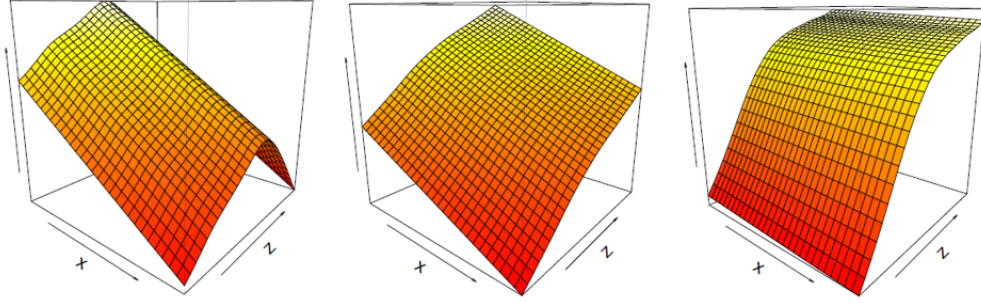


Figure 3.5: Example of thin plate spline basis function using 2 covariates

where p is number of rank for basis, x_i^* is knot location, and,

$$R(x, x_i^*) = \left[(x_i^* - 1/2)^2 - 1/12 \right] \left[(x - 1/2)^2 - 1/12 \right] / 4 \\ - \left[(|x - x_i^*| - 1/2)^4 - 1/2(|x - x_i^*| - 1/2)^2 + 7/240 \right] / 24. \quad (3.4)$$

On the other hand, thin plate spline (Duchon, 1977) can be used for multiple covariates. Thin plate spline function, \mathbf{f} , can be obtained by minimizing

$$\|\mathbf{y} - \mathbf{f}\|^2 + \lambda J_{md}(f), \quad (3.5)$$

where \mathbf{y} is the vector of y_i data and $\mathbf{f} = [f(x_1), f(x_2), \dots, f(x_n)]^T$. $J_{md}(f)$ is a penalty functional measuring the 'wiggleness' of f , and λ is a smoothing parameter, controlling the tradeoff between data fitting and smoothness of f . The wiggleness is related to the degree of flatness. If f is too wiggled (i.e., overfitted), then the function curve is highly up and down in a short range while the function curve becomes nearly flat (too smoothed) when f is not wiggled. Both cases lead to poor prediction. The wiggleness penalty is defined as

$$J_{md} = \int \cdots \int_{R_d} \sum_{v_1 + \cdots + v_d} \frac{m!}{v_1! + \cdots + v_d!} \left(\frac{\partial^m f}{\partial x_1^{v_1}, \dots, \partial x_d^{v_d}} \right)^2 dx_1, \dots, dx_d. \quad (3.6)$$

One example of thin plate spline basis function with 2 covariates is shown in Figure 3.5. In Equation 3.7, \hat{f} is the function that can minimize Equation 3.5. As marked in Equation 3.7, the first terms are related to wiggleness while the second terms are independent of wiggleness.

$$\hat{f}_{\mathbf{x}} = \underbrace{\sum_{i=1}^n \delta_i \eta_{md}(\|\mathbf{x} - \mathbf{x}_i\|)}_{\text{wiggly components}} + \underbrace{\sum_{i=1}^M \alpha_j \phi_j(\mathbf{x})}_{\text{zero wiggly terms}} \quad (3.7)$$

where δ_i and α_j are coefficients to be determined, ϕ_j are linearly independent polynomials spanning the null space of J_{md} , and the basis functions η_{md} are given by

$$\eta_{md}(h) = \begin{cases} \frac{(-1)^{m+1+d/2}}{2^{2m-1} \pi^{\frac{d}{2}} (m-1)! (m-\frac{d}{2})!} h^{(2m-d)} \log(h) & \text{for } d = \text{even} \\ \frac{\Gamma(\frac{d}{2}-m)}{2^{2m} \pi^{\frac{d}{2}} (m-1)!} h^{(2m-d)} & \text{for } d = \text{odd} \end{cases} \quad (3.8)$$

Thin plate regression splines seek to find the balance by reducing the wiggly components of Equation 3.7 and retaining the zero wiggly terms in Equation 3.7. In this fashion, thin plate regression splines are regarded as a powerful approximation method that has little restriction to the burdensome knot location determination and many variables. Detailed formulations and explanation can be found in relevant literature (e.g., (Wood, 2006)).

3.3 Metrics for Prediction Comparisons

In this study, we adopted several metrics to quantitatively compare the prediction performances: Ratio between base cross validation error (CVE_b) and cross validation error (CVE), (CVE_b/CVE), Pearson correlation, ρ , and coefficient of determination, R^2 were adopted to measure how accurately the GAM fits. This choice of metrics is based on the comparable study on machine learning comparisons (Baesens, 2014; Kamdar et al., 2016,b). The higher CVE_b/CVE , ρ , and R^2 , the more accurate prediction. The CVE and CVE_b are defined as,

$$CVE = \frac{1}{N} \sum_{i=1}^N (y_{ex}^i - y_{pr}^i)^2; \quad CVE_b = \frac{1}{N} \sum_{i=1}^N (y_{ex}^i - y_{mean,pr})^2, \quad (3.9)$$

where N is number of data, y_{ex}^i is the i^{th} measured response, y_{pr}^i is i^{th} predicted response according to the cross-validation procedure described later in Figure 3.6, and $y_{mean,pr}$ is the mean of predicted values. ρ and R^2 are defined as

$$\rho = \frac{COV(y_{pr}, y_{ex})}{\sigma_{y_{pr}} \times \sigma_{y_{ex}}}; \quad R_2 = 1 - \frac{\sum_{i=1}^N (y_{ex}^i - y_{mean,pr})^2}{\sum_{i=1}^N (y_{ex}^i - y_{pr}^i)^2}. \quad (3.10)$$

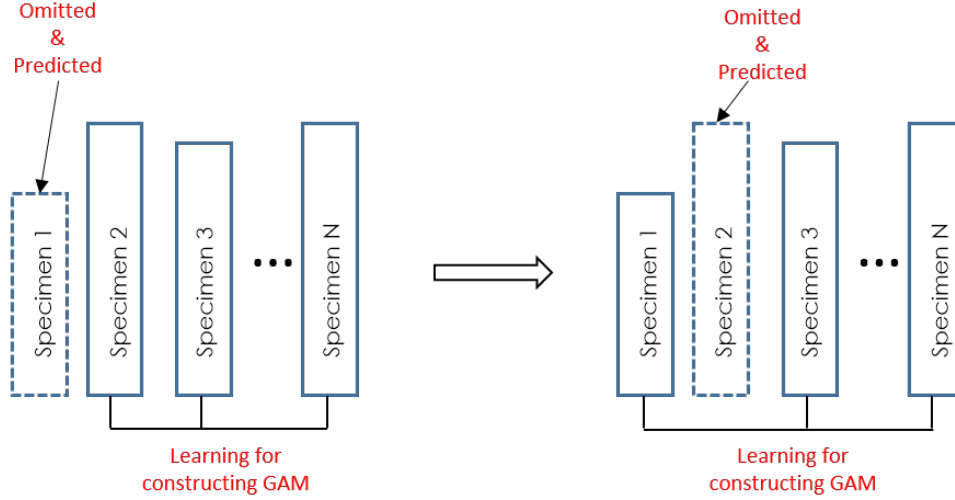


Figure 3.6: Illustration of cross validation: left figure represents that first specimen’s data is omitted. A GAM is constructed by learning all other wall data; right figure shows the same procedure by omitting the second wall data

For comparison among different statistical models, we mainly use CVE_b/CVE , and also used other metrics as additional reference.

3.4 Prediction with GAM

To demonstrate the strong predictive power of GAM, we used the existing database of RC shear wall experiments. In particular, GAM seeks to predict the maximum shear force of rectangular walls. Basic statistical setting is as follows. This study assumes the Gamma distribution in light of the domain-specific nature of data (i.e., real, positive values in existing values). For the link function among variables, we chose *log* since it can easily incorporate multiplicative connections of engineering variables. For the smooth function-related setting, k (i.e. the number of basis dimensions in smooth functions), we adopted typical value of 7 throughout the statistical studies as recommended by literature (e.g. (Wood, 2006)). It should be stressed that this internal setting $k = 7$ is not the number of predictors (variables or parameters) of regression, but how many bases per smooth functions of GAM. It should be noted that smoothing parameter λ is readily optimized in terms of GCV in the relevant library of R .

To demonstrate the flexibility and accuracy of the data-driven statistical prediction, we began with raw data of 10 variables, yet resorted to no prejudices regarding relations or relative significance of variables. The 10 variables in the existing database are: axial force ratio (denoted by afr), wall thickness (thickness), boundary element's thickness (h_b) and width (w_b), wall height (height), wall length (length), primary reinforcing bar's yield strength (fy) and diameter (dia), concrete compressive strength (fc), and boundary element reinforcement ratio (rr_b). Target response is the rectangular shear wall's maximum shear resistance, F_{max} .

Key steps consist of three tasks: (1) exclusion of a test wall specimen, (2) construction of a GAM by learning the remaining wall data, and (3) prediction of the test wall's response. In the cross validation, one of wall data (so-called test sample) is excluded intentionally while and other wall data (learning samples) are used to construct the GAM (Figure 3.6). Thereafter, the maximum shear force of the omitted wall is predicted using the GAM. These processes are repeated throughout all wall data. To systematically assess the prediction power, we used the cross validations. The difference between the predicted F_{max} from GAM and the original value of the omitted wall specimen directly represents how precisely the constructed GAM can predict the target response.

To systematically present the predicted results, the so-called Q-Q plots were used to compare the scaled response of real experiment and predicted value (see Figure 3.7). Importantly, although the statistical models use no prejudices or weighting factors, the predicted responses using two GAMs show good agreements with real experimental data. The promising accuracy is commonly found in both GAM(CRS) and GAM(TPRS). It should be noted that all the statistical predictions in Figure 3.7 are made by the "best" statistical models that only utilized the raw data. As shall be described in the next section, in terms of the prediction, all variables are not necessary, and the combination of too many variables may even decrease the predictive power. Since we departed from no prejudice regarding which variables should be included or excluded in the GAM construction, next section shall describe how we can find the "best" combination of a certain set of variables.

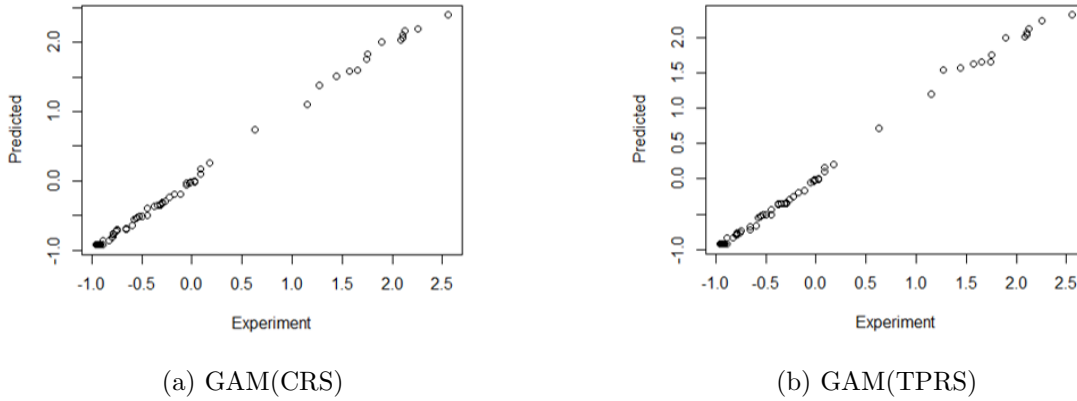


Figure 3.7: Q-Q plot of real experimental data and the predicted value using (a) GAM(CRS); (b) GAM(TPRS). Both axes are unitless owing to the standardized values

3.5 Constructing a Best GAM with a Given Number of Variables

Since we don't resort to any prejudice, the present approach should provide a remedy to how to construct a successful GAM. Challenge is that we are uncertain which variables should be included in the GAM. Indeed, GAM can be constructed using an arbitrarily many number of variables, but prudent selection of variables has a critical impact on GAM. In this study, CVE_b/CVE , Pearson correlation, and the coefficient of determination were used to evaluate how many variables should be selected for GAM. In particular, we departed from variables of existing rectangular wall database in hopes of finding the "best" GAM model that can accurately predict the maximum shear force. We first constructed all possible combinations of variables, and in each case we compared CVE_b/CVE to determine the best combination. In terms of CVE_b/CVE , Table 3.1 and 3.2 summarize the best combination of given number of variables. For instance, among all possible combinations of two variables, GAM(CRS) selects height and h_b (second row of Table 3.1) as best combination. It should be noted that these comparisons are focusing on only the prediction accuracy of the given statistical setting and assumptions. In parentheses, we included the calculated p-value corresponding to the variable.

Overall, Table 3.1 and Figure 3.8 show that the best combination for GAM(CRS) is the combination of six variables: i.e., axial force ratio (afr), wall thickness (thickness), thickness of boundary

Table 3.1: Selection of the best combination of variables for GAM using CRS (p-values in parentheses)

Number of variables	Number of combination	Best combination of variables			CVE_b / CVE	Pearson	R^2
2	45	height(6.24e-11)	hb(1.85e-05)		12.24	0.958	0.918
3	120	height(<2e-16)	hb(3.71e-11)	dia(0.00272)	16.39	0.969	0.939
4	210	height(<2e-16) dia(1.57e-08)	afr(3.11e-13)	hb(5.51e-10)	21.00	0.976	0.952
5	252	height(<2e-16) hb(5.59e-06)	afr(1.73e-13) fc(0.292)	dia(5.51e-08)	22.46	0.978	0.955
6	210	afr(<2e-16) height(9.51e-08)	thickness(<2e-16) fy(7.01e-08)	hb(1.27e-11) dia(3.26e-06)	26.21	0.981	0.962
7	120	afr(<2e-16) height(1.01e-07) fc(0.719))	thickness(<2e-16) fy(2.69e-07)	hb(1.76e-11) dia(4.00e-06)	25.75	0.981	0.961
8	45	afr(<2e-16) wb(3.07e-10) dia(7.38e-05)	height(<2e-16) length(6.60e-09) hb(0.163)	fy(<2e-16) thickness(1.9e-08)	24.64	0.980	0.959
9	10	afr(<2e-16) wb(7.85e-10) dia(9.89e-05)	height(<2e-16) thickness(5.37e-08) hb(0.171)	fy(<2e-16) length(1.00e-08) fc(0.707)	23.61	0.979	0.958
10	1	afr(<2e-16) wb(5.63e-08) dia(0.00999) fc(0.726)	height(<2e-16) length(2.58e-07) hb(0.105)	fy(1.15e-13) thickness(2.0e-06) rrb(0.644)	4.63	0.918	0.784

Table 3.2: Selection of the best combination of variables for GAM using TPRS (p-values in parentheses)

Number of variables	Number of combination	Best combination of variables			CVE_b / CVE	Pearson	R^2	
2	45	length(5.91e-11)	height(1.59e-09)		12.22	0.958	0.918	
3	120	length(<2e-16)	dia(<2e-16)	afr(2.11e-11)	15.70	0.968	0.936	
4	210	length(<2e-16)	height(<2e-16)	afr(1.18e-13)	20.89	0.976	0.952	
		dia(1.51e-11)						
5	252	afr(<2e-16)	thickness(2.06e-09)	fy(3.24e-07)	23.32	0.978	0.957	
		rrb(1.43e-06)	length(0.00033)					
6	210	afr(<2e-16)	thickness(3.76e-09)	fy(9.12e-07)	22.92	0.978	0.956	
		rrb(2.17e-06)	length(0.00044)	fc(0.84103)				
7	120	afr(<2e-16)	height(4.53e-05)	fy(0.000306)	24.33	0.979	0.959	
		thickness(6e-04)	dia(0.002263)	hb(0.010451)				
		length(0.211003)						
8	45	afr(<2e-16)	height(<2e-16)	fy(<2e-16)	22.97	0.979	0.956	
		length(1.40e-05)	thickness(0.0152)	hb(0.1574)				
		dia(0.232)	rrb(0.682)					
9	10	afr(<2e-16)	length(<2e-16)	wb(5.34e-08)	23.93	0.979	0.958	
		fy(1.21e-07)	height(9.44e-04)	rrb(0.0183)				
		dia(0.730)	thickness(0.768)	fc(0.793)				
10	1	afr(<2e-16)	wb(6.25e-05)	height(3.68e-04)	14.88	0.968	0.933	
		fy(8.65e-04)	hb(0.001342)	dia(0.248)				
		length(0.700)	thickness(0.771)	rrb(0.876)				
		fc(0.889)						

element (h_b), wall height (height), primary reinforcing bar’s yield strength (f_y) and diameter (dia). Likewise, Table 3.2 and Figure 3.8 also show that the best combination for GAM(TPRS) has the seven variables: i.e., afr , thickness, h_b , height, f_y , dia, and wall length (length). Interestingly in both GAM cases, axial force ratio was identified as the statistically important variable. Indeed, nearly all the best combinations in Tables 3.1 and 3.2 suggest to include the axial force ratio. It is interesting to notice that this solely data-driven approach also pinpoints the importance of axial force ratio raised by many researchers’ mechanics-based investigations (Okamura et al., 1975; Qian et al., 2008). In some cases, reinforcement ratio at boundary element (rr_b) and concrete compressive strength (f_c) are identified as important. In the second column of Tables 3.1 and 3.2, the number of possible combinations was shown by simple calculations: e.g., if 4 variables are to be selected from 10 total variables, the number of total combinations is $10!/4!(10 - 4)! = 210$. It is noteworthy that the best combination of variables can be different when another statistical model or new dataset are used during constructing a statistical prediction model. Still, this study’s method and approach are meaningful by providing how to harness the accuracy and flexibility of the statistical predictions for systematic data-driven investigation.

3.6 Statistical Prediction VS. High-Precision Computer Simulations

This section addresses important analogy and difference between the statistical prediction and high-precision computer simulations in the earthquake engineering. For the several decades, computer simulations have served as a successful tool for ”prediction” of responses of complex RC structures under seismic loading. Earthquake engineering community has made coordinated efforts to derive high fidelity computational simulation platform such as *OpenSees* (McKenna et al., 2000).

Also, many researchers developed various simulation programs (Cho, 2013; Cui et al., 2010; Orakcal and Wallace, 2006; Sobhaninejad et al., 2011).

On one hand, it is instructive to compare the analogy between predictions by computer simulations and statistical predictions. Both can be used to reproduce responses of real experiments to a certain level of errors. They commonly can be used to predict responses of untested speci-

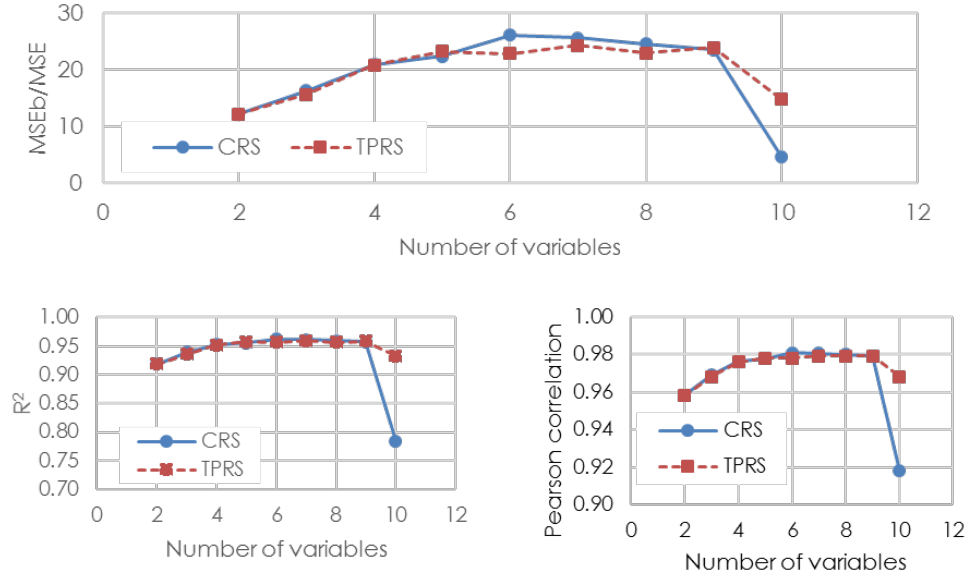
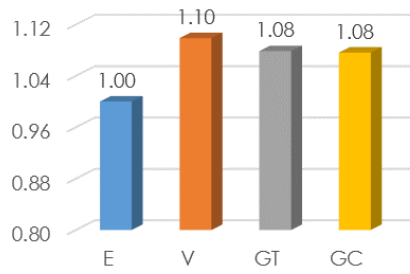
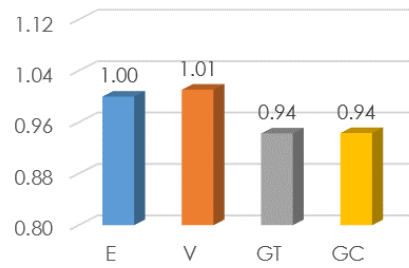


Figure 3.8: Illustration of cross validation: left figure represents that first specimen’s data is omitted. A GAM is constructed by learning all other wall data; right figure shows the same procedure by omitting the second wall data

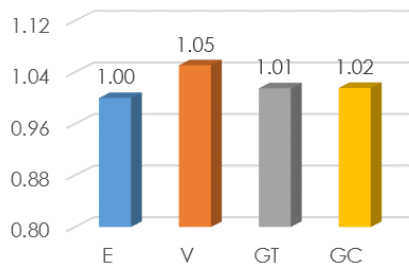
mens with varying parameters (e.g., material strengths or features of reinforcements). We adopted Virtual Earthquake Engineering Laboratory (VEEL) as high-precision computational simulation (Cho, 2013). Figure 3.9 summarizes prediction errors of high-precision simulations and statistical predictions. VEEL generally exhibits a stable range of errors regardless of wall specimens. VEEL’s error ranges are less than 5% for most walls (except for WSH5 of 8% error). On the contrary, the error ranges of the statistical predictions appear to fluctuate: i.e., the error ranges of WSH1, 2, and 5 are less than 5% while other three walls are higher than 5%. The higher error rates of the statistical prediction of WSH wall series may be attributed to the unusual characteristics of the walls. As shall be addressed in the Limitation of statistical prediction section later, these walls have relatively unique geometric features compared to typical walls, and thereby the walls are situated at the boundary of the database. Hence, less data learning was carried out, which appears to cause large error in statistical prediction.



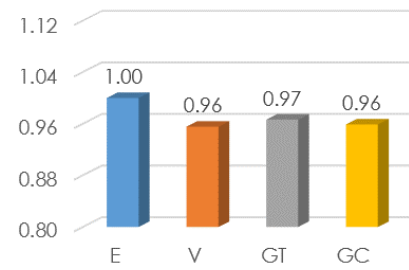
(a)



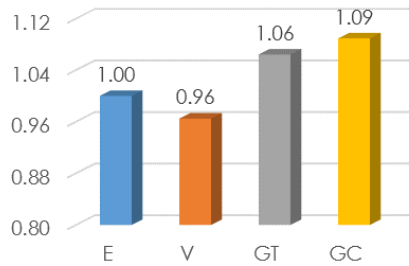
(b)



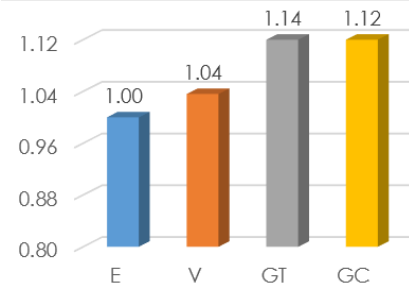
(c)



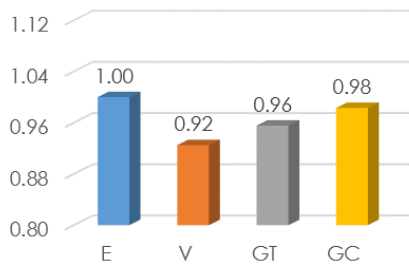
(d)



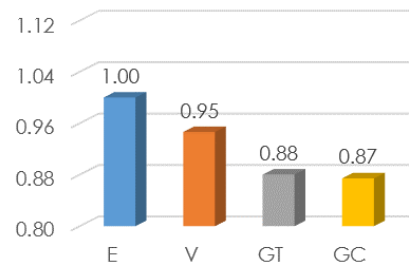
(e)



(f)



(g)



(h)

Figure 3.9: Normalized maximum shear force of experiment (E), VEEL (V), and GAM using TPRS (GT) and CRS (GC) of RW1 and RW2 (Vulcano et al., 1988), and WSH1 through WSH6 (Orakcal and Wallace, 2006). (Note: The value of vertical axis represents the maximum shear force normalized by experimental result; thus, "E" has always one)

On the other hand, there exists important difference. Computer simulations are built upon well-proven engineering principles and relationships among a few parameters. Contrarily, statistical predictions are rooted in implicit interrelations among parameters (i.e. predictors) and pre-specified definitions of relationships are unnecessary. Thus, statistical predictions directly focus on the hidden interrelations of given data. Another difference arises from diversity of prediction results. Computer simulations can predict various responses spanning macroscopic (global force, displacement, etc.) and microscopic behaviors (crack sizes, bar buckling, etc.) while statistical predictions are restricted to observed responses. As seen in Figure 3.10, computer simulations can reproduce continuous responses while statistical predictions often related to discrete values since the observation data are discrete. Computational costs are different. High-precision computer simulations often require expensive computational cost for solving governing equations of the physical problems whereas statistical predictions need relatively cheap calculations.

Importantly, both methods share the common limitation. Basically, computer simulation is a general tool capable of analyzing various geometrical, material, and loading conditions. However, when a new specimen contains substantially innovative materials or structural conditions, computer simulations may need to update engineering principles and constitutive relationships, requiring new real tests and validations. Likewise, statistical predictions may not be suitable for predicting considerably new specimen (i.e., variables are substantially outside the range of existing database). In statistics, this limitation is well known as extrapolation problem. Thus, both methods essentially require real experiments to advance their frontiers. To some extent, computational simulations, statistical predictions, and real experiments should be in a cross-fertilizing relationship for data-driven earthquake engineering.

3.7 Uncertainty Estimation

Statistical prediction naturally includes uncertainty for several reasons. To briefly explain how to incorporate the uncertainty behind the proposed statistical prediction, this section evaluates the "prediction interval" to quantitatively measure uncertainty in GAM prediction. Confidence interval

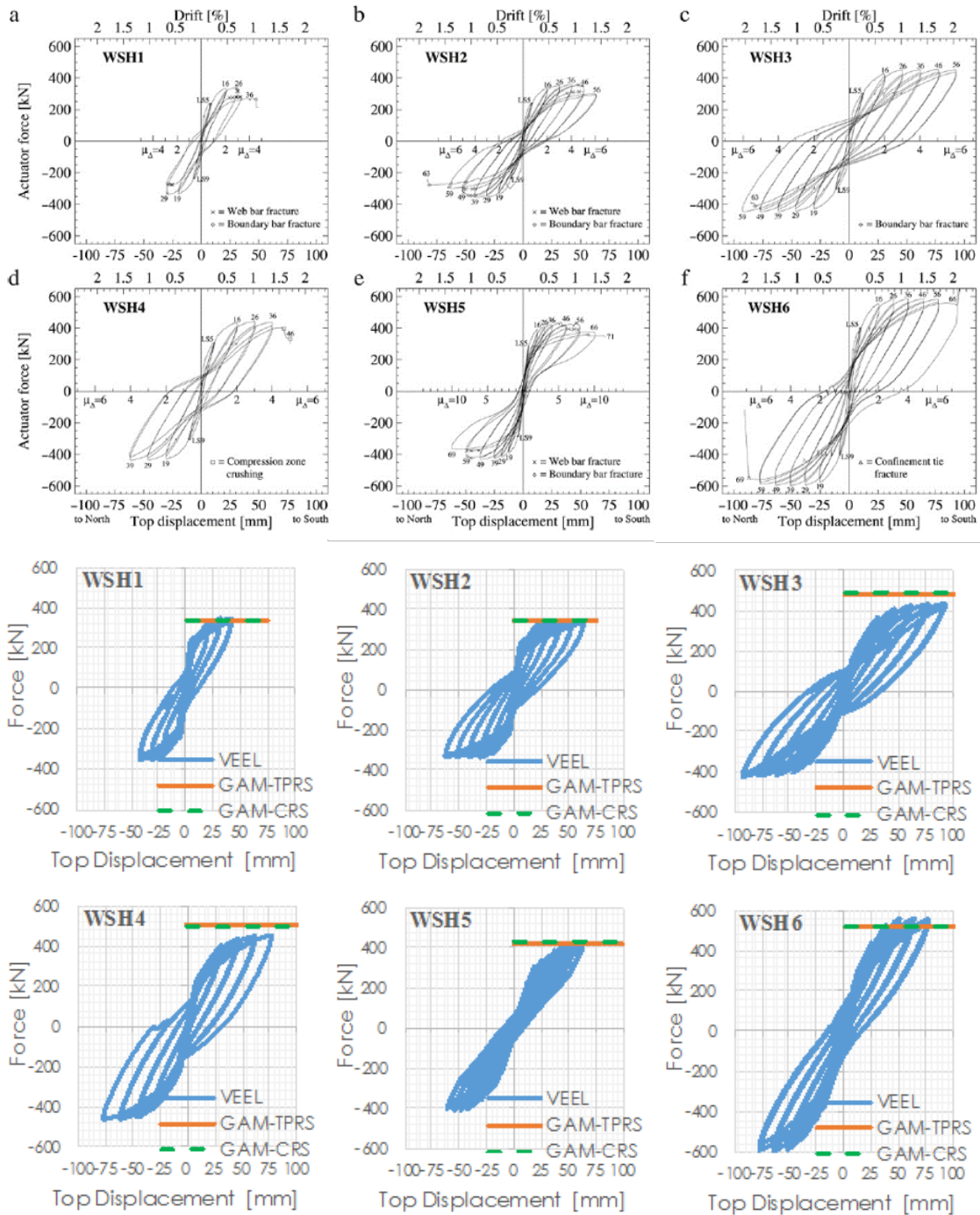


Figure 3.10: Prediction accuracy comparison between high-precision computational simulation (VEEL) and statistical prediction (GAM) result using WSH series: (Top 6 panels) experimental results cited from Orakcal and Wallace (2006); (Bottom 6 panels) prediction results from VEEL, GAM-TPRS, and GAM-CRS. Note that the maximum force is the comparison target

of WSH series' F_{max} , predicted using GAM with the best seven predictor variables, is determined by using the percentile method with bootstrapping (Efron and Tibshirani, 1994). A synthetic sample response set, y_i^* ($i = 1, \dots, \text{sample size}$), is generated by resampling centred residuals. Another GAM is learned by using y_i^* and bootstrapped predicted responses, y_j^* ($j = 1, \dots, B$) are generated, where B is the bootstrap size. We used $B = 2000$ in the current study. The detailed procedure for bootstrapping can be found in authors' previous work (Song et al., 2018b). The confidence interval of WSH series' F_{max} , is estimated from the bootstrap samples by using the percentile method (Efron and Hastie, 2016). The cumulative distribution function of bootstrap samples, \hat{G} , less than b can be represented as

$$\hat{G}(b) = \mathcal{F}\{\hat{y}_i^* \leq b\} / B, \quad i = 1, \dots, B, \quad (3.11)$$

where \mathcal{F} is frequencies of y_i^* . A point having a specific percentile (α) can be obtained by

$$\hat{y}^{*(\alpha)} = \hat{G}^{-1}(\alpha), \quad (3.12)$$

where \hat{G}^{-1} is the inverse function of \hat{G} . The 95% confidence interval is represented by

$$(\hat{y}^{*(0.025)}, \hat{y}^{*(0.975)}). \quad (3.13)$$

The 95% confidence interval of F_{max} for WSH series (i.e., WSH 1 through WSH 6) is shown in Figure 3.11. Circle and x mark depict measured F_{max} and a median value of bootstrap sample, respectively. The measured F_{max} values are located within confidence intervals except WSH 6. The confidence intervals look relatively wide and this may be attributed to unusual features (e.g., geometry) of WSH wall series. These wide confidence intervals will be shortened and the predictive power will be improved when more predictor variables and ample databases are included to GAM prediction in the future extensions.

3.8 Difference from Traditional Statistical Methods

Various statistical methods have been widely used for researches in earthquake engineering. Amongst many, regression analyses are one of the popular methods. The notable difference of the

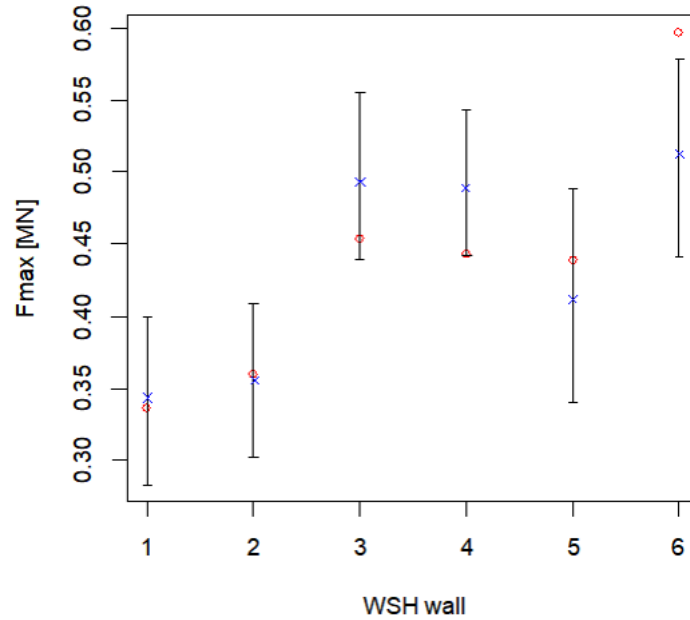


Figure 3.11: 95% confidence interval of WSH wall series' F_{max} estimated from GAM prediction using bootstrap method. Circle and "x" mark represents measured F_{max} and a median value of bootstrap samples, respectively

proposed approach from previous methods is twofold: first, the present statistical prediction allows unspecified relationships among variables of database, and the learning process is solely based on the raw data and a flexible additive model assumption. Second, the present statistical learning and prediction tasks have little restriction to the number of variables.

Traditionally, statistical methods are usually used to confirm a researcher's pre-defined relationship of a set of pre-selected variables. In particular, after prudently selecting a few variables, a researcher seeks to establish a combination of the variables to best match the target response. For instance, traditional statistical methods are used to confirm a relation describing the maximum shear strength of reinforced concrete shear wall (denoted as F_{max} hereafter). Such relationships are well reflected in the design codes (e.g., (ASCE, 2010)). However, in this study, we assumed no previous knowledge on the variables' relationship and their relative importance on the response. In essence, for a given response the proposed statistical approach seeks to find the hidden relationship

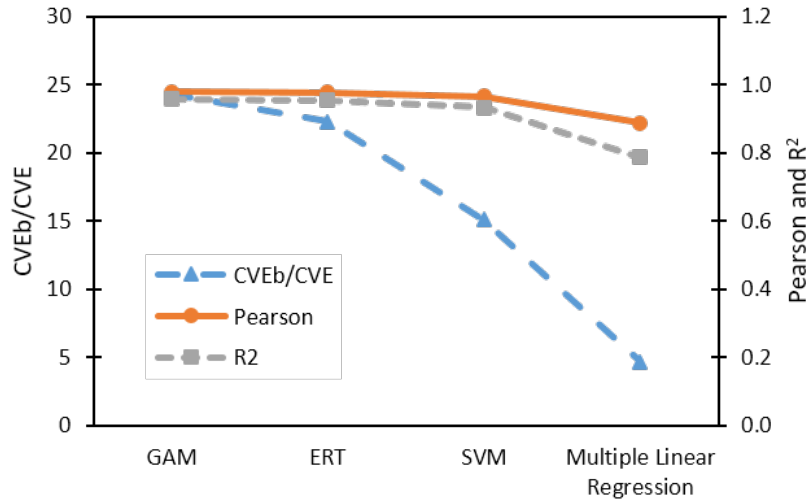


Figure 3.12: Prediction power comparison of GAM against other popular prediction methods

of variables and also significance of variables. To some extent, we seek to achieve and examine a purely data-driven discovery.

In addition to these novel advantages, it is instructive to compare the relative performance of GAM in relation to other well-known prediction methods. In view of popularity, we selected a multiple linear regression, extremely randomized trees (ERT) (Geurts et al., 2006), and support vector machine (SVM) (Cortes and Vapnik, 1995). For the comparison, we used the same dataset with seven predictor variables, which are selected in the previous section. The relative prediction power is summarized in Figure 3.12. GAM exhibits comparable predictive power to ERT and slightly better than SVM. Also, GAM appears to outperform traditional multiple linear regression. It should be noted that optimization of ERT and SVM may improve their performance, and a generalization of this comparative study requires due consideration.

3.9 Limitation of Statistical Prediction

As addressed so far, the statistical learning and prediction are solely based on data. Little relationship is assumed in the learning process. Naturally, the limitation of the statistical prediction stems from the quality of data. Missing data or corrupted values are critical. In particular, if an

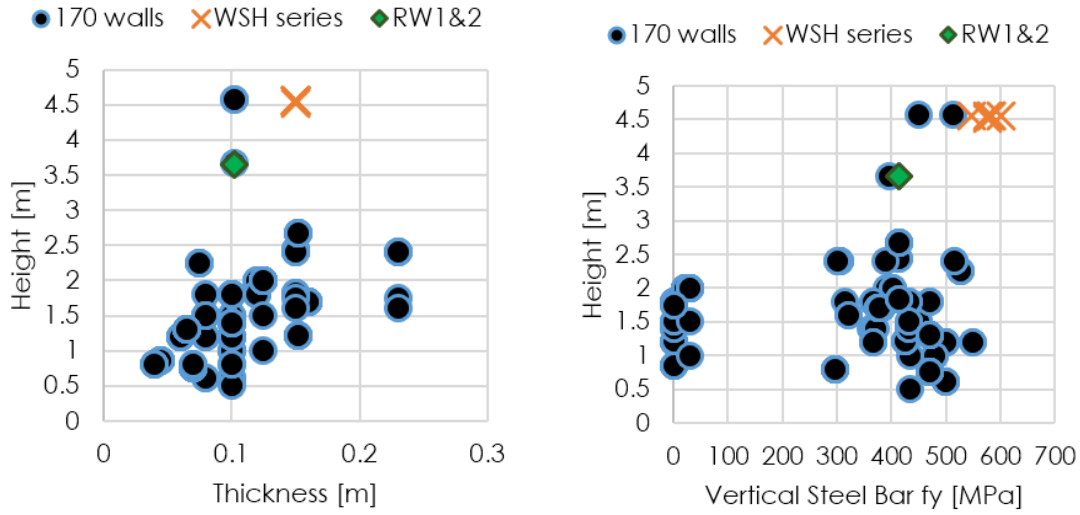


Figure 3.13: Scatter plot of rectangular RCSW specimens showing the ranges of database. WSH wall series occupy the boundary of the database

experimental data has no axial force ratio, the specimen cannot be used for the learning process involving the axial force ratio. It is a substantial loss. Furthermore, if the database has little information of a certain type of structures, the statistical prediction tends to perform poorly; the less data to learn, the less reliability of prediction. Thus, special care should be paid when the statistical prediction is used for predicting a specimen of which attributes are not within the range of the existing database.

To quantitatively explain this issue, we performed two case studies: (1) Statistical prediction after excluding WSH 1 through WSH 6; (2) including WSH series. For each case, we used VEEL, GAM-TPRS and GAM-CRS for predicting the maximum shear strengths F_{max} of RW1, RW2, WSH1 and WSH6. Here, RW1 and RW2 represent typical rectangular RCSW in the database while WSH1 and WSH 6 represent special wall types residing on the boundary of database. Indeed, some structural features of WSH series occupy the boundary of existing database of rectangular RCSWs: e.g., WSH series have the length of $2m$ and height of $4.56m$ that are larger than those of a majority of other 170 walls in the existing database (Figure 3.13).

Table 3.3 presents responses predicted by VEEL, GAM-TPRS, and GAM-CRS without data of WSH series. Table 3.4 shows the prediction results after including WSH series data. In Tables 3.3 and 3.4, numbers close to 1.0 imply accurate predictions. As expected, regardless of the inclusion of WSH series data, the high-precision simulation by VEEL appears to consistently generate accurate prediction since VEEL is based on engineering principles and physical mechanisms (second rows in Tables 3.3 and 3.4). Contrarily, in the statistical learning process, the exclusion of WSH series substantially weakens the accuracy of the statistical prediction (see columns of WSH1 and WSH6 in Table 3.3). Both GAM-TPRS and GAM-CRS exhibit poor prediction of WSH1 and WSH6 with TPRS being worse. Particularly for this data sets, TPRS appears to severely deteriorate without the information of WSH series than CRS (i.e., TPRS’s prediction score is 4.50 for WSH1 and 24.44 for WSH6 while CRS’s prediction score is 0.58 and 0.60 for WSH1 and WSH6, respectively). This case study well describes the extrapolation problem in statistics. However, it should be noted that the both GAM-TPRS and GAM-CRS accurately predict RW1 and RW2 (see columns of RW1 and RW2 in Table 3.3) even without learning WSH series data. Since RW1 and RW2 are typical wall types in the existing database, the statistical prediction is based on successful learning on other similar wall types. Indeed, when there are ample learning data sets, statistical predictions perform comparably or slightly better than VEEL simulation (e.g., compare scores of RW1 and RW2 in Table 3.3).

Table 3.3: Predictions *without* WSH series (F_{max} is normalized by that from experiment)

<i>Without WSH series</i>	RW1	RW2	WSH1	WSH6
VEEL/Experiment	1.10	1.01	1.05	0.95
GAM-TPRS/Experiment	1.00	1.00	4.50	24.44
GAM-CRS/Experiment	0.99	0.99	0.58	0.60

After including WSH series data in statistical models (Table 3.4), the accuracy of both statistical prediction methods (GAM-TPRS and GAM-CRS) is notably improved for WSH series. Especially, the prediction error substantially decreases in the WSH1 case (compare fourth columns in Tables

Table 3.4: Predictions *with* WSH series (F_{max} is normalized by that from experiment)

<i>Without WSH series</i>	RW1	RW2	WSH1	WSH6
VEEL/Experiment	1.10	1.01	1.05	0.95
GAM-TPRS/Experiment	1.08	0.94	1.01	0.88
GAM-CRS/Experiment	1.08	0.94	1.02	0.87

3.3 and 3.4). However, there appears to be a trade-off. In this case, the statistical models need to cover wider ranges of database, and thus the prediction error of RW1 and RW2 cases slightly increases compared to the cases without WSH series. This may be attributed to the fact that TRPS and CRS are smooth functions rather than a perfectly discrete function. Therefore, it is natural to see that additional new data points affect the learning process (regressions) built upon previous data points, albeit slightly.

3.10 R Code for Constructing GAM by Cross-Validation

This section addresses the pseudo *R* code to construct a best GAM with three variables by using cross validation. The full code is shown in Appendix. Appendix A contains the stand-alone version *R* code and Appendix B contains parallel version of *R&Rmpi*. A brief explanation of the codes is as follows.

Note that `Parameter_1` is variable name of the first parameter, `ColumnNumber_1` is integer representing column number for the corresponding parameter. In case of four variables, for example, one more iteration should be added after line 13 and four variables should be included in dataset in line 5 6.

In Table 3.6, we explained how to make a parallel version of constructing a best combination of 3 variables. In the explanation, we assumed 3 slave processors on a high performance computing cluster (named *Condo* cluster) of Iowa State University. To launch and test the provided *Rmpi*

Table 3.5: Description of the stand-alone R code (see Table 3.1 in Appendix)

Line	Explanation of stand-alone code
1	Import "mgcv" library to use <i>gam</i> and <i>predict.gam</i> functions
2	Set working directory in which input and output file are located
3	Read input data from the specified working directory
4	Build dataset which is used for constructing GAM
5	Excludes data which has null value in the specified parameter
6	Generates parameter (i.e., covariate) variable name for output file
7	Initiates a matrix to save data from running
8	Initiates a matrix to save a dataset within main loop
9	Total number of predictor variables
10	All possible combination
11	Number of all possible combination
12	Index vector for main loop
13	Changes column name in dataset for each parameter combination
14	Main loop
15-18	Builds dataset for each parameter combination
19	Vector to save max force predicted
20	Iteration for prediction
21	Make dataset for one intentionally omitted data
22	Make dataset using all data except the omitted data
23	Construct GAM
24-26	Generates data frame which is used for prediction
27	Predicts a response value
28	Save the predicted value to the specified vector for output
29	Calculates mean of predicted response value
30	Calculates CVE
31	Calculates CVEb
32	Determine ratio of CVEb and CVE
33	Determine Pearson correlation, ρ
34	Determines coefficient determination, R^2
35-36	Make output
37	Write output file into the designated working directory

Table 3.6: Description of the parallel version of *R&Rmpi* code (see Table 3.2 in Appendix)

Line	Explanation of <i>Rmpi_main.R</i>
1-2	Import libraries for parallel R (<i>Rmpi</i>) and for <i>gam</i> and <i>predict.gam</i> functions (<i>mgcv</i>)
4	Spawn 3 slaves. Note that the current R script is defined on Master processor.
5	Set working directory in which input and output file are located
6-7	Initialize variables on Master processor
8-9	Initialize variables on Slave processors
10	On all slaves, load the user-defined function named "serial_function.R"
11	Get processor id of Master (i.e. 0)
12	Get processor id's of Slaves (i.e., 1 total slaves)
13	Get total available processors on Master (in this example, 4)
14	Get total available processors on Slaves
15	On all slaves, perform parallel tasks using the user-defined function. Local arguments passed to slaves (e.g., slave 2 will have <i>id</i> =3 and <i>total_proc</i> = 4). Results are stored at <i>output_slaves</i> .
16	User must define their own data gathering command here (e.g., <i>mpi.allgather</i> , etc.)
17	Close all slaves
18	Finalize the parallel tasks
	Explanation of <i>serial_function.R</i> used in <i>Rmpi_main.R</i>
1-3	On each slave, explicitly load <i>Rmpi</i> library and other necessary libraries
4-8	The same as the stand-alone version (see corresponding explanation in Table 3.5)
9-10	Initiates vectors to save data from running
11	Get the total number of slaves (e.g., in this example, 3)
12	Create a cyclic index sequence starting from the slave id (e.g., in slave 2, <i>c_x3</i> = [2, 5, 8, ...])
13-16	The same loops of the stand-alone version (see corresponding explanation in Table 3.5)
17-18	Cyclic job allocation on the last loop
19-	Remainder of the code is the same as the stand-alone version (Table 3.5)

code, one needs to successfully install *OpenMPI* and *Rmpi* libraries on their own computing facility (for installation, see [Yu 2002; <http://www.stats.uwo.ca/faculty/you/Rmpi/>]).

3.11 Remarks on Parallel Processing of *R* & *Rmpi* Code

As seen in the code in Appendix, finding the best combination of an unknown set of variables is computationally intensive. For a large number of variables, the computational cost may pose a challenge. Therefore, we developed an algorithm-oriented parallel computing algorithm using *Rmpi* (Yu, 2002).

Rmpi is a general wrapper that enables R codes to utilize message passing interface over multiple processors. Since *Rmpi* only provides a general environment, this study developed a problem-oriented parallel computing algorithm for the proposed statistical learning and prediction. To ensure effective load balancing, we used the so-called cyclic allocation of the task throughout the slave processors. In particular, this study proposes an algorithm that allows one master processor that can flexibly spawn a number of slave processors. The master processor controls the entire work (e.g., distribution of tasks and collection of results) while the slaves do an assigned work and return outputs to the master. In light of the decreasing computational loads (Figure 3.14), a successful parallelization scheme would be a cyclic job allocation over the slaves. As the problem size increases, this cyclic allocation approaches the optimal parallel load balancing (Cho and Hall, 2012). Figure 3.15 shows a summary of parallel computing performance of the proposed parallel algorithm. The parallel algorithm appears to achieve the reasonable scalability up to 4 slaves, with 3 slaves being the best. But, study revealed that with more than 4 slaves, the parallel performance began to deteriorate due to internal communication overhead and load imbalance. Further elaboration on the present problem-optimized parallel algorithm will be carried out in the future researches.

3.12 Conclusions

In this paper, we expounded upon an advanced statistical approach that can facilitate data-driven researches in the earthquake engineering fields. In particular, the generalized additive model

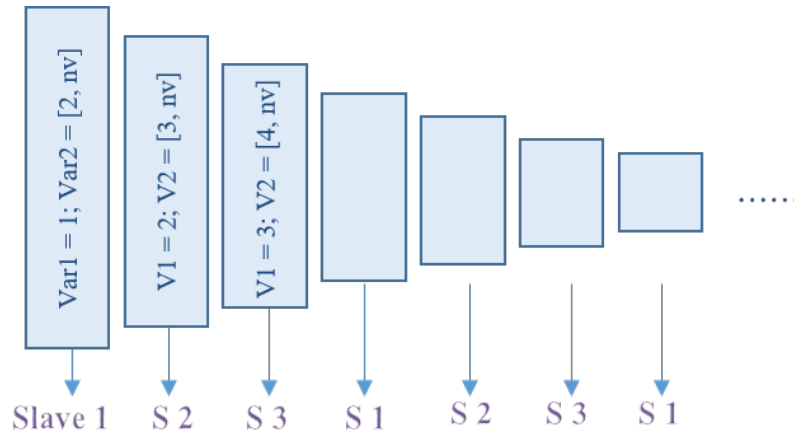


Figure 3.14: Cyclic allocation of the proposed parallel code of *R* & *Rmpi*. Two-variable case is shown with *nv* meaning the total number of variables. Height of box corresponds computation load

(GAM) has been studied and applied to RC shear wall database. Two popular smoothing functions, thin plate regression spline (TPRS) and cubic regression spline (CRS), are systematically examined. Validations and applications to real-world earthquake engineering database revealed a promising capability of the statistical prediction. Compared to the high-precision computer simulation results, the statistical prediction appears to hold reasonable accuracy in reproducing responses of a wide range of RC shear wall specimens. Computationally, the statistical approach appears to be superior over high-precision computer simulations. Notably, those predictions were made without pre-specified relationships among variables of database. Results suggest that as far as statistical prediction accuracy is concerned, not all variables (i.e. structural attributes) are necessary, which implies there may exist relative significances among some attributes. The currently suggested prediction model can be improved with inclusion of more predictor variables and databases, which will be available as a web-based framework to the earthquake engineering community in the near future, and this will help researchers and engineers obtain prediction result with an acceptable accuracy within a short time. Detailed code and parallel computing algorithm are presented. As community-level database continues to evolve, the proposed statistical learning and prediction approaches will shed light on the new data-driven discovery in earthquake engineering fields. All the

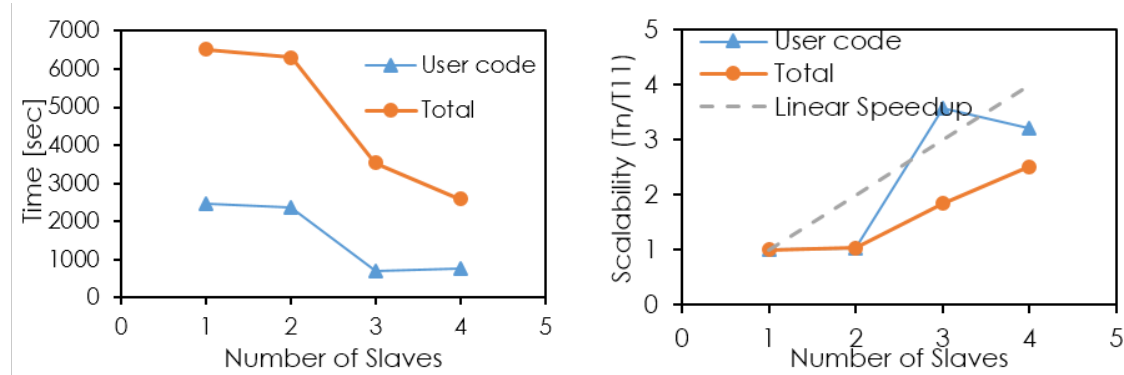


Figure 3.15: Parallel computing performance of *R* & *Rmpi* code for finding the best 5-variable combination out of 252 total combinations. "User code" means the time spent on execution of user-defined codes while "Total" means the total elapsed wall clock time of the parallel code (attained from *proc.time()* of *R*)

developed codes will be shared upon request to the authors. All the statistical (*R* and *Rmpi*) codes developed in this work are made publicly available at (Cho, I., 2017).

3.A Appendix

R codes are shown in Tables A3.1 and A3.2.

3.13 Acknowledgments

This research is supported by the research funding of Department of Civil, Construction, and Environmental Engineering of Iowa State University. Generous research funding from Black & Veatch is appreciated. The simulations of this paper is partially supported by the HPC@ISU equipment at Iowa State University, some of which has been purchased through funding provided by NSF under MRI grant number CNS 1229081 and CRI grant number 1205413. Special thanks are due to Professor John F. Hall for his productive guidance regarding nonlinear analysis methods, and also to Professor Sri Sritharan for valuable discussion on earthquake engineering experiments.

Table A3.1: R code for constructing a best GAM using TPRS (3-variable combination)

```

1  library(mgcv)
2  setwd("WorkingDirectory")
3  import = read.csv(file="InputDataName.csv", head=TRUE, sep=",")
4  dataset = data.frame(Parameter_1=import[,ColumnNumber_1],
5  Parameter_2=import[,ColumnNumber_2], Parameter_3=import[,ColumnNumber_3])
6  dataset = subset(dataset, dataset[ColumnNumber_1] != "NA" &
7  dataset[ColumnNumber_2] != "NA" & dataset[ColumnNumber_3] != "NA")
8  label.x = c("ParameterName_1", "ParameterName_2", "ParameterName_3")
9  output=NULL
10 dataset_combi = NULL
11 nPredictor = length(dataset)-1
12 combination = combn(nPredictor, nCombi)
13 lenCombi = length(combination)/nCombi
14 index = as.integer(seq(1, lenCombi))
15 char = c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10")
16 for (x in index)
17   dataset_combi = data.frame(dataset[ncol(dataset)])
18   colnames(dataset_combi)[1] = "y1"
19   for (y in 1:nCombi)
20     dataset_combi[,char[y]] = dataset[combination[y,x]]
21   maxforce_pred=vector()
22   for(z in 1:nrow(dataset_combi))
23     dataset1 = dataset_combi[c(z),]
24     dataset2 = dataset_combi[-c(z),]
25     fit = gam(y1 ~ s(x1, k=7)+s(x2, k=7)+s(x3, k=7), data=dataset2,
26     family=Gamma(link=log))
27     dataset_pred = data.frame(dataset1[1])
28     for (w in 1:nCombi)
29       dataset_pred[,char[w]] = dataset1[w+1]
30     predicted = predict.gam(fit, newdata=dataset_pred, type="response", se=TRUE)
31     maxforce_pred[z] = predicted$fit
32     mean_pred = mean(maxforce_pred)
33     cve = sum((maxforce_pred-dataset_combi$y1)^2)/length(maxforce_pred)
34     cveb = sum((dataset_combi$y1-mean_pred)^2)/nrow(dataset_combi)
35     cveb.cve = cveb/cve
36     pearson = cov(maxforce_pred, dataset_combi$y1)/sd(maxforce_pred)/sd(dataset_combi$y1)
37     R2 = 1-cve/cveb
38     predictor = paste(label.x[combination[,x]],collapse=' ')
39     output = rbind(output, data.frame(predictor,cveb.cve, pearson, R2))
40 write.csv(output,file="FileName.csv")

```

Table A3.2: *Rmpi&R* code for constructing a best GAM using TPRS (3-variable combination and 3 slaves)

	<i>Rmpi_main.R</i>
1	library(Rmpi)
2	library(nlme)
3	mode = "TPRS"
4	setwd("WorkingDirectory")
5	mpi.spawn.Rslaves(nslaves = 3)
6	mpi.bcast.cmd(source("serial_function.R"))
7	id=mpi.comm.rank()
8	mpi.bcast.cmd(id=mpi.comm.rank())
9	nSlaves=mpi.comm.size()-1
10	mpi.bcast.cmd(nSlaves=mpi.comm.size()-1)
11	mpi.bcast.cmd(output_slave=serial_function(2, mode,id,nSlaves))
12	mpi.bcast.cmd(output_slave=serial_function(3, mode,id,nSlaves))
13	output_master1 = NULL
14	output_master2 = NULL
15	for (x in 1:nSlave)
16	result1=mpi.recv.Robj(x,1)
17	result2=mpi.recv.Robj(x,2)
18	output_master1 = rbind(output_master1,result1)
19	output_master2 = rbind(output_master2,result2)
20	write.csv(output_master1,file=" FileName1.csv")
21	write.csv(output_master2,file=" FileName2.csv")
22	mpi.bcast.cmd(q("no"))
23	mpi.quit()
	<i>serial_function.R</i>
1	library(mgcv)
2	dataset = data.frame(Parameter_1=import[,ColumnNumber_1],
3	Parameter_2=import[,ColumnNumber_2], Parameter_3=import[,ColumnNumber_3])
4	dataset = subset(dataset, dataset[ColumnNumber_1] != "NA" &
5	dataset[ColumnNumber_2] != "NA" & dataset[ColumnNumber_3] != "NA")
6	label_x = c("ParameterName_1", "ParameterName_2", "ParameterName_3")
7	output=NULL
8	dataset_combi = NULL
9	nPredictor = length(dataset)-1
10	combination = combn(nPredictor, nCombi)
11	lenCombi = length(combination)/nCombi
12	each = as.integer(lenCombi / nSlaves)
13	if (id==nSlaves) index = as.integer(seq(1+each*(id-1), lenCombi))
14	else index = as.integer(seq(1+each*(id-1), each*id))
	#use the same code as the serial R code in Table 3.1 of Appendix from line 13 to line 36
	return (output)

Bibliography

- ASCE (2010). *Minimum design loads for buildings and other structures*, volume 7. Amer Society of Civil Engineers.
- Baesens, B. (2014). *Analytics in a big data world: The essential guide to data science and its applications*. John Wiley & Sons.
- Caffisch, R. E. (1998). Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49.
- Cho, I. H. (2013). Virtual earthquake engineering laboratory capturing nonlinear shear, localized damage and progressive buckling of bar. *Earthquake Spectra*, 29(1):103–126.
- Cho, I. H. and Hall, J. F. (2012). Parallelized implicit nonlinear fea program for real scale rc structures under cyclic loading. *Journal of Computing in Civil Engineering*, 26(3):356–365.
- Cho, I. (2017). Data-driven computational science and engineering. <https://sites.google.com/site/ichoddcse2017/home/Computational-Sci--Eng/gam-for-earthquake-eng>.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- Cui, Y., Olsen, K. B., Jordan, T. H., Lee, K., Zhou, J., Small, P., Roten, D., Ely, G., Panda, D. K., and Chourasia, A. (2010). Scalable earthquake simulation on petascale supercomputers. In *High Performance Computing, Networking, Storage and Analysis (SC), 2010 International Conference for IEEE*, pages 1–20. IEEE.
- Duchon, J. (1977). *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, pages 85–100. Springer, Berlin, Heidelberg.
- Efron, B. and Hastie, T. (2016). *Computer Age Statistical Inference*, volume 5. Cambridge University Press, New York, NY, USA.
- Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press, Boca Raton, FL, USA.
- Fishman, G. (1995). *Monte Carlo: concepts, algorithms, and applications*. Springer Science & Business Media, New York.
- Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.
- Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223.
- Gu, C. (2013). Smoothing spline anova models. *Springer Science and Business Media*.

- Hacker, T. J., Eigenmann, R., Bagchi, S., Irfanoglu, A., Pujol, S., Catlin, A., and Rathje, E. (2011). The neeshub cyberinfrastructure for earthquake engineering. *Computing in Science & Engineering*, 13(4):67–78.
- Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC Press, Boca Raton, FL, USA.
- Kamdar, H., Turk, M., and Brunner, R. (2016a). Machine learning and cosmological simulations ii. hydrodynamical simulations. *Monthly Notices of the Royal Astronomical Society*, 457(2):11621179.
- Kamdar, H. M., Turk, M. J., and Brunner, R. J. (2016b). Machine learning and cosmological simulationsi. semi-analytical models. *Monthly Notices of the Royal Astronomical Society*, 455(1):642–658.
- McKenna, F., Fenves, G., Scott, M., and Jeremic, B. (2000). Open system for earthquake engineering simulation (opensees).
- Okamura, H., Watanabe, K., and Takano, T. (1975). Deformation and strength of cracked member under bending moment and axial force. *Engineering Fracture Mechanics*, 7(3):531–539.
- Orakcal, K. and Wallace, J. W. (2006). Flexural modeling of reinforced concrete walls-experimental verification. *ACI Structural Journal*, 103(2):196.
- Park, J. and Chen, Y. (2012). Understanding and improving the seismic design of shear walls. final year projects. Technical report, Dept. of Civil and Natural Resources Engineering, University of Canterbury.
- Qian, J., Wei, Y., Zhao, Z., Cai, Y., Yu, Y., and Shen, L. (2008). Experimental study on seismic behavior of src shear walls with high axial force ratio. *Journal of Building Structures*, 29(2):43–50.
- Rathje, E. M., Dawson, C., Padgett, J. E., Pinelli, J.-P., Stanzione, D., Adair, A., Arduino, P., Brandenberg, S. J., Cockerill, T., and Dey, C. (2017). Designsafe: New cyberinfrastructure for natural hazards engineering. *Natural Hazards Review*, 18(3):06017001.
- Sobhaninejad, G., Hori, M., and Kabeyasawa, T. (2011). Enhancing integrated earthquake simulation with high performance computing. *Advances in Engineering Software*, 42(5):286–292.
- Song, I., Cho, I., Tessitore, T., Gursik, T., and Ceylan, H. (2018). Data-driven prediction of runway incursions with uncertainty quantification. *Journal of Computing in Civil Engineering*, 32(2):04018004.
- Vulcano, A., Bertero, V. V., and Colotti, V. (1988). Analytical modeling of r/c structural walls. In *Proceedings of the 9th World Conference on Earthquake Engineering*, volume 6, pages 41–46.
- Wood, S. (2006). *Generalized additive models: an introduction with R*. CRC Press.
- Wood, S. N. (2001). mgcv: Gams and generalized ridge regression for r. *R news*, 1(2):20–25.
- Wood, S. N. (2003). Thin plate regression splines. *the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114.
- Yu, H. (2002). Rmpi: parallel statistical computing in r. *R News*, 2(2):10–14.

CHAPTER 4. EFFICIENT VARIABLE SELECTION METHODS FOR ADVANCED STATISTICAL LEARNING AND PREDICTION OF RIGID PAVEMENT SYSTEMS

A paper submitted and under review for publication in *Journal of Transportation Research Board*,
(2018)

Ikkyun Song, Sunghwan Kim, Halil Ceylan, and In-Ho Cho

Abstract

Since the pavement conditions are closely related to the driving conditions and serious accidents, accurate prediction of pavement responses has been an important research area for pavement design and management plan. Advanced statistical prediction can deal with non-linear, multivariate data offering meaningful statistical information such as relative importance of variables. Given an advanced statistical model, variable selection (VS) task is often used to identify the optimal combination of some variables which enables the prediction model to achieve its best performance. However, VS for multivariate, large data sets is computationally challenging. This study seeks to investigate efficient VS methods that can swiftly lead to the best prediction performance of the generalized additive model (GAM). Recently, GAM is recognized as advanced statistical models due to its non-parametric, flexible, and unspecified multivariate-friendly smooth functions. We investigated several VS (i.e., backward, forward, and direct) methods for GAM using two practical pavement data sets (i.e., pavement internal stress and overlay data). Results suggest that a backward selection using Akaike information criterion appears to hold similar efficiency as the exhaustive direct VS, being order of magnitude fast. As anticipated, resultant p-values help elucidate the relative importance of selected variables in the prediction.

4.1 Introduction

As of 2016, there are public roads of 4,140,108 miles and vehicle miles of travel are over 3 trillion in the United States (FHWA, 2017). Most of the public roads are paved for safety, driving quality, maintenance, etc. The performance of pavement is directly related to the safety of drivers and passengers because a poor condition of pavements such as distresses could induce an abnormal driving condition, leading to major transportation fatalities and serious injuries. Proper pavement design and management plan are indispensable to prevent these accidents.

To achieve this goal, the prediction of pavement responses and performances is important because it helps engineers to understand the underlying relationship between explanatory variables and responses and better establish a plan for a pavement design and maintenance. There have been many efforts to predict pavement responses and performances using statistical methods. For example, Salama et al. (2006) used single and multiple linear regression to investigate the impact of truck-related parameters, such as axle and truck types, on pavement responses. Heba and Assaf (2017) used a Bayesian linear regression method to predict missing part of historical data for a pavement performance model.

There also have been machine learning (ML) approaches. Ceylan et al. (1998, 1999) developed an artificial neural network (ANN) model for a jointed concrete airfield pavement to investigate pavement responses such as stresses and deflections. Gopalakrishnan and Kim (2011) used a support vector machine (SVM) to predict hot mix asphalt stiffness. They found that the stiffness prediction performance of SVM with less controlling parameters for optimizations was comparable to the ANN. Tabatabaee et al. (2013) developed two-stage pavement performance prediction strategy. They used a support vector classifier first and a recurrent neural network in the next stage to increase the prediction accuracy.

ML methods are convenient to use and provide decent prediction performance; however, the pathway between input and output in ML is unclear, which makes researchers feel difficulty interpreting prediction results. On the other hand, prediction of statistical models is based on statistical theories and methodologies, which helps researchers better understand the relationship between in-

put and output and build a better predictive model based on their knowledge about the data (Cho et al., 2018). Another advantage of statistical methods is that they can identify the relative importance of predictor variables in predictive models. Although simple or multiple linear regression methods are handy, but their prediction performance becomes worse when predicting highly non-linear data.

To tackle this issue, we adopted an advanced statistical model, generalized additive model (GAM) (Hastie and Tibshirani, 1990), to predict pavement responses and performances accurately, elucidate the best predictor variables, and provide the relative importance of predictor variables. GAM is a non-parametric statistical method in which covariates enter the model without any prejudices or assumptions on variables. GAM covers a wide range of statistical distributions, enabling to accurately predict complex pavement data with substantial nonlinearity whereas a simple linear model can be used only for data with a linear relationship. The noticeable prediction performance of GAM was demonstrated by (Song et al., 2017, 2018c,b). The detailed theory and advantages of GAM shall be explained in the later section.

Objectives of this study are to (1) introduce an advanced statistical method, GAM to the pavement research community and apply GAM to predict pavement responses (i.e., stresses) and performances (i.e., International Roughness Index (IRI)), (2) investigate efficient variable selection methods for the best statistical prediction, (3) elucidate the relative importance of the best predictor variables, and (4) suggest the optimal GAM setting.

This paper is organized as follows: we introduce the GAM and describe its central notion and strength. Practical data sets used for GAM prediction are briefly explained. Several variable selection strategies are addressed, and their relative prediction performances are compared to the direct search method. An optimized number of spline bases for accurate GAM prediction is also investigated.

4.2 Overview of Generalized Additive Model

Generalized additive model (GAM) (Hastie and Tibshirani, 1990) is a generalized linear model, holding substantial flexibility and general applicability. Rather than using predefined parameters or distributions, GAM leverages unspecified smooth functions. By the flexible feature of unspecified smooth functions, covariates do not need to have a set of fixed parameters. The general form of GAM can be given by:

$$g(\mu_i) = f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + \dots, \quad (4.1)$$

where g is a smooth link function; the expectation $\mu_i \equiv \mathbb{E}(Y_i | \mathbf{x}_i)$; Y_i is from some exponential family of distribution (e.g., normal, binomial, or gamma distribution); f_j are smooth functions of covariates x_{ji} (Wood, 2006). For example, Y_i would be the maximum tensile stress of i^{th} pavement sample and x_i could be thickness, modulus, etc. In essence, the GAM can have multiple unspecified smooth functions and each smooth function have one covariate. For a concise explanation of the central notion of the GAM, the following description only involve a single covariate. Extension to the multivariate case is straightforward, and details can be found in (Wood, 2006). Let the GAM be $\mathbb{E}(Y | x)$, and the smoothing function f can be given by:

$$f(x) = \sum_{j=1}^k b_j(x)\beta_j \quad (4.2)$$

where $b_j(x)$ is the j^{th} basis function and β_j is an unknown parameter. Model fitting can be done by maximizing the corresponding likelihood with a penalty term which is represented as:

$$\lambda \int [f''(x)]^2 dx \quad (4.3)$$

where λ is *smoothing parameter*. λ is internally optimized by GAM to balance smoothness of regression and accuracy of prediction. The optimum λ value can be chosen in such a way that the model fits accurately by minimizing the generalized cross-validation (GCV) score (Golub et al., 1979). There are two popular spline bases for smooth functions of GAM: (a) thin plate regression spline (TPRS) and (b) cubic regression spline (CRS). For details of TPRS and CRS, one can refer to (Wood, 2006). For implementing GAM, a publicly opened *R* library, *mgcv* (Wood, 2011) is used.

4.3 Description of Pavement Databases for Model Development

Two pavement databases utilized in this study, namely as *concrete overlay* and *rigid airport pavements*, were retrieved from authors' recently completed Iowa Highway Research Board (IHRB) Project TR-698 "Concrete Overlay Performance on Iowa's Roadways" (Gross et al., 2017) and from authors' on-going project Federal Aviation Administration (FAA) Project "Implementing a Multiple-Slab Response Model for Top-Down Cracking Mode in Rigid Airport Pavements" (Kaya et al., 2018; Rezaei-Tarahomi et al., 2018), respectively. These databases were used to find the best predictor variables for accurate predictions and identify the relevant importance of best predictor variables in GAM. While the detailed descriptions on *concrete overlay* database are found in (Gross et al., 2017) and the detailed descriptions on *rigid airport pavements* database are found in (Kaya et al., 2018; Rezaei-Tarahomi et al., 2018), the information related to GAM developments for both databases is summarized herein.

The *concrete overlay* database is Iowa concrete overlay historical database (Gross et al., 2017) having about 1,130 data records including information of concrete overlay type, distress, pavement age, climatic related data, and IRI for about 380 concrete overlay projects totaling about 1,490 miles in Iowa. 25 explanatory variables are used as predictor variables and IRI is predicted using GAM. Depending on variable combinations, we investigate four different cases: (1) *case 1* (all variables), (2) *case 2* (variables without climatic related variable), (3) *case 3* (variables without distress variables), and (4) *case 4* (variables without distress and climatic related variables). The best predictor variables for these four cases will be identified using GAM.

The *rigid airport pavements* database is rigid airport structure and response synthetic database obtained by using a 3D-FE computer program called Finite Element Analysis-FAA (FEAFAA) to develop a surrogate computational response model or procedure suitable for implementation next generation of airport pavement design procedures under on-going project FAA Project "Implementing a Multiple-Slab Response Model for Top-Down Cracking Mode in Rigid Airport Pavements" (Kaya et al., 2018; Rezaei-Tarahomi et al., 2018). The *rigid airport pavements* database includes information of rigid airport pavement structures, materials, mechanical and thermal loading condi-

Table 4.1: Variable description of concrete overlay and rigid airport pavements data

Data (No. of sample)	Variable (abbreviation)	
<i>Concrete overlay</i> (1,133)	Low severity transverse cracks (TRANS.L)	High severity right wheel path faulting (RT_FT_SEV3)
	Moderate severity transverse cracks (TRANS.M)	Very high severity right wheel path faulting (RT_FT_SEV4)
	High severity transverse cracks (TRANS.H)	Overlay type (TYPE)
	Moderate severity D-cracks (DCRACK.M)	Overlay thickness (THICK)
	High severity D-cracks (DCRACK.H)	Joint spacing (JT.SP)
	Moderate severity joint spalls (JSPALL.M)	Age (AGE)
	High severity joint spalls (JSPALL.H)	Traffic (TRAFFIC)
	Low severity left wheel path faulting (LT_FT_SEV1)	Annual average temperature (AAT)
	Moderate severity left wheel path faulting (LT_FT_SEV2)	Annual average wind speed (AAWS)
	High severity left wheel path faulting (LT_FT_SEV3)	Annual average sun shine (AAS)
	Very high severity left wheel path faulting (LT_FT_SEV4)	Annual average precipitation (AAP)
	Low severity right wheel path faulting (RT_FT_SEV1)	Annual average relative humidity (AARH)
	Moderate severity right wheel path faulting (RT_FT_SEV2)	International roughness index (IRI)*
	<i>Rigid airport pavements</i> (2,000)	slab modulus (PS.MOD)
PCC slab thickness (PS.THICK)		Loading position in the x directin (X.OFFSET)
PCC slab Poisson ratio (PS.POISS)		Loading position in the y directin (Y.OFFSET)
Subbase 1 modulus (SB1.MOD)		Loading angle (ANGLE)
Subbase 2 thickness (SB1.THICK)		Joint stiffness (JOINT.SX)
Subbase 3 Poisson ratio (SB1.POISS)		Temperature gradient (TEMP.GRAD)**
Subbase 1 modulus (SB2.MOD)		Thermal coefficient (THERM.COEF)**
Subbase 2 thickness (SB2.THICK)		Maximum tensile stress on the surface in the x direction (σ_{xx_top})*
Subbase 3 Poisson ratio (SB2.POISS)		Maximum tensile stress on the surface in the y direction (σ_{yy_top})*
Subgrade modulus (SG.MOD)		Maximum tensile stress on the bottom in the x direction (σ_{xx_bot})*
Subgrade Poisson ratio (SG.POISS)	Maximum tensile stress on the bottom in the y direction (σ_{yy_bot})*	
Slab width (X)		

*: prediction target variable

** : variable only for the *case TM*

tions, and rigid pavement responses (i.e., maximum tensile stresses on top and bottom of portland cement concrete (PCC) slab). About 2,000 simulation data are obtained using mechanical or simultaneous thermal and mechanical loadings. The mechanical loading is set to be imposed by Boeing B777 aircraft. *Case M* and *case TM* refer to the simulation data obtained by using only mechanical loading and simultaneous thermal and mechanical loadings, respectively. 17 (19) variables for the *case M* (*case TM*) are used as predictor variables and maximum tensile stresses on the top and bottom of PCC slabs are predicted using GAM. The detailed descriptions of variables of both *concrete overlay* and *rigid airport pavements* databases are summarized in Table 4.1.

4.4 Best Predictor Variables for GAM Prediction

In this section, we investigate the best predictor variables for the *case 1* through *4* of *concrete overlay* and the *case M* and *case TM* of *rigid airport pavements* data. To find the best predictor variables for the *concrete overlay* and *rigid airport pavements* data, all possible variable combinations are investigated using TPRS and CRS by implementing a parallel program (see details in (Song et al., 2018b)). Hereafter, this approach is referred to *direct search*.

Figure 4.1 shows the number of best predictor variables of four cases of *concrete overlay* data with TPRS and CRS bases. It turned out that the use of all predictor variables does not always result in the highest prediction accuracy. In Figures 4.1a and 4.1b, R^2 value drops sharply when using all predictor variables. The numbers of predictor variables for the most accurate GAM prediction are 15, 11, 8, and 5 for *case 1* through *4*. TPRS led to a better prediction performance than CRS for all cases.

It seems that the distress information plays a significant role in IRI prediction. *Case 1* and *2* includes distress variables unlike *case 3* and *4*, and R^2 values from GAM predictions using *case 1* and *2* (i.e., 0.654 and 0.650) are fairly higher than those using *case 3* and *4* (i.e., 0.526 and 0.497). On the other hand, climatic related variables attribute to a better prediction when distress variables are not included in the predictors. In Figure 4.1, when changing from *case 4* (without climatic and distress variables) to *case 3* (without distress variables), R^2 changes from 0.497 to 0.526 while R^2 slightly increases from 0.650 to 0.654 when changing from *case 2* (without climatic variables) to *case 1* (all variables).

Figure 4.2 shows the number of best predictor variables of four cases of *rigid airport pavements* data with TPRS and CRS bases. These results also show the use of all variables does not lead to the highest accuracy of GAM prediction, but compared to the *concrete overlay* cases, the drop of R^2 when using all variables is not sharp. The numbers of predictor variables for the most accurate GAM prediction are 11, 14, 13, and 13 for 4 response variables of *case M* and 6, 12, 7, and 11 for those of *case TM*. When including two variables regarding thermal loading in predictors, GAM requires fewer predictor variables to produce the best prediction performance.

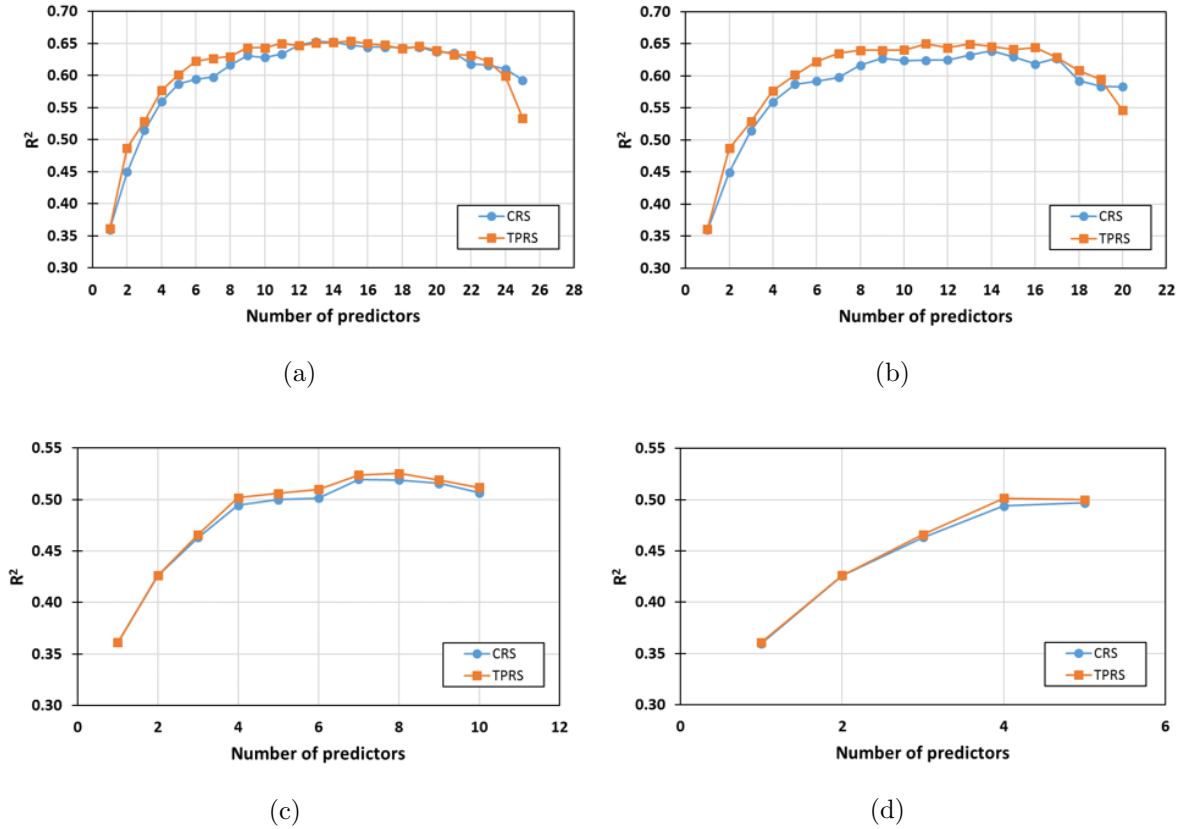
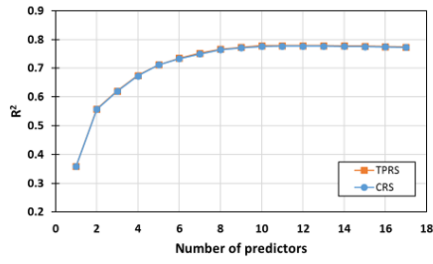


Figure 4.1: The number of predictor variables selected by direct search for the most accurate prediction of (a) *case 1*, (b) *case 2*, (c) *case 3*, and (d) *case 4* of *concrete overlay* data

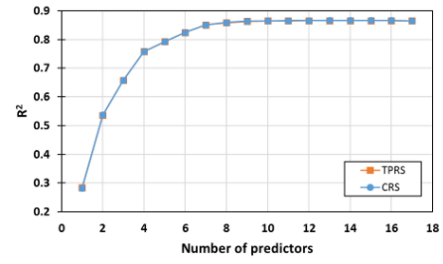
Interestingly, both TPRS and CRS led to almost the same prediction performance for all cases. This might be because the *rigid airport pavements* is simulation data generated by the FEA program, so there might be no high non-linearity between predictor variables, resulting in no difference of prediction performance between TPRS and CRS.

4.5 Relative Importance of Predictor Variables in GAM Prediction

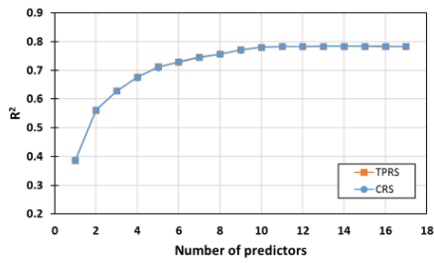
One of the attractive advantages of statistical models than machine learning is interpretability. Like other statistical models, GAM also provides p-value of each variable in the fitted model to present the relative importance of predictor variables in prediction. Table 4.2 shows the best predictor variables selected by the *direct search* and their p-values for each data. Variables are



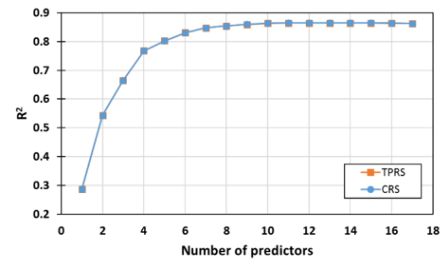
(a)



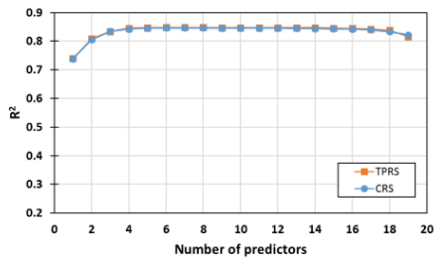
(b)



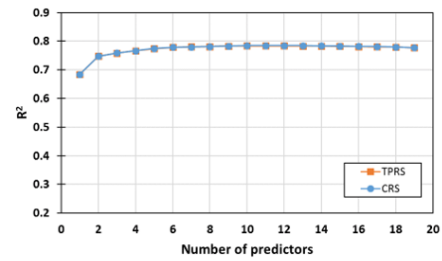
(c)



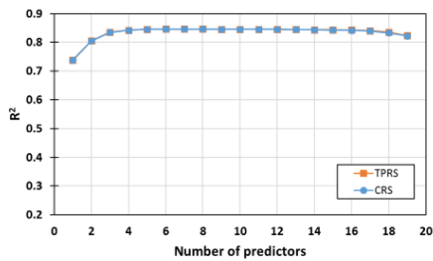
(d)



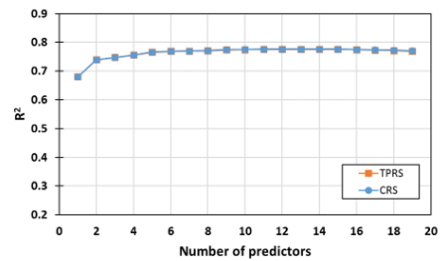
(e)



(f)



(g)



(h)

Figure 4.2: The number of predictor variables selected by direct search for the most accurate prediction of (a) σ_{xx_top} , (b) σ_{xx_bot} , (c) σ_{yy_top} , and (d) σ_{yy_bot} for *case M*; (e) σ_{xx_top} , (f) σ_{xx_bot} , (g) σ_{yy_top} , and (h) σ_{yy_bot} for *case TM* of *rigid airport pavements* data

listed in ascending order by p-value and the variables with smaller p-values represent relatively important variables in GAM prediction.

For the concrete overlay data, AGE, THICK, JT_SP, and TYPE seem to be the most important variables for all cases. TRAFFIC is only included in *case 3* and *4*, which means it might not be relatively important in GAM prediction when distress variables are included in predictors.

For the *case M* of *rigid airport pavements* data, thickness and modulus variables of the slab seem to play an important role in GAM prediction. It turned out that TEMP_GRAD is the most important predictor variable for all target responses of *case TM*. THERM_COEF plays a significant role in GAM prediction of the maximum tensile stresses on the top of the slab than the bottom of the slab.

4.6 Efficient Variable Selection Strategy

For accurate prediction, including appropriate variables into a predictive model is important. The use of all explanatory variables does not always guarantee the best prediction performance (Song et al., 2018c,b). The *direct search* examines all possible variable combinations, but it requires a high computing demand depending on the size of data and number of variables.

We investigate several variable selection methods to find the best predictor variables in an efficient manner. We compare the forward and backward methods (Efroymson, 1960) with Akaike information criterion (AIC) and p-value being the criteria. AIC is an estimator to measure the quality of statistical models and a smaller AIC value means a better model. A coefficient of determination, R^2 is used as the criterion to measure prediction accuracies.

4.6.1 Forward variable selection procedure descriptions

Forward variable selection method begins to build a regression model with no variables. One variable is added one by one until a criterion is satisfied. The selection steps based on p-value are as follows: Given the universal set of predictor variables, $\mathbf{U} = \{x_1, x_2, \dots, x_n\}$, where n is the number of total predictor variables,

Table 4.2: Selection of the best combination of variables for GAM using CRS (p-values in parentheses)

Data	Target response	Predictor variable (p-value)			R^2
<i>concrete overlay (case1)</i>	IRI	AGE (< 2e-16)	THICK (<2e-16)	JT.SP (4.58e-12)	0.654
		TYPE (5.64e-06)	AARH (1.04e-04)	RT_FT_SEV2 (1.80e-04)	
		AAP (0.00839)	TRANS_M (0.00899)	JSPALL_H (0.00913)	
		LT_FT_SEV4 (0.0722)	JSPALL_M (0.190)	TRANS_H (0.206)	
<i>concrete overlay (case2)</i>	IRI	RT_FT_SEV3 (0.566)	LT_FT_SEV1 (0.649)	LT_FT_SEV4 (0.764)	0.650
		AGE (<2e-16)	THICK (<2e-16)	JT.SP (1.83e-12)	
		TYPE (1.17e-05)	RT_FT_SEV2 (6.17e-06)	JSPALL_H (0.00106)	
		TRANS_M (0.00389)	LT_FT_SEV4 (0.142)	TRANS_H (0.287)	
<i>concrete overlay (case3)</i>	IRI	RT_FT_SEV3 (0.294)	RT_FT_SEV4 (0.801)		0.526
		AGE (<2e-16)	THICK (2.13e-14)	JT.SP (3.64e-08)	
		TYPE (0.000130)	AAWS (0.000171)	AARH (0.00215)	
		AAP (0.0175)	TRAFFIC (0.118)		
<i>concrete overlay (case4)</i>	IRI	AGE (<2e-16)	THICK (3.78e-15)	JT.SP (1.51e-09)	0.497
		TYPE (5.09e-04)	TRAFFIC (0.0803)		
<i>Rigid airport pavements (case M)</i>	σ_{xx_top}	SB1_THICK (<2e-16)	SB2_THICK (<2e-16)	PS_THICK (<2e-16)	0.779
		X_OFFSET (<2e-16)	PS_MOD (<2e-16)	SB1_MOD (<2e-16)	
	σ_{xx_bot}	SB2_MOD (<2e-16)	Y_OFFSET (<2e-16)	SG_MOD (3.44e-10)	0.866
		Y(1.57e-08)	JOINT_SX (0.0103)		
		PS_THICK (<2e-16)	SG_MOD (<2e-16)	SB1_THICK (<2e-16)	
		PS_MOD (<2e-16)	SB1_MOD (<2e-16)	ANGLE (<2e-16)	
	σ_{yy_top}	X_OFFSET (<2e-16)	SB2_MOD (<2e-16)	X (5.55E-12)	0.784
		SB1_MOD (<2e-16)	X (<2e-16)	PS_POISS (0.00163)	
σ_{yy_bot}	SG_MOD (8.92e-12)	Y (2.79e-04)		0.864	
	JOINT_SX (4.84e-02)	SB2_POISS (0.899)			
<i>Rigid airport pavements (case TM)</i>	σ_{xx_top}	SB1_THICK (<2e-16)	PS_THICK (<2e-16)	SB2_THICK (2.56e-10)	0.848
		X_OFFSET (<2e-16)	SB2_MOD (<2e-16)	PS_MOD (<2e-16)	
	σ_{xx_bot}	SB1_MOD (<2e-16)	X (<2e-16)	Y_OFFSET (1.99e-14)	0.776
		SG_MOD (8.92e-12)	Y (3.51e-07)	SG_POISS (4.26e-07)	
		JOINT_SX (4.84e-02)			
		PS_THICK (<2e-16)	SG_MOD (<2e-16)	SB1_THICK (<2e-16)	
	σ_{yy_top}	PS_MOD (<2e-16)	SB1_MOD (<2e-16)	ANGLE (<2e-16)	0.846
		SB2_MOD (<2e-16)	Y_OFFSET (<2e-16)	X (3.82E-13)	
σ_{yy_bot}	SB2_THICK (3.91e-10)	Y (1.64e-03)	PS_POISS (0.0045)	0.784	
	SG_POISS (0.0359)				
	TEMP_GRAD (<2e-16)	PS_MOD (<2e-16)	THERM_COEF (<2e-16)		
	X_OFFSET (0.00268)	Y_OFFSET (0.0378)	SB2_THICK (0.0442)		
σ_{yy_top}	TEMP_GRAD (<2e-16)	PS_THICK (<2e-16)	PS_MOD (<2e-16)	0.846	
	X_OFFSET (<2e-16)	SG_MOD (<2e-16)	SB1_THICK (<2e-16)		
σ_{yy_bot}	ANGLE (<2e-16)	SB1_MOD (1.97e-10)	X (3.23e-09)	0.784	
	THERM_COEF (0.0105)	SG_POISS (0.113)	JOINT_SX (0.630)		
	TEMP_GRAD (<2e-16)	PS_MOD (<2e-16)	THERM_COEF (<2e-16)		
	Y (0.0439)	SB2_THICK (0.0675)	Y_OFFSET (0.0735)		
σ_{yy_bot}	SB1_POISS (0.207)			0.784	
	TEMP_GRAD (<2e-16)	PS_THICK (<2e-16)	PS_MOD (<2e-16)		
σ_{yy_bot}	SG_MOD (<2e-16)	Y_OFFSET (<2e-16)	ANGLE (<2e-16)	0.784	
	SB1_THICK (<2e-16)	SB1_MOD (5.76e-11)	THERM_COEF (0.0236)		
σ_{yy_bot}	Y (0.0329)	X_OFFSET (0.311)		0.784	

1. Start with no variables in the regression model ($\mathbf{X}_{best}^{(t=0)} = \emptyset$). Define a candidate set of variables as

$$\mathbf{X}_{cand}^{(t)} \equiv \{x \mid x \in \mathbf{U}, x \notin \mathbf{X}_{best}^{(t)}\}$$

2. For each variable in $\mathbf{X}_{cand}^{(t)}$, check their p-value, $p(x)$ when the variable enters the regression model. The set of these p-values is defined as

$$\mathbf{P}_{cand}^{(t)} \equiv \{p(x) \mid x \in \mathbf{X}_{cand}^{(t)}\}$$

3. Find and keep the variable which results in the lowest p-value in the current model from the step 2,

$$\mathbf{X}_{best}^{(t+1)} = \mathbf{X}_{best}^{(t)} \cup \{x_{best}\}, \text{ where } x_{best} \equiv \underset{x \in \mathbf{X}_{cand}^{(t)}}{\operatorname{argmin}} (p(x))$$

4. Continues above steps until $\min(\mathbf{P}_{cand}^{(t)}) < \alpha_{crit}$, where α_{crit} is the criterion p-value (e.g., 0.05).

Similarly, the forward selection based on AIC is as follows: Given the universal set of predictor variables, $\mathbf{U} = \{x_1, x_2, \dots, x_n\}$,

1. Start with no variables in the regression model ($\mathbf{X}_{best}^{(t=0)} = \emptyset$). Define a candidate set of variables as

$$\mathbf{X}_{cand}^{(t)} \equiv \{x \mid x \in \mathbf{U}, x \notin \mathbf{X}_{best}^{(t)}\}$$

2. For each variable in $\mathbf{X}_{cand}^{(t)}$, check the AIC value of the model, $AIC(x)$ when the variable is included in the regression model. The set of these AIC values is defined as

$$\mathbf{AIC}_{cand}^{(t)} \equiv \{AIC(x) \mid x \in \mathbf{X}_{cand}^{(t)}\}$$

3. Find and keep the variable which results in the smallest AIC in the current model from the step 2,

$$\mathbf{X}_{best}^{(t+1)} = \mathbf{X}_{best}^{(t)} \cup \{x_{best}\}, \text{ where } x_{best} \equiv \underset{x \in \mathbf{X}_{cand}^{(t)}}{\operatorname{argmin}} (AIC(x))$$

4. Continues above steps until $\min(\mathbf{AIC}_{cand}^{(t+1)}) < \min(\mathbf{AIC}_{cand}^{(t)})$.

4.6.2 Backward variable selection procedure descriptions

Backward variable selection method begins to build a regression model with all variables. One variable is excluded from the regression model one by one until a criterion is satisfied. The selection steps based on p-value are as follows: Given the universal set of predictor variables, $\mathbf{U} = \{x_1, x_2, \dots, x_n\}$,

1. Start with all variables in the regression model ($\mathbf{X}_{best}^{(t=0)} = \mathbf{U}$).
2. From $\mathbf{X}_{best}^{(t)}$, exclude a variable whose p-value is the largest in the regression model,

$$\mathbf{X}_{best}^{(t+1)} = \mathbf{X}_{best}^{(t)} - \{x_{worst}\}, \text{ where } x_{worst} \equiv \underset{x \in \mathbf{X}_{best}^{(t)}}{\operatorname{argmax}} (p(x))$$

3. Fit the model again with $\mathbf{X}_{best}^{(t+1)}$ and goto the step 2.
4. Continues above steps until $\max_{x \in \mathbf{X}_{best}^{(t+1)}} < \alpha_{crit}$.

The backward selection based on AIC is similar to the case of p-value, which is as follows: Given the universal set of predictor variables, $\mathbf{U} = \{x_1, x_2, \dots, x_n\}$,

1. Start with all variables in the regression model ($\mathbf{X}_{best}^{(t=0)} = \mathbf{U}$). Define a candidate set of variables as

$$\mathbf{X}_{cand}^{(t)} \equiv \{x \mid x \in \mathbf{X}_{best}^{(t)}\}$$

2. For all variables in $\mathbf{X}_{cand}^{(t)}$, check the AIC value, $AIC(x)$ when each variable is removed from the regression model. The set of these AIC values is defined as

$$\mathbf{AIC}_{cand}^{(t)} \equiv \{AIC(x) \mid x \in \mathbf{X}_{cand}^{(t)}\}$$

3. From the model, exclude a variable which results in the smallest AIC value when the variable is removed from the regression model,

$$\mathbf{X}_{best}^{(t+1)} = \mathbf{X}_{best}^{(t)} - \{x_{worst}\}, \text{ where } x_{worst} \equiv \underset{x \in \mathbf{X}_{cand}^{(t)}}{\operatorname{argmin}} (AIC(x))$$

4. Fit the model again with $\mathbf{X}_{best}^{(t+1)}$ and goto the step 2.
5. Continues above steps until $\min(\mathbf{AIC}_{cand}^{(t+1)}) < \min(\mathbf{AIC}_{cand}^{(t)})$.

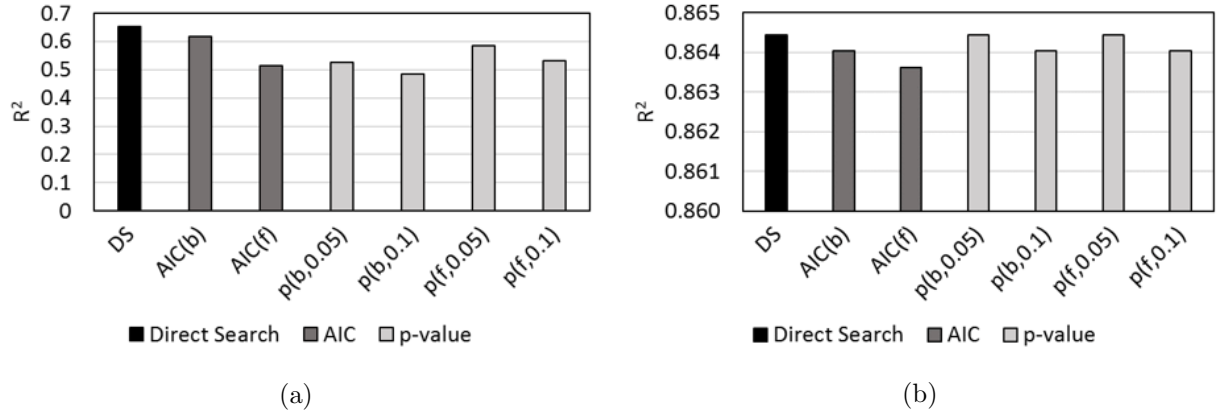


Figure 4.3: Prediction performances using different variable selection methods: (a) *concrete overlay*; (b) *rigid airport pavements*. DS stands for *direct search*, AIC(b) for backward selection using AIC, and p(f,0.05) for forward selection using p-value of 0.05, etc

4.6.3 Comparison of variable selection methods

Figure 4.3 shows the comparison of prediction performance between the *direct search*, forward and backward methods. As expected, the *direct search* always finds the best predictor variable combination and leads to the highest R^2 value.

For the *concrete overlay* data, the *direct search* led to the highest prediction accuracy and the backward selection using AIC results in the second highest. Among the variable selection methods using the p-value, the forward selection using the p-value of 0.05 produces the highest.

For the *rigid airport pavements* data, the *direct search* and the forward and backward selection methods using the p-value of 0.05 led to the highest p-value. In addition, the backward selection method using the AIC resulted in the next highest p-value. However, the difference of R^2 values between all methods is not significantly large. This might be because the *rigid airport pavements* data were obtained from the FEA simulation results which are produced by engineering-based principles; therefore, there is seldom uncertainty in the model, which makes almost the same prediction results between different variable selection methods.

In terms of computing time, when using the *direct search*, it took about 7 days and 5 days for the *concrete overlay* and *rigid airport pavements* data while the forward and backward selection

methods took less than 1 hour. Therefore, the *direct search* is recommended in the case that the highest prediction accuracy is necessary regardless of computing time while the backward selection using AIC is encouraged to use if an adequate prediction accuracy is acceptable and short computing time is allowed.

4.7 Impact of Distribution Family on Prediction Performance

In this section, the impact of distribution family of the response variable on the prediction performance. Multiple GAM models for the *rigid airport pavements* data are built using Gamma, Gaussian, and Poisson distributions and their relative prediction performances are compared. Figure 4.4 shows the best prediction results for each target response of the *case M* and *case TM*. It seems that Gamma distribution produces better prediction performances in most cases than Gaussian distribution. Meanwhile, Poisson distribution turns out to produce higher R^2 values than Gamma distribution. This might be because when using Poisson distribution, the response variable is considered as an integer which is rounded from actual values. This simplification may lead to better prediction results, but the predicted responses are values with different digits; therefore, Gamma distribution is recommended to use.

4.8 Parameter Study: Impact of Spline Base

In GAM, the number of spline base (k) should be specified. We investigate the impact of k on the GAM prediction of the concrete overlay data using TPRS. Figure 5 shows the results of the *case 1* through *4*. The k values to produce the highest R^2 are 11, 10, 73 and 28 for *case 1* through *4*. In general, k of 10 appears to be an appropriate choice with acceptable accuracy because, for the *case 1*, *3*, and *4*, the R^2 values from the prediction using 10 spline bases are very similar to the highest R^2 of each case. Wood (Wood, 2006) also recommends 10 as a suitable k value.

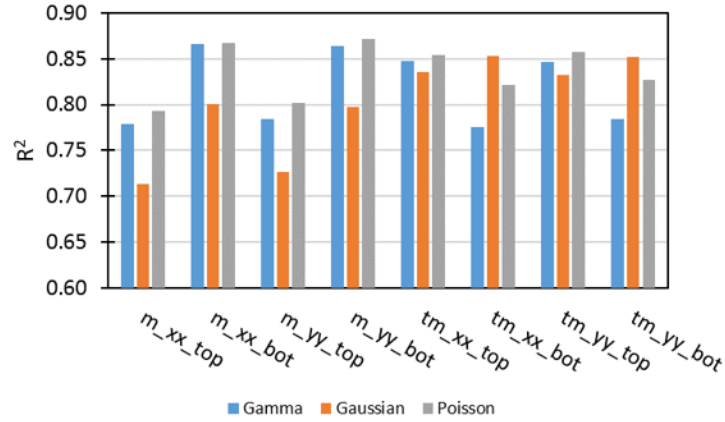


Figure 4.4: Prediction performance depending on different distribution families (Gamma, Gaussian, and Poisson), in which m_xx_top stands for the maximum tensile stress in the x direction on the top of the slab with the mechanical loading condition and tm_yy_bot stands for the maximum tensile stress in the y direction on the bottom of the slab with thermal and mechanical loading, etc

4.9 Conclusions

This study investigated the efficient variable selection methods and elucidate the relative importance of predictor variables for GAM prediction using field survey pavement data, *concrete overlay* and simulated airport pavement data, *rigid airport pavements*. In particular, the GAM is a flexible and non-parametric statistical method which enables to predict complex data with highly non-linearity.

Several variable selection methods, including the *direct search* and forward and backward selection based on p-value and AIC, are compared to investigate an efficient way to find the best predictor variables in GAM prediction. Results show that the *direct search* appears to always produce the best prediction results while the backward selection method using AIC produces acceptable prediction results; however, the computing time of backward selection was much smaller than that of the *direct search* method. Unless the highest prediction accuracy is the only goal regardless of the computing time, the backward selection using AIC would be a better choice as an efficient way to find the best predictor variables.

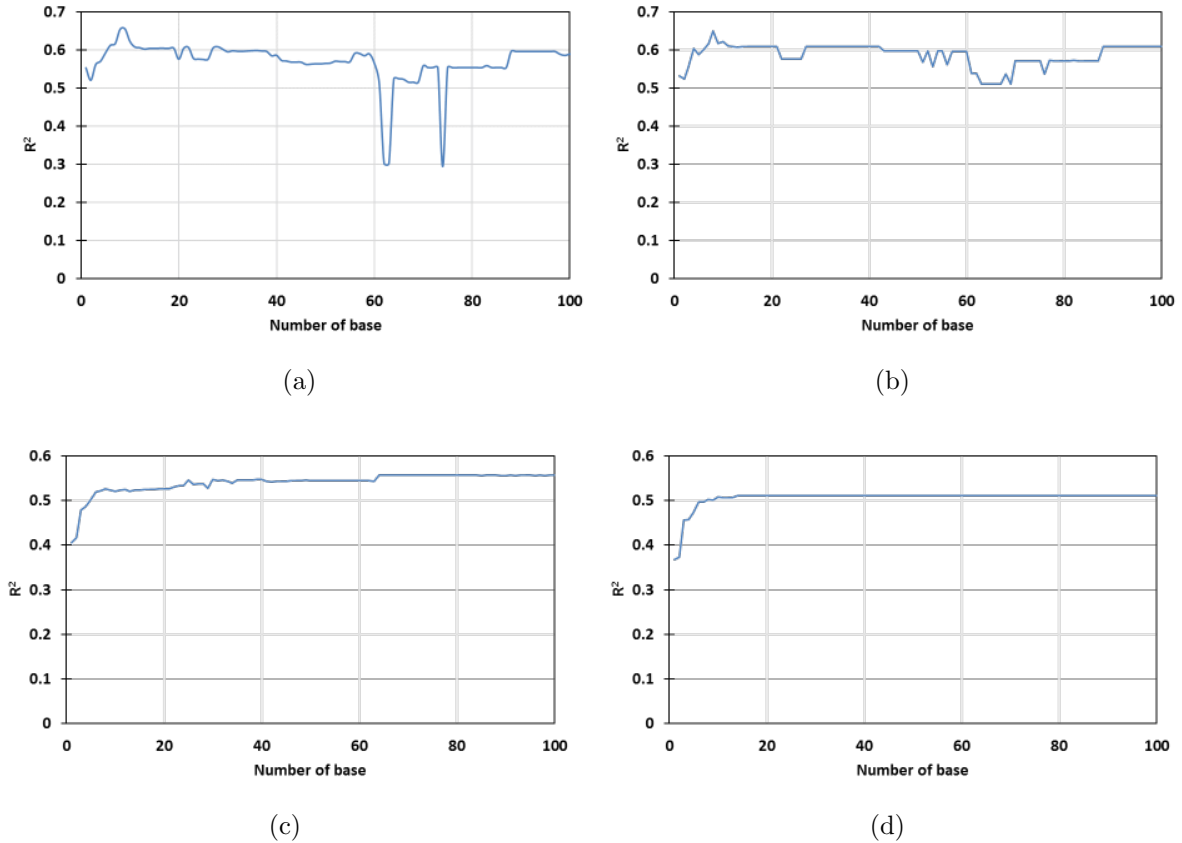


Figure 4.5: Impact of number of TPRS base (k) on GAM prediction: (a) *case 1*; (b) *case 2*; (c) *case 3*; (d) *case 4* of the *concrete overlay* data

The relative importance of variables in GAM prediction also has been investigated. For the *concrete overlay* database, age, thickness, joint spacing, and overlay type variables turn out to be important than others. Traffic variable appears to be only useful when distress variables are not included in GAM prediction. For the *rigid airport pavements* data, variables of thickness and modulus of pavement turn out to play a significant role in GAM prediction for the mechanical loading case. Temperature gradient appears to be the most important predictor variable for the simultaneous thermal and mechanical loading case.

The optimal GAM setting for accurate prediction has been addressed. Several distribution families are compared in terms of prediction accuracy depending on the distribution selection.

Results show the Gamma distribution is suitable for pavement data. The recommended number of spline base is also investigated and 10 is recommended as the suitable number of spline base.

The advanced statistical learning and prediction using GAM can suggest the best predictor variables with efficient variable selection methods as well as the relative importance of predictor variables in the predictive model. This feature will help the pavement engineering community understand the relationship between predictor and response variables, and help stakeholders establish a pavement management plan based on the knowledge obtained from GAM prediction results.

4.10 Acknowledgments

The authors gratefully acknowledge the Federal Aviation Administration (FAA), the Iowa Highway Research Board (IHRB), and the Iowa Department of Transportation (IA DOT) for supporting this study. The contents of this paper reflect the views of the authors who are responsible for the facts and accuracy of the data presented within. The contents do not necessarily reflect the official views and policies of the FAA, IA DOT, and Iowa State University. This paper does not constitute a standard, specification, or regulation.

Bibliography

- Ceylan, H., Tutumluer, E., and Barenberg, E. (1999). Artificial neural networks for analyzing concrete airfield pavements serving the boeing b-777 aircraft. *Transportation Research Record: Journal of the Transportation Research Board*, (1684):110–117.
- Ceylan, H., Tutumluer, E., and Barenberg, E. J. (1998). Artificial neural networks as design tools in concrete airfield pavement design. In *Airport Facilities: Innovations for the Next Century. Proceedings of the 25th International Air Transportation Conference. American Society of Civil Engineers*.
- Cho, I. H., Chen, A., Alipour, A., Shafei, B., Laffamme, S., Song, I., Yan, J., et al. (2018). Development of a computational framework for big data-driven prediction of long-term bridge performance and traffic flow.
- Efroymson, M. A. (1960). *Multiple regression analysis*, chapter 17. John Wiley & Sons, Inc., New York.
- FHWA (2017). Public road mileage, lane-miles, and vmt 1900 - 2016.
- Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223.
- Gopalakrishnan, K. and Kim, S. (2011). Support vector machines approach to hma stiffness prediction. *Journal of Engineering Mechanics*, 137(2).
- Gross, J., King, D., Harrington, D., Ceylan, H., Chen, Y., Kim, S., Taylor, P., and Kaya, O. (2017). Concrete overlay performance on iowa’ roadway.
- Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC Press, Boca Raton, FL, USA.
- Heba, A. and Assaf, G. J. (2017). Road performance prediction model for the libyan road network depending on experts knowledge and current road condition using bayes linear regression. In *International Congress and Exhibition” Sustainable Civil Infrastructures: Innovative Infrastructure Geotechnology”*, pages 153–167. Springer.
- Kaya, O., Rezaei-Tarahomi, A., Ceylan, H., Gopalakrishnan, K., Kim, S., and Brill, D. R. (2018). Neural networkbased multiple-slab response models for top-down cracking mode in airfield pavement design. *Journal of Transportation Engineering, Part B: Pavements*, 144(2):04018009.
- Rezaei-Tarahomi, A., Kaya, O., Ceylan, H., Kim, S., and Brill, D. R. (2018). Evaluation of artificial neural network architectures and training processes for developing rigid airport pavement multiple-slab surrogate response models.

- Salama, H. K., Chatti, K., and Lyles, R. W. (2006). Effect of heavy multiple axle trucks on flexible pavement damage using in-service pavement performance data. *Journal of transportation engineering*, 132(10):763–770.
- Song, I., Cho, I., and Tessitore, T. (2017). Advanced statistical learning and prediction of complex runway incursion. In *Airfield and Highway Pavements 2017*, pages 38–50.
- Song, I., Cho, I., Tessitore, T., Gurcsik, T., and Ceylan, H. (2018a). Data-driven prediction of runway incursions with uncertainty quantification. *Journal of Computing in Civil Engineering*, 32(2):04018004.
- Song, I., Cho, I. H., and Wong, R. K. W. (2018b). An advanced statistical approach to data-driven earthquake engineering. *Journal of Earthquake Engineering*, 0(0):1–25.
- Tabatabaee, N., Ziyadi, M., and Shafahi, Y. (2013). Two-stage support vector classifier and recurrent neural network predictor for pavement performance modeling. *Journal of Infrastructure Systems*, 19(3):266–274.
- Wood, S. (2006). *Generalized additive models: an introduction with R*. CRC Press.
- Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1):3–36.

CHAPTER 5. IMPACTS OF FRACTIONAL HOT-DECK IMPUTATION ON LEARNING AND PREDICTION OF ENGINEERING DATA

A paper submitted and under review for publication in *IEEE Transactions on Knowledge and Data Engineering*, (2018)

Ikkyun Song, Tong Tong, Jongho Im, Halil Ceylan, and In-Ho Cho

Abstract

In the broad engineering fields, missing data is a common issue which often causes undesired bias and sparseness impeding rigorous data analyses. To tackle this problem, many imputation theories have been proposed and widely used. However, prior methods often require distributional assumptions and/or prior knowledge regarding data which may cause some difficulty to routine engineering research. Essentially, the fractional hot-deck imputation (FHDI) is an assumption-free imputation method, holding a broad applicability in the engineering domains. FHDI's internal parameters and impact on statistical and machine learning methods, however, have been rarely understood. Thus, this study investigates the behavior and impacts of FHDI on prediction methods including generalized additive model, support vector machine, extremely randomized trees, and artificial neural network, for which four practical datasets (appliance energy, air quality, phenotypes, and weather) are used. Results show that FHDI performs better for improving the prediction accuracy compared to a simple naive method which cures missing data using the mean value of attributes, and FHDI has a gradually positive effect on prediction accuracy with decreasing response rates. Regarding an optimal setting, 30 to 35 is recommended for the FHDI's internal categorization number while 5 is recommended for the FHDI's randomly selected donors numbers, which is interestingly aligned with Rubin's recommendation to other comparable methods.

5.1 Introduction

Data missing is widely observed in surveys and experiments. It prevents researchers from obtaining a reliable conclusion from data analysis due to biasness and sparseness. For example, Brown and Kros (2003) and Roth (1994) highlighted how missing data causes wrong data analysis. To overcome this problem, a suitable remedy is required in research involving data mining and analysis.

Imputation is one of the most popular methods for handling incomplete data. It fills in missing data with plausible values to create complete data. Depending on the imputation size, the imputation can be classified into single imputations and repeated imputations. A single value is imputed to each missing value in single imputations, whereas each missing value is replaced with several values in repeated imputations. Although single imputation is a convenient method, it has a weakness that uncertainty due to missing data may not be reflected in imputation processes. This issue can be handled by repeated imputation methods.

There are two repeated imputation methods: multiple imputation and fractional hot-deck imputation. Multiple imputation (MI) (Rubin, 1976) fills in missing data using multiple plausible values, leading to better consideration of the uncertainty of missing data. Fractional imputation (Kalton and Kish, 1984) is another notable imputation method and it can further reduce imputation variances and provides consistent variance estimations compared to MI. Fractional hot-deck imputation (FHDI) is a fractional imputation taking two advantages of the hot-deck imputation: First, imputed values are built upon observed responses, not artificial values, thereby preserving the distribution features of the original data; Second, a strong model assumption is not necessary for imputation (Yang and Kim, 2016). This study uses the FHDI (Kim and Fuller, 2004) as an imputation method.

There have been numerous studies addressing the impact of imputation on machine learning (ML) approach. Farhangfar et al. (2008) studied the impact of imputation on the classification error reduction. Six imputation methods and six classifiers were used to measure how an imputation of missing data decreases classification errors. They concluded that there does not exist a universal

imputation method which can ensure the largest error reduction for any classification, and so the choice of imputation method depends on classification types. Batista and Monard (2003) investigated the influence of four imputation methods (i.e., k-nearest neighbor, mean of mode imputation, internal methods in C4.5 and CN2) on ML performance. They concluded that the k-nearest neighbor outperforms other imputation methods. Heltshe et al. (2012) examined multiple imputation methods to improve the prediction power of the pesticide use. Lin et al. (2017) adopted the MI to impute missing data for sensitivity analysis and obtained regression coefficients similar to those from a complete dataset without missing values. Wang et al. (2016) also used the nearest neighbor scheme for sensitivity analysis to investigate the effect of missing data on travel time predictions. They found that prediction performances become worse when missing rates increase. Su et al. (2008) utilized multiple ML classifiers to resolve the data sparsity of a dataset for collaborative filtering. They found that the collaborative filtering using an imputation outperforms the traditional collaborative filtering. Yoo et al. (2017) used the MI to cure missing data for a better prediction of graft survivals after kidney transplants. They found that MI improves prediction accuracies. However, the impacts of FHDI on statistical and ML regression have been rarely investigated, which is strongly needed in view of promising applicability of FHDI.

This study aims to (i) introduce a relatively new FHDI to a wide array of engineering community, (ii) elucidate the impact of FHDI on statistical learning (SL) and ML prediction performance, and (iii) identify optimal settings and conditions of FHDI by performing a comprehensive sensitivity analysis covering different response rates, initial categorization numbers, and donor numbers with multiple datasets that have a large number of instances and attributes.

The outline of the paper is as follows: we briefly summarize the theory and default settings of FHDI, an advanced SL method (i.e., generalized additive model (GAM)), and three ML methods (i.e., support vector machine (SVM), extremely randomized trees (ERT), and artificial neural network (ANN)). After a brief explanation of four practical engineering datasets used in this study, key imputation procedures of FHDI are presented. Finally, several aspects of the impact of FHDI

on SL and ML prediction performance are addressed, and the results of detailed sensitivity analyses and recommendation for FHDI are presented.

5.2 Theory: Fractional Hot-Deck Imputation

This section summarizes the central notion of the FHDI, and one is referred to Im et al. (2015) for details. Suppose that we have p -dimensional variable $\mathbf{y} = \{y_1, y_2, \dots, y_p\}$ from a finite population U . Let A be the index set of possible samples from U and δ_{pi} be a response indicator function for y_{pi} , where $i \in A$. The function δ_{pi} takes a value 1 when y_{pi} is observed and zero otherwise. Let A_R and A_M be the subsets of respondents and nonrespondents respectively, where $A_R = \{i \in A; \delta_{1i}\delta_{2i} \cdots \delta_{pi} = 1\}$ and $A_M = \{i \in A; \delta_{1i}\delta_{2i} \cdots \delta_{pi} = 0\}$. Denote n_R and n_M be the size of A_R and A_M . Let $\mathbf{y}_{i,obs}$ and $\mathbf{y}_{i,mis}$ be respectively the observed and missing parts of \mathbf{y}_i . Let \mathbf{z} be the discretized values of \mathbf{y} , and $\mathbf{z}_{i,obs}$ and $\mathbf{z}_{i,mis}$ be categorical variables corresponding to $\mathbf{y}_{i,obs}$ and $\mathbf{y}_{i,mis}$ respectively. For example, assume that there is a sample, $\mathbf{y}_i = \{7, 2, NA, 5, NA\}$, then $\mathbf{y}_{i,obs} = \{7, 2, 5\}$ and $\mathbf{y}_{i,mis} = \{NA, NA\}$, where NA denotes a missing value.

Note that \mathbf{z} plays imputation cells in the implementation of the hot-deck imputation. Let $D_i = \{\mathbf{z}_{i,mis}^{*(1)}, \dots, \mathbf{z}_{i,mis}^{*(M)}\}$ be the set of all possible $\mathbf{z}_{i,mis}$ values, where M is the number of donors on a recipient i . Here, recipients are the subset of nonrespondents who have at least one missing item and donors are the subset of respondents whose observed values are used to fill in missing values of the recipients.

Using a finite mixture model, under the missing at random condition, the conditional distribution of $f(\mathbf{y}_{i,mis} | \mathbf{y}_{i,obs})$ is approximated by

$$f(\mathbf{y}_{i,mis} | \mathbf{y}_{i,obs}) \cong \sum_{s=1}^{M_i} p(\mathbf{z}_{i,mis}^{*(s)} | \mathbf{z}_{i,obs}) f(\mathbf{y}_{i,mis} | \mathbf{z}_{i,obs}, \mathbf{z}_{i,mis}^{*(s)}), \quad (5.1)$$

where $p(\mathbf{z}_{i,mis}^{*(s)} | \mathbf{z}_{i,obs})$ is a conditional cell probability. The conditional cell probability is generally unknown, and thus it should be estimated properly. We use the EM algorithm for that purpose (see Im et al. (2015) for details). When the estimated conditional cell probability is defined as

$$\hat{\pi}_{s|g} = \hat{p}(\mathbf{z}_{g,obs}, \mathbf{z}_{g,mis}^{*(s)}) / \sum_{s=1}^{M_i} \hat{p}(\mathbf{z}_{g,obs}, \mathbf{z}_{g,mis}^{*(s)}), \quad (5.2)$$

where \hat{p} represents an estimated probability, then, the FEFI estimator of $Y_p = \sum_{i=1}^N y_{pi}$ is defined as

$$\hat{Y}_{p,FEFI} = \sum_{i \in A} \omega_i \left\{ \delta_{pi} y_{pi} + \sum_{g=1}^G (1 - \delta_{pi}) a_{ig} \sum_{s=1}^{M_i} \hat{\pi}_{s|g} \hat{\mu}_{gs} \right\}, \quad (5.3)$$

where $a_{ig} = \sum_{s=1}^{M_i} a_{igs}$, $a_{igs} = 1$ when $(\mathbf{z}_{i,obs}, \mathbf{z}_{i,mis}) = (\mathbf{z}_{i,obs}, \mathbf{z}_{i,mis}^{*(s)})$ and 0 otherwise. Assume that $\omega_{ij,FEFI}^* = \sum_{g=1}^G a_{ig} \sum_{s=1}^{M_i} \hat{\pi}_{s|g} \{\omega_j \delta_j a_{jgs} / \sum_{l \in A} \omega_l \delta_l a_{lgs}\}$, and

$$\hat{\mu}_{gs} = \frac{\sum_{j \in A} \omega_j \delta_j a_{jgs} y_{pj}}{\sum_{j \in A} \omega_j \delta_j a_{jgs}}. \quad (5.4)$$

Then, Equation 5.3 can be changed to

$$\hat{Y}_{i,FEFI} = \sum_{i \in A} \omega_i \left\{ \delta_{pi} y_{pi} + (1 - \delta_{pi}) \sum_{j \in A} \omega_{ij,FEFI}^* y_{pj} \right\}. \quad (5.5)$$

M donors can be selected by using systematic probability proportional to size (PPS) sampling procedure, and then, the fractional hot-deck imputation (FHDI) estimator of Y_p is defined as

$$\begin{aligned} \hat{Y}_{p,FHDI} &= \sum_{i \in A} \omega_i \left\{ \delta_{pi} y_{pi} + \sum_{g=1}^G (1 - \delta_{pi}) a_{ig} \sum_{s=1}^{M_i} \hat{\pi}_{s|g} \bar{y}_{pi}^* \right\} \\ &= \sum_{i \in A} \omega_i \left\{ \delta_{pi} y_{pi} + (1 - \delta_{pi}) \sum_{j \in A} \omega_{ij}^* \bar{y}_{pi}^{*(j)} \right\}, \end{aligned} \quad (5.6)$$

where $\bar{y}_{pi}^{*(j)}$ is the j^{th} donor of y_{pi} , $\bar{y}_{pi}^* = M^{-1} \sum_{j=1}^M y_{pi}^{*(j)}$, and $\omega_{ij}^* = M_i^{-1}$. Equation 5.6 can be expressed using the FEFI estimator:

$$\begin{aligned} \hat{Y}_{p,FHDI} &= \hat{Y}_{p,FEFI} + (\hat{Y}_{p,FHDI} - \hat{Y}_{p,FEFI}) \\ &= \hat{Y}_{p,FEFI} + \sum_{i \in A} \omega_i (\bar{y}_{pi}^* - \hat{\mu}), \end{aligned} \quad (5.7)$$

where $\hat{\mu} = \sum_{i \in A} \omega_i y_i / \sum_{i \in A} \omega_i$.

5.3 Theory: Statistical Learning and Machine Learning Methods

5.3.1 Statistical learning: GAM

GAM (Hastie and Tibshirani, 1990) is a generalized linear modal. Compared to other statistical methods with predefined distribution and parameters, GAM has more flexibility and general

applicability because of undefined smooth functions (Wood, 2006). The superior prediction power of GAM has been recently investigated in engineering domains (Song et al., 2018a,b,c). A general form of GAM can be represented as:

$$g(\mu_i) = f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + \cdots + f_j(x_{ki}), \quad (5.8)$$

where g is a link function, $\mu_i \equiv E(Y_i)$, Y_i is a response variable from an exponential family of distribution, f_j is a smooth function of a single or multiple covariates. This non-specified smooth function gives GAM flexibility in complex datasets. For brevity of explanation, the following description involves a normally distributed single variable, but generalization to multiple variables is straightforward (Wood, 2006). A smoothing function can be represented as

$$f(x) = \sum_{j=1}^k b_j(x) \beta_j, \quad (5.9)$$

where $b_j(x)$ is the j^{th} basis function and β_j is an unknown parameter.

The model fitting can be achieved by maximizing the corresponding likelihood with a penalty term $\lambda \int [f''(x)]^2 dx$, where λ is a smoothing parameter. When λ value is too large, an over-smoothed estimate is made; oppositely, it leads to an under-smoothed estimate with a too small λ value. The error becomes the largest in the both extreme cases. The appropriate selection of λ can be achieved by minimizing generalized cross validation (GCV) score. This GCV score-based optimization of lambda is automatically done by the library of *mgcv*, a GAM package in *R* (Wood, 2011).

For constructing GAM, proper bases need to be selected. Cubic regression spline (CRS) (Wood, 2006) and thin plate regression spline (TPRS) (Wood, 2006) are two popular bases. In this study, TPRS is selected as a base function owing to its generality and flexibility for multivariate data sets. TPRS (Duchon, 1977) can be used for multiple covariates and be determined by minimizing $\|\mathbf{y} - \mathbf{f}\|^2 + \lambda J_{md}(\mathbf{f})$, where \mathbf{y} is the vector of y_i data and the set of smoothness functions $\mathbf{f} = [f(x_1), f(x_2), \cdots, f(x_n)]^T$, $J_{md}(\mathbf{f})$ is a penalty functional measuring the 'wiggleness' of \mathbf{f} . The trade-off between data fitting and smoothness of \mathbf{f} can be adjusted by $\lambda J_{md}(\mathbf{f})$. The wiggleness

penalty is defined as

$$J_{md} = \int \cdots \int_{R^d} \sum_{v_1 + \cdots + v_d} \frac{m!}{v_1! \cdots v_d!} \left(\frac{\partial^m f}{\partial x_1^{v_1} \cdots \partial x_d^{v_d}} \right)^2 dx_1 \cdots dx_d \quad (5.10)$$

In this study, GAM is adopted as the advanced statistical regression method to measure prediction accuracies after FHDI imputation. λ is automatically optimized according to GCV scores by the R library, and the number of bases is set as 10, which is the default setting.

5.3.2 Recap and settings of the adopted machine learning methods

ML is a popular field in computer science which mainly deals with learning and predicting the relationship between inputs and outputs of a given dataset. Amongst many popular ML methods, this study selected three methods that are widely used in a broad engineering domain.

5.3.2.1 Extremely randomized tree

Extremely randomized tree (ERT) (Geurts et al., 2006) is a tree-based ensemble method for supervised classification and regression problem, which selects splits, attributes and cut-points totally or partially at random. Compared to other tree-based methods, ERT splits nodes by choosing cut-points fully at random and using whole learning samples to grow the trees. Learning samples and test samples are used for building models and computing its accuracy, respectively. The algorithm are run a number of times (e.g. 10 times) on each dataset and mean square-errors are estimated for regression. The brief explanation on key processing steps for ERT is as follows:

For the total input size K ($1 \leq K \leq N$),

1. Input vector $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K)$ is randomly selected, where \mathbf{X} is the input data which is used to predict the target data.

2. For each selected input vector \mathbf{X}_i , calculate its minimum and maximum value to be its interval $[X_i^{min}, X_i^{max}]$. From the interval, a few cut-points, X_c are randomly selected, and then, splits are selected which are less than the cut-points.

3. Select the best splits by using the $Score(s, S)$. S is the subset of the input. For regression problems (Kamdar et al., 2016),

$$Score(s, S) = \frac{var(y | S) - \frac{|S_l|}{|S|}var(y | S_l) - \frac{|S_r|}{|S|}var(y | S_r)}{var(y | S)}, \quad (5.11)$$

where S_l and S_r are two subsets from sample S , $var(y|S)$ is the variance of the output y in the sample S . Absolute values of S_l or S_r are lower than n_{min} which denotes the minimum sample size for splitting a node.

4. Check the conditions of process listed above: (1) all description data are constant in the absolute value of S which is a subset of input; (2) the boolean output is constant in the absolute value of S ; (3) $|S|$ is lower than n_{min} which denotes the minimum sample size for splitting a node. If the result are all satisfied the conditions listed above, the model will stop splitting. Otherwise, the model needs to be re-built by repeating the entire process until it is satisfied.

ERT has advantages over other tree-based methods. As reported (Geurts et al., 2006), ERT may have more accuracy and have a larger number of leaves and deeper level of tree, which will speed up the training process.

In this study, the ERT is used as an advanced machine learning-based regression method to compare prediction performances after curing data by using FEFI and FHDI. The basic default settings of the adopted ERT program were used without tuning. The R package, **extraTrees** (Simm et al., 2014), is used for the implementation of ERT. The number of tree used in this study is five hundred; the node size is set as one; the number of random cuts for each input data is set as one; the number of input data tried at each node is set as $K/3$.

5.3.2.2 Artificial neural networks

Artificial neural networks (ANN) (McCulloch and Pitts, 1943) is a mathematical representation and may be regarded as a generalizations of existing statistical models. In 1943, a neuron network model in human brains began to be used in computer science area. Since then, it has been widely used in broad areas, including social network, speech recognition, and computer vision (Ghasemi, 2017).

ANN is built from perceptrons which are the most basic form of a neural network and do the same role of neuron in model. The perceptrons are combined in each layer, but the layer of perceptrons are independent to the other layers. In the input layer, each perceptron stands for one variable. The perceptron in the hidden layer is received from the input layer. One perceptron in output layer stands for each response variable and received input from the perceptrons in the final hidden layer. Between input and output layers, there are non-linear functions which can transfer and modify the data from input layer to the output layer. The weight, W_{ij} , is the connection between perceptrons, which controls the influence of input and adjusts the output error. i and j represent different layers of perceptrons. The weighted perceptron composes a linear function and a threshold is added to change the linear function to non-linear function for better training. To briefly explain the key notion of ANN, a well-known ANN algorithm (Jang et al., 1997) is summarized below:

For an input (X_1, X_2, \dots, X_K) and an output (Y_1, Y_2, \dots, Y_K) , The input in neuron j in preview layer, O_j , can be expressed as follows,

$$O_j = f \left(\sum_i W_{ij} O_i \right), \quad (5.12)$$

where O_i is an output from neuron i , f is the activation function which can be given by $f(x) = 1/(1+e^{-x})$. The mean square error (denoted as E) of output O_j is

$$E = \frac{1}{2} (T_j - O_j)^2, \quad (5.13)$$

where T_j is a target value. Based on the gradient descent method, the adapted weight ΔW_{ij} can be defined as

$$\Delta W_{ij} = -\frac{\partial E}{\partial O_j} \frac{\partial O_i}{\partial W_{ij}} = \delta_j g O_i, \quad (5.14)$$

where δ_j is the error signal which equals $-\partial E/\partial O_j$, g is an adaptation gain. If j is in the output layer,

$$\delta_j = (T_j - O_j)(1 - O_j)O_j, \quad (5.15)$$

If j is not in the output layer which may be in the hidden layer,

$$\delta_j = (1 - O_j) \sum_k \delta_k W_{jk}. \quad (5.16)$$

According to previous calculation Equations 5.15 and 5.16, the modified gradient descent update weight is

$$\Delta W(m) = -\delta_j g O_i + \alpha \Delta W(m-1), \quad (5.17)$$

where m is the number of iteration to calculate the weight.

For ANN-based predictions, this study adopted the *R* package, **neuralnet** (Fritsch and Guenther, 2016). A logistic activation function and a gradient descent algorithm were used for implementation. The hidden layer is set as one and the number of perceptrons is set at ten, which are default settings in **neuralnet**.

5.3.2.3 Support vector machines

Support vector machines (SVM) (Cortes and Vapnik, 1995) is a supervised learning method which can learn independent dimensional feature space. It texts categorization by high dimensional input spaces, uses few irrelevant features and sparse document vectors. SVM creates a linear separating hyperplane in a higher-dimensional space and constructs a maximum margin separator using a kernel trick. The linear separating hyperplane line is described as

$$g(\vec{x}) = \vec{w}^T \vec{x} + b, \quad (5.18)$$

where $\vec{w} = \sum_i \alpha_i y_i \vec{x}_i$ that is a weight vector for the linear combination of training points, α_i is a Lagrange multiplier, and x_i and y_i are descriptive data $T(x_i, y_i)$. The b controls the distance between training points and a hyperplane. The maximum margin can be determined as

$$\min_{w,b} \frac{1}{2} \|\vec{w}\|^2 \quad (5.19)$$

subject to $y_i w^T \leq 1$.

SVM can be used for regressions to deal with continuous variables. SVM tries to find a relative flat hyperplane for a better regression. A formula of SVM is:

$$\text{Minimize } \frac{1}{2} \sum_{i=1}^N w_i^2 + C \sum_{i=1}^n (\varepsilon_k + \varepsilon_k^*) \quad (5.20)$$

subject to

$$y_k - w^T \varphi(x_k) - b \leq \varepsilon + \varepsilon_k^* \quad (5.21)$$

$$w^T \varphi(x_k) + b - y_k \leq \varepsilon + \varepsilon_k^* \quad (5.22)$$

$$\varepsilon, \varepsilon_k, \varepsilon_k^* \geq 0, \quad (5.23)$$

where, ε_k and ε_k^* are slack variables; C is a positive constant that controls the trade-off between the penalty and margin; φ is kernel function.

SVM adds support vectors to maximize margins for minimizing prediction errors. Meanwhile, it is similar with a neural network in two aspects: first, SVM has a universal approximation property; second, SVM uses any of the rule extraction methods for making it more comprehensible and accurate. For SVM-based prediction, we used an *R* package, **e1071** (Meyer et al., 2017). After a preliminary study, we found that the radial basis relatively performed well for our data sets. Thus, throughout the predictions in this study, the radial bases is used as a kernel function. The parameter for all kernels is set to $1/(\text{data dimension})$, and the cost of constraints violation is set as one.

5.4 Materials

For investigating the impact of FHDI on regression, we choose four datasets that mainly have 9 to 14 attributes and varying number of instances (i.e., 6,000 - 42,000) and attributes' values are continuous. The summary of datasets is shown in Table 5.1. Datasets of appliance energy and air quality are obtained from UC Irvine machine learning repository (Bache and Lichman, 2013).

Table 5.1: Summary of datasets used in the current study

Name	Instances	Attributes	Description
Appliance energy	19,735	14	Appliance energy use in houses
Air quality	41,757	9	Air quality in Beijing
Phenotype	5,931	13	Effect of genotype on maize hybrid yield
Weather	36,220	13	Ozone in the United States

5.5 Imputation

To implement FHDI, we used **FHDI** package in *R* (Im et al., 2018). Because the four datasets don't have missing values, we intentionally created multiple incomplete datasets to be imputed. For each original dataset, five datasets that have different total response rates (i.e., 10% to 50% with 10% interval) are produced to investigate the impact of FHDI imputation on SL and ML prediction with varying response rates. Total response rate means the proportion of fully observed samples (i.e., samples without missing values). In the case of 10% total response, for example, 10% of samples don't have missing values and 90% of samples have missing values. The reason for using total response rates instead of missing rates is that the FHDI method requires fully observed samples for imputation. Hereafter, $N\%$ -dataset refers to the cured dataset using $N\%$ total response, where $N = \{10, 20, \dots, 50\}$.

Since missing of some attributes' value is inconceivable (e.g., date), missing values are not generated for those attributes. 14 attributes of the appliance energy dataset are assumed not to have missing values to make full samples under the missing at completely at random condition.

FHDI's key imputation procedure consists of 4 steps: (1) cell construction with categorization; (2) estimation of cell probabilities; (3) construction of fractional weights; (4) imputation. These steps are briefly explained in Table 5.2. First, in the cell construction step, variables are divided into several categories so that a recipient has enough donors for imputation. In the cell probability

estimation step, the joint cell probability for each attribute is estimated using a modified EM algorithm (Im et al., 2015). Next, fractional weights are calculated using the estimated joint cell probabilities. Finally, missing values cured by being filled with the observed values of donors.

FHDI uses partial donors while FEFI employs all available donors for imputation. In the later sections, particular categorization numbers and donor numbers are recommended for reducing prediction errors from parameter studies.

We can check whether FHDI imputations are conducted appropriately by comparing mean values of attributes between the original and cured datasets. The mean values should be similar each other when imputations are conducted correctly. It should be noteworthy to provide an example of FHDI implementation result to show the performance of FHDI. Using the appliance dataset, we generated three datasets that have a different missing rate (i.e., 10, 30, and 50%). The datasets were cured by using the FHDI with the categorization number of 35 and the donor number of 5. The results are shown in Table 5.3. The ratio of an attribute’s mean value in the original dataset to that in the cured dataset by FHDI is used to exhibit the performance of FHDI. The mean ratios are 1.001, 1.0037, and 1.0081 for 10, 30 and 50% missing rates, respectively. This result indicates that the FHDI works appropriately because the ratios are very close to 1.0. Meanwhile, the largest ratio is 1.1058 for the attribute ‘2’ and 50% missing rate.

5.6 Impact of FHDI on Statistical and Machine Learning-Based Regression

Our goal is to investigate the impact of FHDI on the prediction performance. To examine this impact, we use the normalized root mean square error (denoted as nRMSE), i.e., the ratio of the RMSE from a prediction using cured datasets to the RMSE using original datasets. This measures how much errors are increased from the prediction models using the cured datasets compared to the prediction models using original datasets. Hereafter, for brevity $nRMSE_{imputation,dataset,method}$ denotes a nRMSE from a prediction using a *method* after an application of *imputation* to a *dataset*, where $imputation = \{ori, FEFI, FHDI\}$ and *ori* stands for original which means an imputation is not applied; $dataset = \{app, air, phe, wea\}$ and *app*, *air*, *phe*, and *wea* stand for appliance

Table 5.2: Four key steps for FHDI method

Step	Description
Cell construction	Attribute values are transformed to a categorization number, k , to make a cell. The available range of k is 1 to 35 in the current version of <i>FHDI</i> package.
Cell probability estimation	Probability for each unique observed cell pattern is estimated by EM algorithms. Sum of all cell probabilities is 0.
Fractional weights construction	A fractional weight for each donor is determined to fill a missing part with imputed values.
Imputation	Missing values are imputed by donors. FHDI uses some selected donors while FEFI uses all possible donors.

energy, air quality, phenotype, and weather, respectively; $method = \{GAM, SVM, ERT, ANN\}$. For example, $nRMSE_{ori, phe, GAM}$ represents the nRMSE from a prediction using GAM with the original phenotype dataset and $nRMSE_{FEFI, wea, SVM}$ denotes the nRMSE from a prediction using SVM with the weather dataset cured by FEFI. Note that each subscript can be used separately (e.g., $nRMSE_{ori}$).

5.6.1 Positive role of FHDI on prediction accuracy improvement

To briefly touch upon the positive role of FHDI on prediction accuracy improvement, we cured various datasets with different missing rates by using FHDI and a naive methods. Then, the target response of each dataset is predicted using the four regression methods (i.e., GAM, SVM, ERT, and ANN). Here, the naive method means curing missing values using the mean value of the corresponding variable. For example, suppose that 10 of 100 instances of a variable are missing and the mean value of 90 instances is 2.0. Then, the 10 missing values are filled with the value of 2.0. For the FHDI, we used maximum values of k and M without prejudice (i.e., $k=35$ and $M=n$). Table 5.4 presents the comparison of prediction results after curing by FHDI and the naive method. Almost all prediction results using the FHDI method are better than those using the naive

Table 5.3: Expectation ratio (i.e., expectation $E[.]$ of each attribute in the original full data set divided by that of cured data set by FHDI) with different missing rates (10, 30 and 50%). The appliance energy dataset is used.

Attribute	$E[Y]/E[Y_{FHDI}]$			Attribute	$E[Y]/E[Y_{FHDI}]$		
	10%	30%	50%		10%	30%	50%
1	1.0023	0.9993	1.0116	8	1.0000	0.9998	0.9998
2	1.0105	1.0583	1.1058	9	0.9994	0.9964	0.9945
3	1.0000	1.0002	1.0002	10	1.0000	0.9999	1.0000
4	1.0000	1.0003	1.0001	11	1.0006	1.0022	1.0020
5	1.0000	0.9998	0.9997	12	0.9987	0.9959	0.9991
6	1.0002	1.0005	1.0010	13	1.0003	0.9986	1.0001
7	1.0000	0.9998	1.0000	14	1.0021	1.0015	1.0002

method in terms of prediction error. The difference of prediction error between the FHDI and naive method is especially remarkable in the result using the phenotype dataset. The nRMSE from the prediction using the naive method is 17 times larger than that using FHDI in the 50%-response rate case. These results show that the FHDI outperforms the simple naive method in terms of prediction accuracy.

Figures 5.1-5.8 show the influence of different response rates on prediction performances. The value of the vertical axis represents nRMSE. For example, 1.05 represents that $nRMSE_{FEFI}$ is 5% larger than $nRMSE_{ori}$. Most cases exhibit that the higher response rate, the lower RMSE. This shows the sensitivity of missing values to prediction performances. The maximum increment of nRMSE was 5 (Figure 5.2a).

In Figure 5.2a, $nRMSE_{FEFI, phe, GAM}$ appears to be larger than that for other datasets. In Table 5.4, the RMSE from the GAM prediction with phenotype is remarkably smaller than that with other datasets. To investigate the reason behind this salient trend, the coefficient of variance (CV) of RMSE (i.e., the ratio between RMSE and the mean of target response) is used. In Figure 5.9, CV of RMSE values are not changed significantly as the missing rate is changed from 10% to

50%. The $nRMSE_{FEFI, phe, GAM}$, however, reduces sharply as the missing rate increases while that for other datasets is not changed considerably. This implies that the CV of RMSE may influence the impact of a response rate on prediction performances. In particular, the CV of RMSE from phenotype prediction is notably smaller than that of others, resulting in a dramatic drop of the $nRMSE_{FEFI, phe, GAM}$. This shows that the impact of response rates may become significant for a prediction where the CV of RMSE is substantially small.

5.6.2 Impact of the categorization number

We investigate the impact of the initial categorization number (k) on the prediction performance using GAM, SVM, ERT, and ANN with 4 datasets. The maximum value of k is 35 in the current version of FHDI program (Im et al., 2018). Seven different k values (i.e., 5 to 35 with 5 interval) are used for imputation to examine the influence of k value on prediction performances. Overall, the nRMSE appears to decrease as k value increases. It is noticeable in the trends in the air quality and phenotype datasets, (see Figures 5.2 and 5.3). Also, from the regression by GAM of the phenotype dataset using 10% response rate, RMSE decreases from 7.2 to 6.1 as k increases from 5 to 35. Based on these parametric study results, 30 and 35 are recommended for the initial categorization number of FHDI. It should be noted that the current limit of 35 stems from the coarse-size categorization of continuous variables in the FHDI R package's *CellMake* function that uses 35 letters (0 to 9, and a to z) internally. This limit will be extended in the future upgrade of the FHDI R package.

It should be noteworthy to explain why large k values result in better imputation results. Let X be a random variable and $\{x\}_{i=1}^n$ be observed samples of X . The imputation can be implemented when we know the distribution of X because FHDI is a hot deck imputation. The distribution of X can be approximated by the distribution of Z , where Z is the categorized variable of X and $Z=\{1, \dots, k\}$. The empirical distribution function of X and Z , $\hat{F}_n(x)$ and $\hat{F}_k(z_x)$ are given by

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{x_i \leq x}, \quad (5.24)$$

Table 5.4: Comparison of RMSE values from predictions using datasets that were pre-cured by FHDI or a Naive method

Method	Response rate	Appliance				Air quality				Phenotype				Weather			
		FHDI		Naive		FHDI		Naive		FHDI		Naive		FHDI		Naive	
		RMSE	nRMSE	RMSE	nRMSE	RMSE	nRMSE	RMSE	nRMSE	RMSE	nRMSE	RMSE	nRMSE	RMSE	nRMSE	RMSE	nRMSE
GAM	10%	89.07	1.0054	90.67	1.0235	75.13	1.0077	80.72	1.0828	5.17	6.1030	13.57	15.9692	39.03	1.0958	29.03	0.8150
	20%	88.77	1.0021	90.53	1.0218	74.75	1.0026	79.13	1.0615	4.08	4.8105	20.06	23.6024	37.91	1.0643	32.02	0.8988
	30%	88.6	1.0002	89.93	1.0151	74.6	1.0006	81.60	1.0945	3.02	3.5612	23.88	28.0985	37.66	1.0574	33.95	0.9530
	40%	88.62	1.0003	89.67	1.0122	74.68	1.0017	83.44	1.1192	3.43	4.0477	26.35	30.9980	36.48	1.0243	35.58	0.9990
	50%	88.62	1.0003	89.37	1.0088	74.62	1.0009	84.39	1.1320	1.63	1.9217	27.74	32.6324	36.27	1.0182	36.32	1.0197
SVM	10%	81.28	1.1323	87.89	1.2244	64.92	1.2841	75.74	1.4980	3.65	1.5020	4.98	2.0536	20.62	1.2362	32.07	1.9222
	20%	76.63	1.0675	84.10	1.1716	60.19	1.1904	68.46	1.3541	3.19	1.3140	8.33	3.4322	19.30	1.1566	36.12	2.1646
	30%	75.18	1.0474	81.92	1.1412	57.84	1.1439	79.46	1.5716	3.09	1.2710	12.36	5.0906	18.14	1.0873	38.05	2.2809
	40%	73.30	1.0212	79.24	1.1039	55.46	1.0970	83.39	1.6493	2.89	1.1897	18.15	7.4782	17.97	1.0769	38.47	2.3059
	50%	72.81	1.0143	77.76	1.0833	53.75	1.0631	88.08	1.7420	2.47	1.0174	24.78	10.2098	17.63	1.0569	38.44	2.3041
ERT	10%	71.80	1.0796	72.28	1.0867	58.60	1.3358	63.55	1.4485	9.51	1.6912	16.31	2.8994	18.37	1.2458	25.52	1.7310
	20%	70.02	1.0528	70.47	1.0595	55.40	1.2629	58.62	1.3362	8.39	1.4925	22.64	4.0255	17.14	1.1626	29.28	1.9859
	30%	69.17	1.0400	70.05	1.0533	52.61	1.1992	65.89	1.5020	8.45	1.5022	24.72	4.3944	16.32	1.1069	31.23	2.1183
	40%	67.34	1.0125	68.50	1.0300	50.16	1.1434	69.58	1.5861	7.70	1.3691	26.45	4.7024	15.93	1.0807	32.81	2.2256
	50%	66.83	1.0048	68.18	1.0251	48.58	1.1074	71.84	1.6375	6.59	1.1709	28.21	5.0156	15.59	1.0575	32.93	2.2335
ANN	10%	86.64	1.0234	91.67	1.0829	67.23	1.0752	77.94	1.2467	5.92	1.8433	17.48	5.4422	21.53	0.9964	64.34	2.9780
	20%	84.88	1.0026	86.84	1.0257	65.79	1.0523	71.63	1.1456	4.49	1.3972	20.34	6.3307	21.58	0.9986	65.79	3.0448
	30%	86.40	1.0206	87.34	1.0316	65.24	1.0435	81.63	1.3056	4.36	1.3572	24.14	7.5158	21.58	0.9989	71.82	3.3237
	40%	84.42	0.9972	85.73	1.0126	63.76	1.0197	81.24	1.2994	3.97	1.2372	29.03	9.0378	21.19	0.9808	79.29	3.6696
	50%	84.08	0.9932	86.86	1.0260	64.01	1.0238	83.92	1.3422	3.99	1.2436	28.33	8.8198	21.22	0.9820	74.79	3.4615

$$\hat{F}_k(z_x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{z_i \leq z_x}, \quad (5.25)$$

where $\mathbf{1}$ is an indicator function; z_i is a converted value of x . We want to minimize $\sup_x |\hat{F}_k(z_x) - \hat{F}_n(x)|$. For some constant c and all $x \in \mathbb{R}$, $|\hat{F}_k(z_x) - \hat{F}_n(x)| \leq c|k^{-1} - n^{-1}|$, and $|k^{-1} - n^{-1}|$ converges to zero as $k \rightarrow n$, and $n \rightarrow \infty$. Therefore, this implies that we may have better imputation results when using large k value.

5.6.3 Impact of donor numbers

FHDI does not use all available donors to maximize computational efficiency. Instead, FHDI uses M donors selected by PPS sampling. In the case that the number of all available donors is less than specified M value, all available donors are used. The FHDI has additional variance than FEFI due to the selection of donors and the variance is shown in the second term in Equation 5.7. When M is large enough, the FHDI result is asymptotically close to the FEFI result. Table 5.5 summarizes the impact of M on the prediction accuracy. The nRMSE are almost constant as M increase, which means M of five is large enough for FHDI. Figure 5.5-5.8 show the change in M value is not likely to significantly affect the prediction performance because M of 5 is large enough. This result is in line with the recommendation from Rubin (1976) in which 2 to 10 is recommended as a donor number and M of 5 is a default value in the relevant library, **mice** (Buuren and Groothuis-Oudshoorn, 2011)

Table 5.5: Impact of donor numbers on prediction using the weather dataset with 50% response and GAM

M	5	15	25	35	45	55
nRMSE	1.0932	1.0933	1.0932	1.0932	1.0932	1.0930

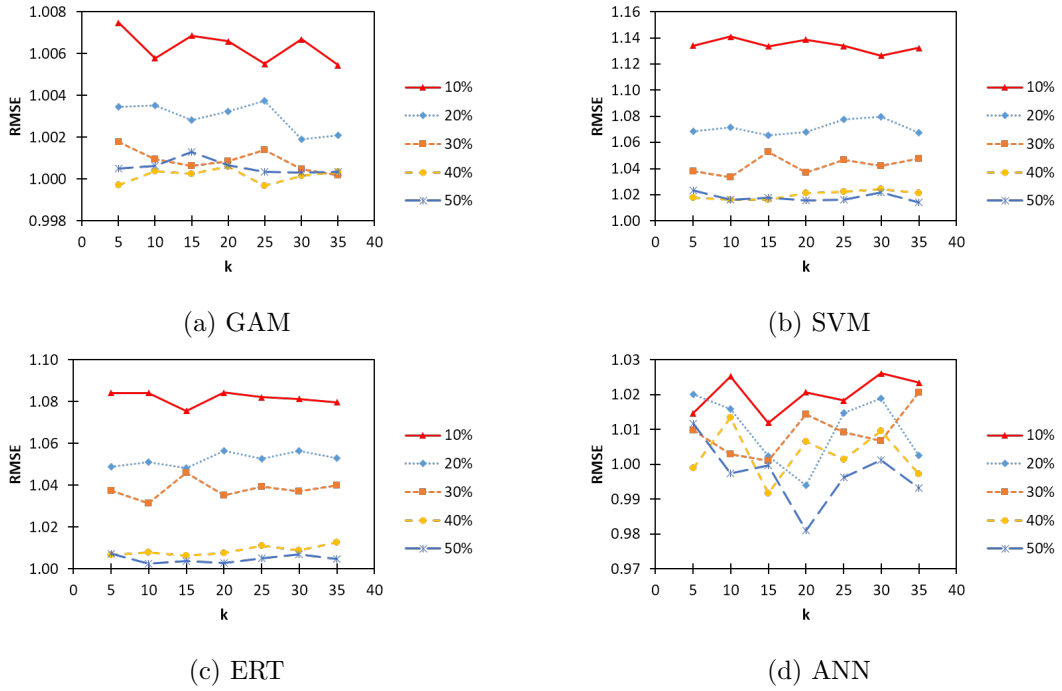


Figure 5.1: Impact of categorization numbers on prediction (appliance energy data set is used). 10 to 50% response rates are investigated.

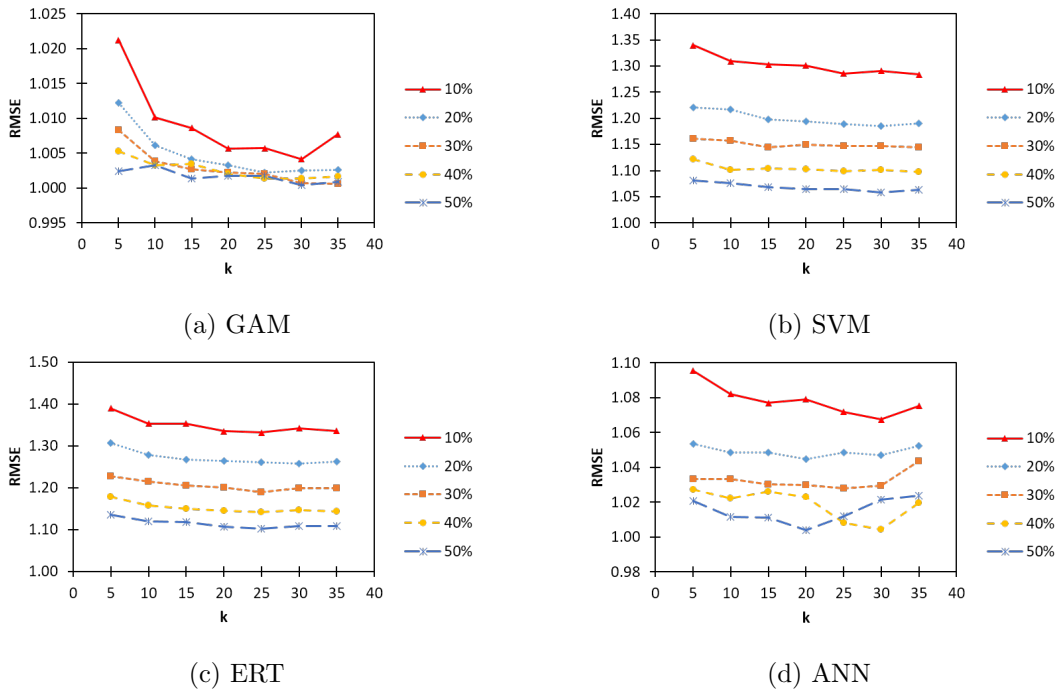


Figure 5.2: Impact of categorization numbers on prediction (air quality data set is used). 10 to 50% response rates are investigated.

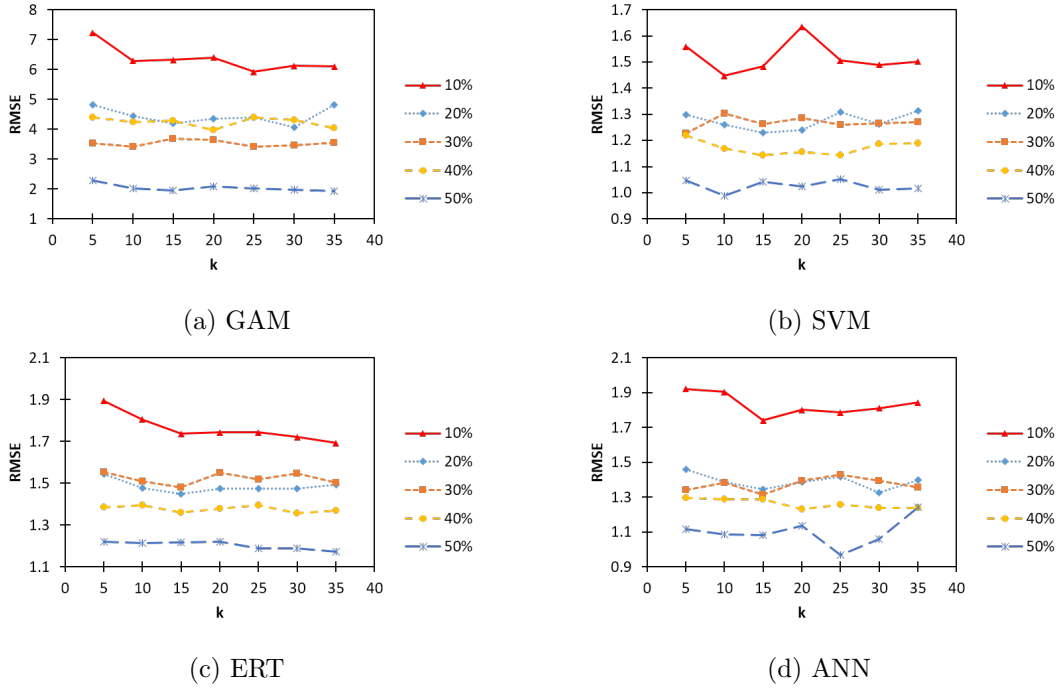


Figure 5.3: Impact of categorization numbers on prediction (phenotype data set is used). 10 to 50% response rates are investigated.

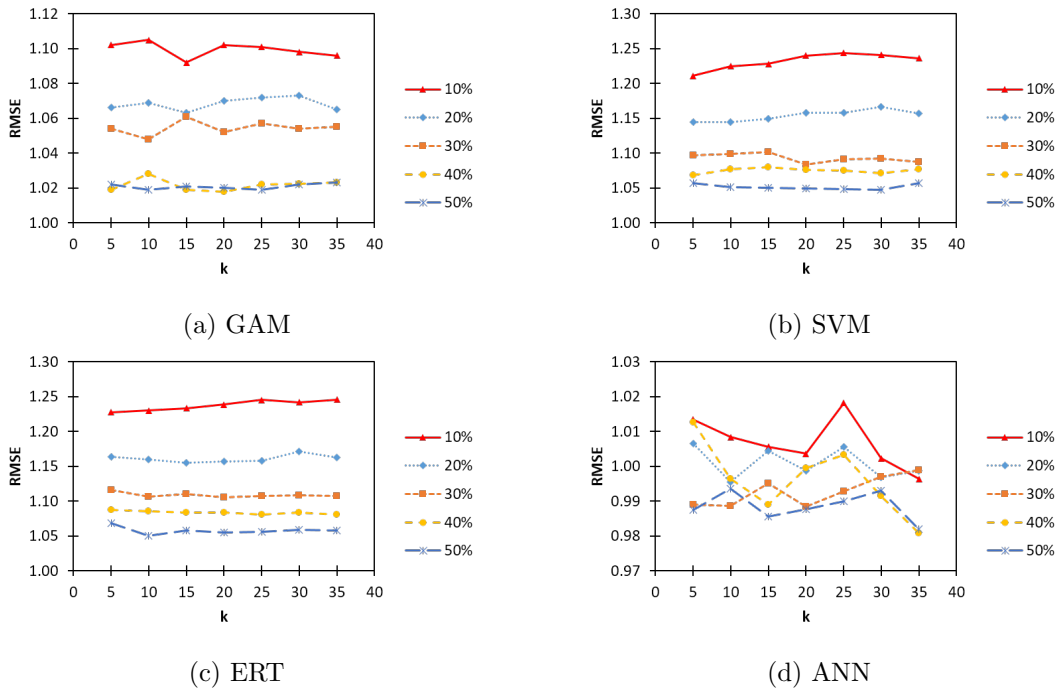


Figure 5.4: Impact of categorization numbers on prediction (weather data set is used). 10 to 50% response rates are investigated.

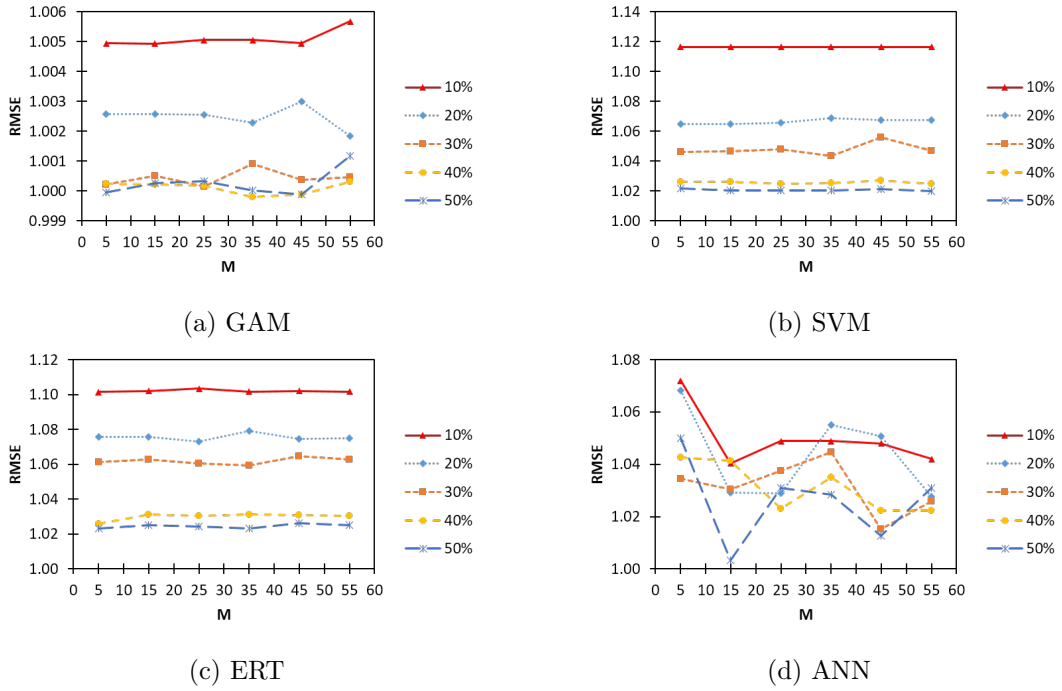


Figure 5.5: Impact of donor numbers on prediction (appliance energy data set is used). 10 to 50% response rates are investigated.

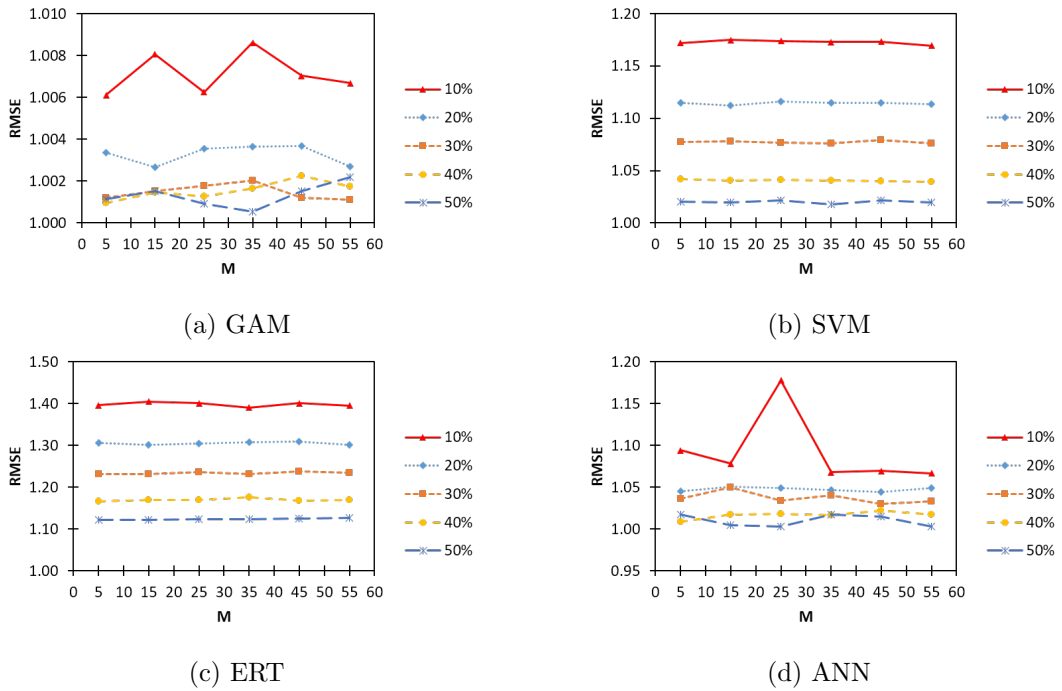


Figure 5.6: Impact of donor numbers on prediction (air quality data set is used). 10 to 50% response rates are investigated.

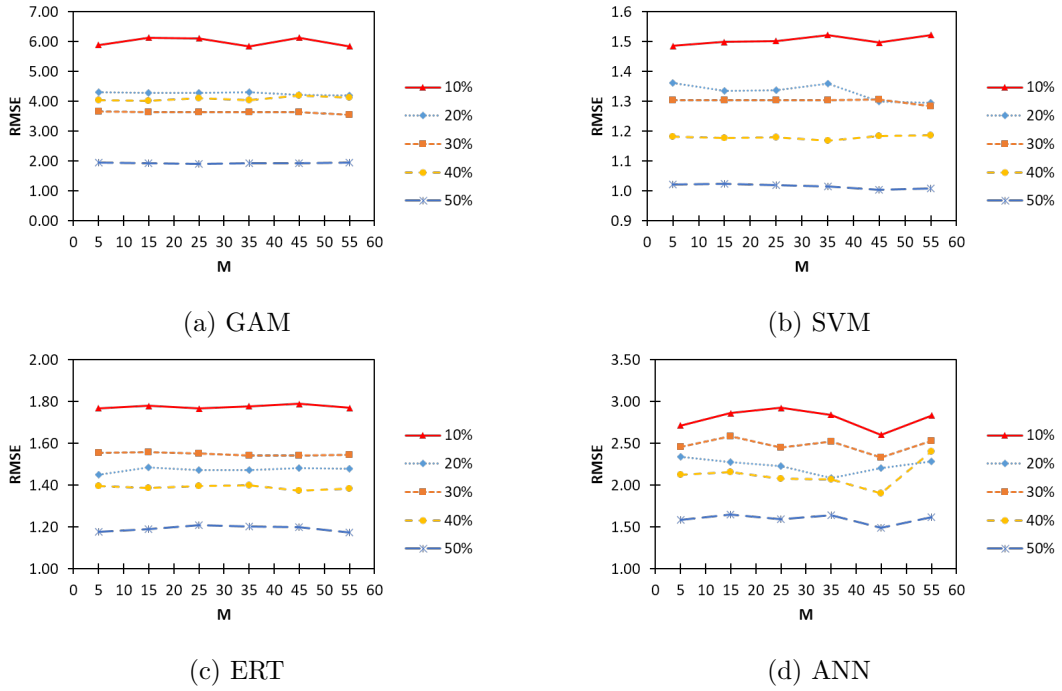


Figure 5.7: Impact of donor numbers on prediction (phenotype data set is used). 10 to 50% response rates are investigated.

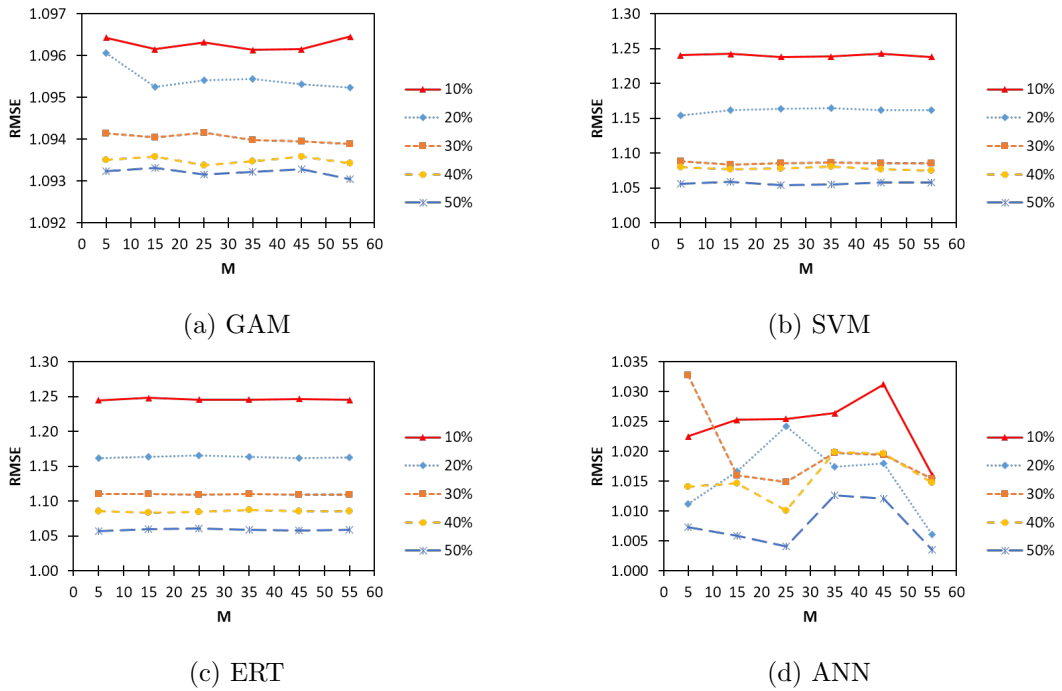


Figure 5.8: Impact of donor numbers on prediction (weather data set is used). 10 to 50% response rates are investigated.

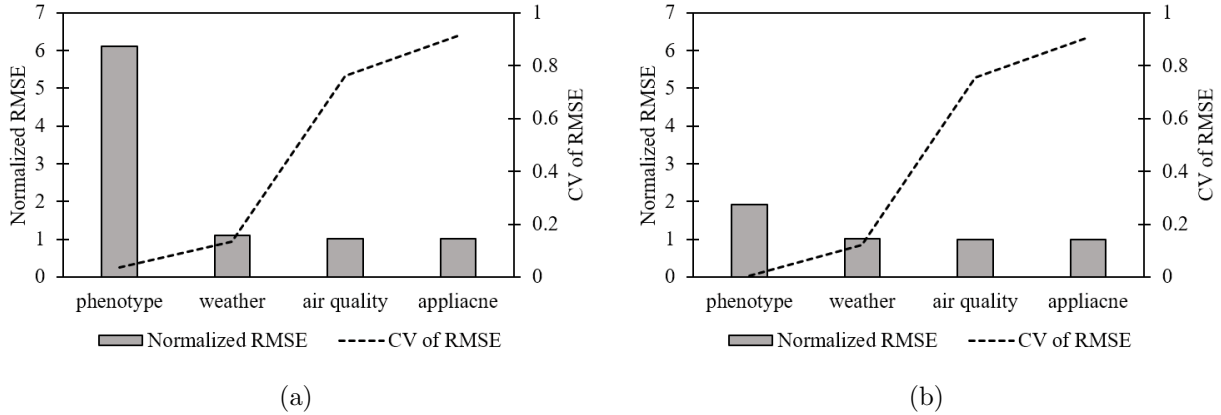


Figure 5.9: Relationship between coefficient of variance (CV) of RMSE and normalized RMSE from (a) 10%-dataset and (b) 50%-dataset.

5.6.4 Impact of extreme data missing

In this section, we investigate the impact of an extreme data missing on prediction performances. multiple datasets, having 10% to 50% missing rates with 10% interval, are intentionally generated using the appliance energy data set. It should be noted that 10% of instances of the original dataset is left without missing values since both FEFI and FHDI require fully observed instances for imputation. The other 90% data are used to introduce missing values. For example, suppose that we have 100 instances in an original dataset. 10 instances are left without any changes while missing values are made at other 90 instances. The datasets generated from the original dataset are cured by FHDI and the target response (i.e., appliance energy data set) is predicted using four regression methods. Figure 5.10 represents the prediction result. The percentage in Figure 5.10 represents missing rates, not response rates. As the missing rate changed from 10% to 50%, the minimum and maximum increment rates of RMSE are about 4% and 29% when using SVM and ERT, respectively. Also, as the missing rate changed from 10% to 50%, RMSE increases are about 11% and 5.5% when using GAM and ANN, respectively. This result suggests that depending upon data type the impact of high missing rate may be substantially large.

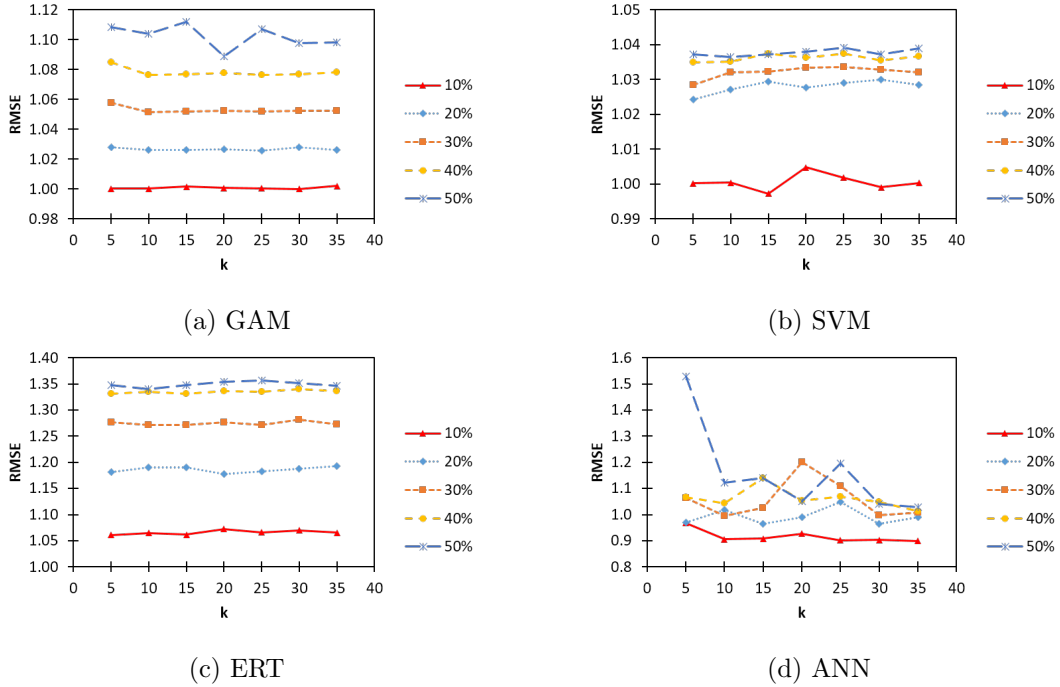


Figure 5.10: Impact of extreme missing rates on prediction (appliance energy data set is used). 10 to 50% missing rates are investigated.

5.7 Conclusions

This study investigated behaviors and impacts of FHDI on prediction performance of statistical and machine learning methods. We explored and examined various cases using four different practical engineering datasets under response rates ranging from 10% to 50% and a wide range of FHDI's two internal parameters (i.e., categorization numbers k and the number of donors M). Amongst popular statistical and machine learning methods, we adopted GAM, SVM, ERT, and ANN to understand the quantitative impacts of FHDI on the regression prediction. With the normalized RMSE (nRMSE) being a metric for prediction accuracy, results show that the FHDI outperforms the simple naive method which fills in missing data using mean values of attributes, and also confirm the gradually increasing positive influence of FHDI on improving prediction performance as the response rates increase. Detailed case studies for k and M suggest that k within 30 and 35 and M value around 5 are recommendable for general engineering data. This recommendation

appears to be in line with the recommended donor number of the Rubin's multiple imputation. An investigation into the extreme missing data cases shows that the prediction accuracy is significantly affected, e.g., the maximum increment of nRMSE was about 30% as the missing rate is increased to 50% from 10%. The research results presented herein will benefit a broad audience of engineering domains. Particularly, general engineering missing data can be tackled by an assumption-free, easy-to-use imputation method like FHDI, with which subsequent data analyses can be facilitated with a better statistical rigor.

5.8 Acknowledgments

This research is supported by the research funding of Department of Civil, Construction, and Environmental Engineering of Iowa State University. The parallel computing research reported herein is partially supported by the HPC@ISU equipment at ISU, some of which has been purchased through funding provided by NSF under MRI grant number CNS 1229081 and CRI grant number 1205413. This work was supported (in part, I. Cho) by the National Science Foundation under grants CBET-1605275. The data sharing of Dr. Lawrence and Dr. Cetin is appreciated.

Bibliography

- Bache, K. and Lichman, M. (2013). UCI machine learning repository.
- Batista, G. E. A. P. A. and Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533.
- Brown, M. L. and Kros, J. F. (2003). Data mining and the impact of missing data. *Industrial Management & Data Systems*, 103(8):611–621.
- Buuren, S. v. and Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- Duchon, J. (1977). *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, pages 85–100. Springer, Berlin, Heidelberg.
- Farhangfar, A., Kurgan, L., and Dy, J. (2008). Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41(12):3692–3705.
- Fritsch, S. and Guenther, F. (2016). *neuralnet: Training of Neural Networks*. R package version 1.33.
- Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.
- Ghasemi, F., M. A. F. A. P.-S. H. (2017). Deep neural network in biological activity prediction using deep belief network. *Applied Soft Computing*, 62(251).
- Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC Press, Boca Raton, FL, USA.
- Heltshe, S. L., Lubin, J. H., Koutros, S., Coble, J. B., Ji, B.-T., Alavanja, M. C. R., Blair, A., Sandler, D. P., Hines, C. J., Thomas, K. W., Barker, J., Andreotti, G., Hoppin, J. A., and Beane Freeman, L. E. (2012). Using multiple imputation to assign pesticide use for non-responders in the follow-up questionnaire in the agricultural health study. *J Expos Sci Environ Epidemiol*, 22(4):409–416.
- Im, J., Cho, I., and Kim, J. (2018). *FHDI: Fractional Hot Deck and Fully Efficient Fractional Imputation*. R package version 1.2.2.
- Im, J., Kim, J.-K., and Fuller, W. A. (2015). Two-phase sampling approach to fractional hot deck imputation. In *Proceedings of the Survey Research Methods Section*, pages 1030–1043.
- Jang, J., Sun, C., and Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. NJ:Prentice-Hall.

- Kalton, G. and Kish, L. (1984). Some efficient random imputation methods. *Communications in Statistics-Theory and Methods*, 13(16):1919–1939.
- Kamdar, H., Turk, M., and Brunner, R. (2016). Machine learning and cosmological simulations ii. hydrodynamical simulations. *Monthly Notices of the Royal Astronomical Society*, 457(2):11621179.
- Kim, J. K. and Fuller, W. (2004). Fractional hot deck imputation. *Biometrika*, 91(3):559–578.
- Lin, C.-C., Li, C.-I., Liu, C.-S., Lin, W.-Y., Lin, C.-H., Yang, S.-Y., and Li, T.-C. (2017). Development and validation of a risk prediction model for end-stage renal disease in patients with type 2 diabetes. *Scientific Reports*, 7(1):10177.
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2017). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.6-8.
- Roth, P. L. (1994). Missing data: A conceptual review for applied psychologists. *Personnel Psychology*, 47(3):537–560.
- Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.
- Simm, J., de Abril, I. M., and Sugiyama, M. (2014). *Tree-Based Ensemble Multi-Task Learning Method for Classification and Regression*.
- Song, I., Cho, I., Phares, B., and Sharma, A. (2018a). A computational framework for statistical data-curing and prediction of bridge and traffic big data. *Manuscript submitted for publication*.
- Song, I., Cho, I., Tessitore, T., Gurcsik, T., and Ceylan, H. (2018b). Data-driven prediction of runway incursions with uncertainty quantification. *Journal of Computing in Civil Engineering*, 32(2):04018004.
- Song, I., Cho, I. H., and Wong, R. K. W. (2018c). An advanced statistical approach to data-driven earthquake engineering. *Journal of Earthquake Engineering*, 0(0):1–25.
- Su, X., Khoshgoftaar, T. M., Zhu, X., and Greiner, R. (2008). Imputation-boosted collaborative filtering using machine learning classifiers. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 949–950. ACM.
- Wang, J., Tsapakis, I., and Zhong, C. (2016). A spacetime delay neural network model for travel time prediction. *Engineering Applications of Artificial Intelligence*, 52(Supplement C):145–160.
- Wood, S. (2006). *Generalized additive models: an introduction with R*. CRC Press.
- Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1):3–36.

- Yang, S. and Kim, J. K. (2016). Fractional imputation in survey sampling: a comparative review. *Statistical Science*, 31(3):415–432.
- Yoo, K. D., Noh, J., Lee, H., Kim, D. K., Lim, C. S., Kim, Y. H., Lee, J. P., Kim, G., and Kim, Y. S. (2017). A machine learning approach using survival statistics to predict graft survival in kidney transplant recipients: A multicenter cohort study. *Scientific Reports*, 7(1):8904.

CHAPTER 6. A COMPUTATIONAL FRAMEWORK FOR STATISTICAL DATA-CURING AND PREDICTION OF BRIDGE AND TRAFFIC BIG DATA

A technical report published in the Midwest Transportation Center

Ikkyun Song, In-Ho Cho, Brent Phares, and Anuj Sharma

Abstract

Systematic accumulation of bridge and traffic big data has been successful by virtue of advanced structural health monitoring systems and automated sensing technologies. Still, active use of data for long-term decision-making and strategic planning is in its infancy owing to big data-rooted challenges including severe data complexity, high dimensionality, intrinsic missing data, lack of powerful learning and prediction methods, etc. This study sought to develop a computational framework that can transform, merge, and importantly cure bridge and traffic big data to improve statistical learning and prediction. We produced a hybrid big data by merging bridge and traffic data for which we introduced an assumption-free multivariate imputation method for curing intrinsic missing data. A parallel computing algorithm was implemented for scalability. Validations focused on years-long structural and traffic sensor data collected from a target bridge in Iowa. Results show that the proposed framework appear to help improve statistical quality and prediction accuracy.

6.1 Introduction

Recently, data-driven research has been essential in the engineering fields, enabling researchers to gain valuable knowledge from data. Examples can be found in broader engineering domains. For instance, Lv et al. (2015) developed a traffic flow model using a deep learning method while Perera and Mo (2016) used a similar deep learning approach to generate a condensed database

regarding ship performances and navigation information for a general use. Le and Jeong (2017) developed a methodology for integrating heterogeneous construction engineering terminologies into representative terms by using a neural network.

Growing community-level databases are also noteworthy. Due to advances in strain measurement technologies, bridge health monitoring (BHM) systems using various types of sensors were developed to systematically accumulate relevant data (e.g., (Jang et al., 2010; Ko and Ni, 2005; Li et al., 2004; Ntotsios et al., 2009)). Despite this active collection, the databases have been rarely used to build prediction models for data-driven bridge managements. Li et al. (2003) proposed a statistical model to represent a specific daily cycle for fatigue assessment of a specific bridge using multiple linear regressions. Yet, the general use of such a specific model is challenging because the daily strain history pattern and the pulse size do not remain constant. Rather, they may fluctuate depending on other factors such as ambient temperature and real-world traffic flows. Therefore, generalized predictive models for bridge strain data need to be developed for general use.

Another significant problem is missing data. In the BHM system, missing data issue appears inevitable due to many causes including human-induced accidents or mistakes, mechanical malfunctions, or environmental disruptions. The dataset from a real-world target bridge used in this study also has substantial missing values at some timeframes due to sensor malfunctions, irregular measurement times, traffic closure for maintenance, etc. To overcome the missing data issue with a statistical rigor, this study adopted one of the most flexible and general statistical imputation methods, the so-called fractional hot-deck imputation (FHDI) method. As shall be presented in detail, all the missing values of bridge big data have been cured by FHDI prior to building a statistical prediction model.

To achieve a higher predictive power and generality of the proposed framework, this study adopted the generalized additive model (GAM) (Hastie and Tibshirani, 1990). GAM has been mainly used to develop a general and flexible prediction core for strain behavior of bridges. GAM is a flexible, nonparametric statistical model which has little restrictions on the number of variables

and complex distributions of large data. GAM's high prediction accuracy and flexibility have been well demonstrated by authors' prior works (Song et al., 2017, 2018b,c).

Objectives of this study are to (1) develop a systematic computational framework for collecting, transforming and merging bridge and traffic data, (2) create a hybrid bridge-traffic dataset, (3) apply an advanced data-curing method to the hybrid dataset, (4) develop a statistical prediction model with the best combination of predictors based on a direct search algorithm, and (5) investigate impacts of data-curing and inclusion of hybrid data on the prediction accuracy improvement. It should be noted that the target responses are long-term behavior prediction rather than real-time fluctuation prediction, for which a future extension will be developed.

The outline of this paper is as follows. The central procedure of data collection, transformation, and the fusion of bridge sensors and traffic flow data will be addressed. The statistical theories of a data-curing method (i.e. FHDI) and statistical prediction of GAM will be summarized. A direct search algorithm in conjunction with GAM analyses will be presented to explain how to find the best variable combination. A comparison against a correlation-based variable selection approach will follow. The impact of data-curing and the hybrid data on prediction accuracy improvement will be addressed. Before conclusion, a parallel computing strategy tailored for the algorithms of this study will be provided.

6.2 Methodology

6.2.1 Data collection

The target bridge is located in the eastbound I-80 over Sugar Creek in Iowa. 71 sensors are installed in multiple locations of the bridge (i.e., 53 on the bottom and 18 on the top) to measure strains on the top and bottom flanges, and temperatures of steel, concrete, and air. The detailed instrumental plan is shown in Figure 6.1.

Each sensor measures strains and temperatures at its location with the frequency of 250 Hz. A raw data file was generated for every minute from June 2014 to October 2016, and a single file includes all the data (i.e., date, time, temperature, and strain) measured by all sensors. Such all

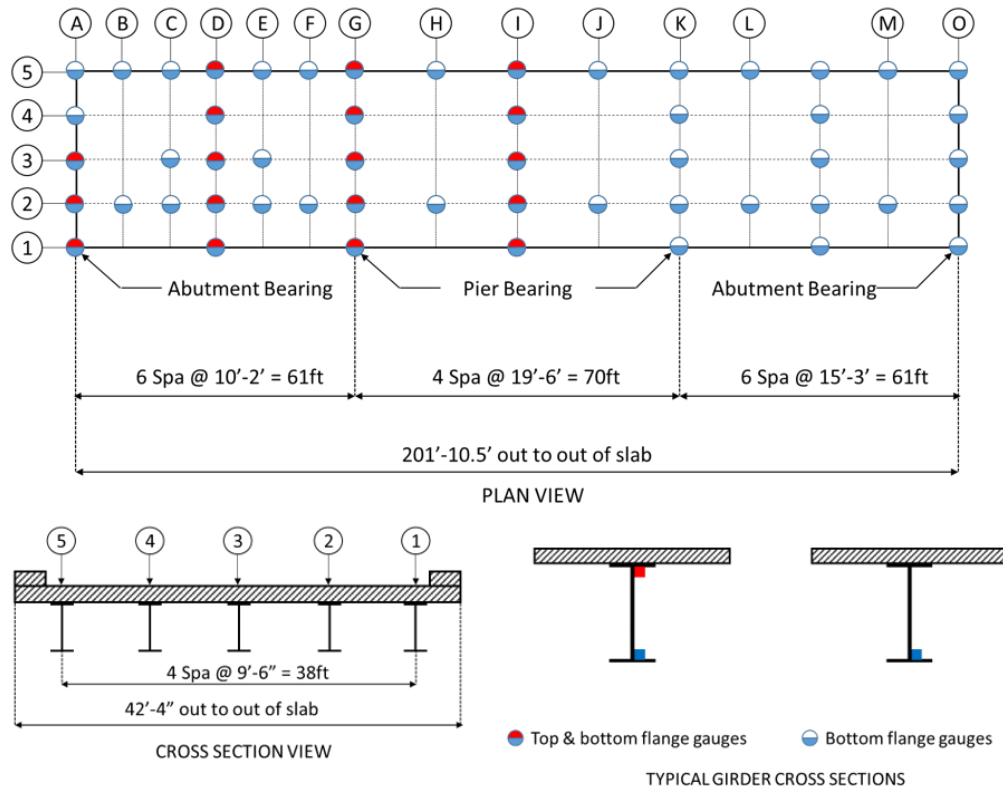


Figure 6.1: Instrumentation plan of sensors of the target bridge

data in short time intervals are not appropriate for the long-term prediction; therefore the raw data needs to be squeezed and converted to an interpretable form to facilitate the subsequent statistical analyses. The procedure of extraction and transformation of data shall be described in the following section.

6.2.2 Data extraction and transformation

The raw data files are text-based files, and thus size is too large (6 terabytes) to be directly used in statistical inferences such as variable selection using multiple GAM analyses. Therefore, we processed the raw files to extract the only information we want and generated compact binary files using high-performance computing (HPC) techniques. The information related to peak strains is extracted from raw data and stored in binary files while other information, such as strain values between peak strains, are discarded. From this step, the data size reduces from 6 terabytes to 1

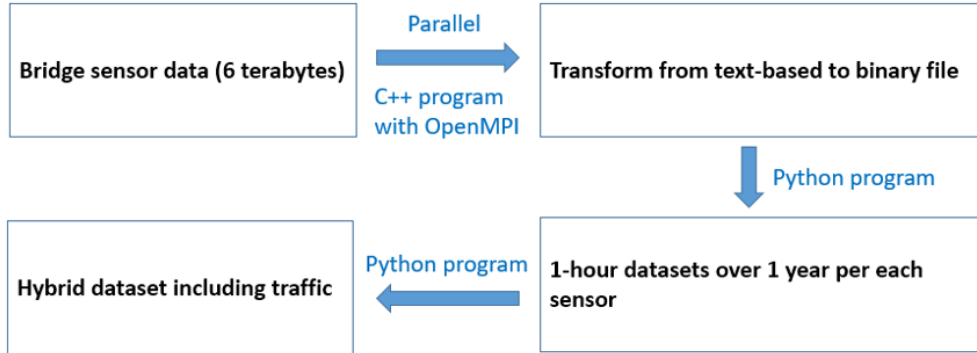


Figure 6.2: Flow chart showing data-transformation from raw bridge and traffic data to the final hybrid data set

gigabyte. The parallel strategy for this procedure shall be addressed in the latter section. The entire workflow of these procedures is shown in Figure 6.2.

First, we transformed text-based files into binary files that include peak strains in pulses (see Figure 6.3a). The top and bottom peak strains are determined such that strains more than 5μ (hereafter, μ stands for 10^{-6}) away from the median strain value. Peak strains adjacent to the median value of strains (i.e., less than 5μ strain from the median) are considered as noises. Here, the reason for selecting the 5μ as a threshold is that there exist a number of peak strains within 5μ and so, considering that the yield strain of steel is 0.002, those strains might not be significant compared to peak strains outside the 5μ range (see Figure 6.3b). Note that since median strain values are changed over time, peak strains are determined based on the median strain over 1 minute.

Next, the binary files are transformed to 1-hour csv-formatted datasets in which one instance contains 8 digits of date (e.g., 20161115), month, day, hour, day of week, steel temperature, concrete temperature, air temperature, median strain, number of measurement and frequencies of peak strain. Peak strains have the bin size of 5μ and the range is between -100μ and 100μ . An example of the histogram of peak strains, measured by one sensor over 1 year, is shown in Figure 6.4. The noticeable range of peak strains appears to be between -20μ and 20μ , but this range varies depending on sensor locations. The summary of datasets during these transformation steps is shown in Table 6.1.

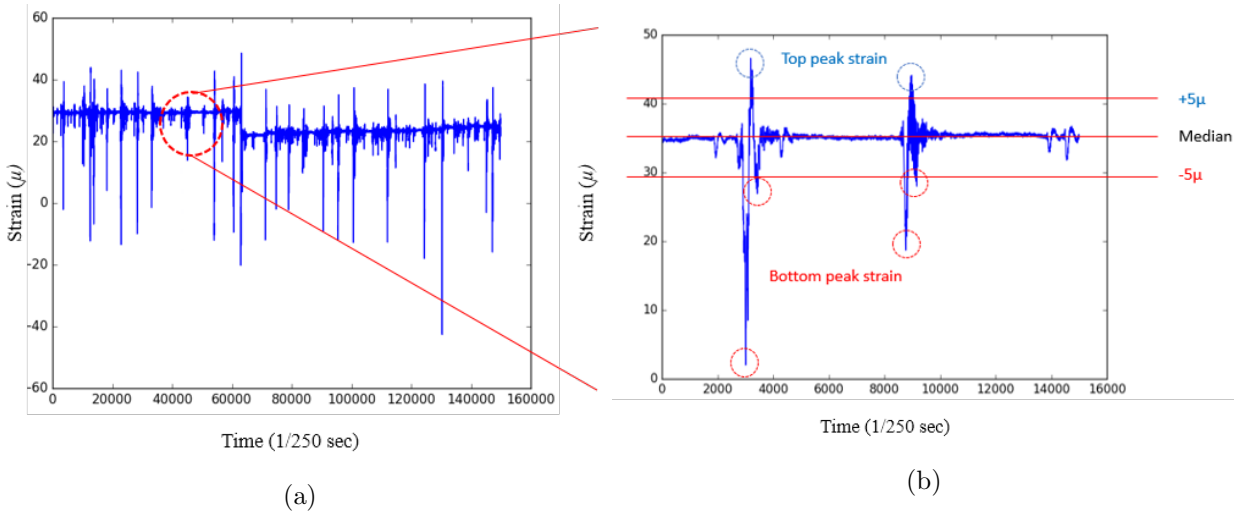


Figure 6.3: Strain history over (a) 10 minutes and (b) 1 minute. *Top* peak and *bottom* peak strains are selected outside the range between $+5\mu$ and -5μ from the median strain value

6.2.3 Data merging with traffic data

Traffic is directly related to strain behavior in bridges. Heavy traffic generates a large number of strain fluctuations and traffic by large vehicles (i.e., truck) produces large strain peak values while small vehicles generate small strain peak values. Traffic information, therefore, may significantly impact on the prediction of bridge strain response. To investigate this impact, traffic data measured from a location near the target bridge is merged into the bridge strain dataset for investigation. The traffic is measured per five minutes and it has three categories: i.e., small-, medium-, and large-sized vehicle.

6.2.4 Data curing: FHDI

FHDI (Kim and Fuller, 2004) is one of the advanced statistical methods to cure missing data. It has little need of statistical assumption and prior knowledge about the original data because FHDI takes an advantage of hot-deck imputation in which imputed values are only taken from observed samples. FHDI also provide a consistent variance estimation while the multiple imputation (Rubin, 1987) estimates a variance inconsistently. These features make reasonable and reliable data curing.

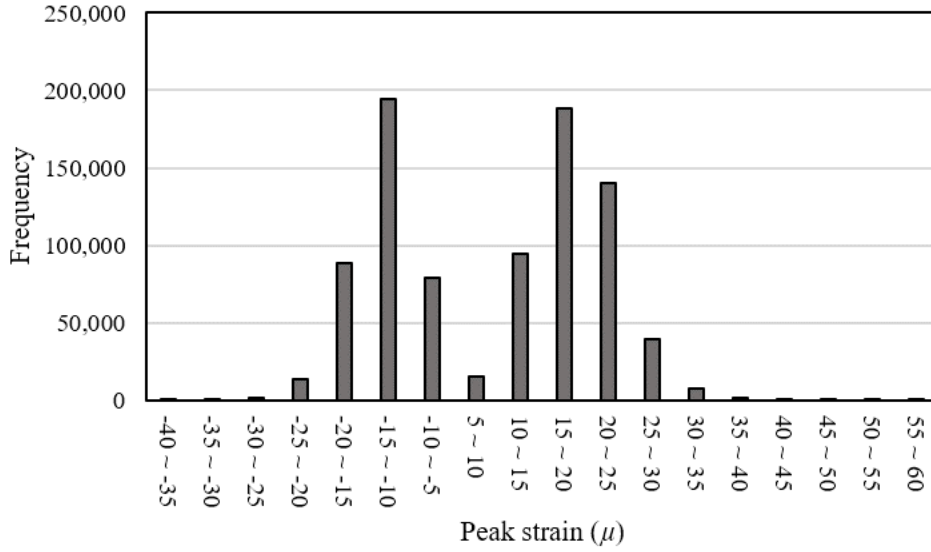


Figure 6.4: Histogram of peak strains

FHDI generates *donors* to be used to fill in missing values. Here, *donors* are sample sets that have imputed values and are calculated using the imputation estimators. For curing one missing value, multiple *donors* are used. Depending on how to select *donors*, there are two imputation estimators: (1) fully efficient fractional imputation (FEFI) estimator and (2) FHDI estimator. FEFI uses all *donors* for curing missing values while FHDI uses some selected *donors*. The imputation estimators are shown in Equations 6.1 and 6.2.

$$\hat{Y}_{FEFI} = \sum_{c=1}^C \sum_{i \in A_c} \omega_i \left\{ \delta_i y_i + (1 - \delta_i) \sum_{j \in A} \omega_{ij,FEFI}^* y_j \right\}. \quad (6.1)$$

where c is index for partitioned groups where it takes values on $\{1, 2, \dots, C\}$; A is the index set of all samples and is partitioned into C groups; A_c is index set of a group; ω_i is sampling weight of i^{th} recipient; y_i is the i^{th} recipient; $\delta_i = 1$ when y_i is observed, otherwise $\delta_i = 0$; $\omega_{ij,FEFI}^*$ is fractional weight for the FEFI estimator.

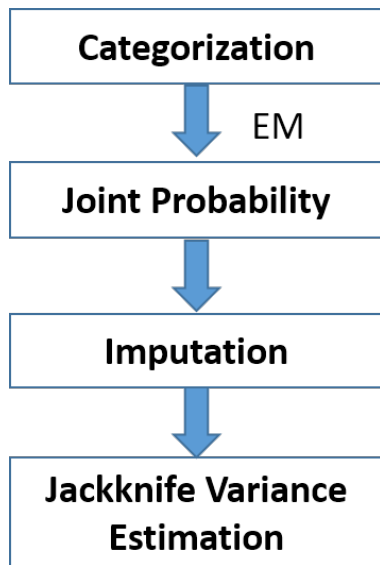
$$\hat{Y}_{FHDI} = \sum_{c=1}^C \sum_{i \in A_c} \omega_i \left\{ \delta_i y_i + (1 - \delta_i) \sum_{j \in A} \omega_{ij,FHDI}^* y_i^{(j)} \right\}. \quad (6.2)$$

Table 6.1: Summary of datasets during the transformation process from the raw data to the final dataset

Dataset name (data format)	Attribute	Description
Raw data (text-based format)	Date, time, temperature, strain	Raw data measured with 250 Hz by sensors installed in the bridge
Binary data (binary format)	Date, time, average temperature, peak strain, number of measurement	A single instance contains information for 1 minute
1-hour dataset (csv format)	Date, time, day of week, average temperature, number of measurement and median of strain over 1 hour, strain frequencies	A single instance contains information for 1 hour
1-hour dataset with traffic (csv format)	Date, time, day of week, average temperature, number of measurement and median of strain over 1 hour, strain frequencies, traffic	Final hybrid dataset merged with traffic data

where $\omega_{ij}^* y_i^{(j)}$ is the fractional weight for the FHDI estimator; $y_i^{(j)}$ is the i_{th} imputed value of y_i ; M is number of *donors*.

For the implementation of FHDI, the *R* package named by FHDI (Im et al., 2018) is used. Figure 6.5 shows four steps for the implementation and Figures 6.5b through 6.5d show the change of the dataset throughout the steps. First, all samples are partitioned into multiple sets of groups to secure enough number of *donors* for imputation. Here, the initial number of groups, k and *donors*, M are set by users. If *donors* are not enough, some groups are combined to secure enough *donors* (Figure 6.5c). At least two *donors* are required for imputation. The impacts of k and M are investigated in (Song et al., 2018a). Once groups for variables are determined appropriately, joint probabilities of *donors* are calculated for each recipient. Here, the modified Expectation-Maximization (EM) algorithm is used to estimate joint probabilities. An EM algorithm is an iterative method to find the maximum likelihood estimate of a parameter which is a joint probability in this study. In the first E-step, the initial conditional probabilities are computed and then the conditional probabilities are updated to maximize the likelihood in the M-step. The updated probabilities enter the E-step. These procedures continue until it converges. Lastly, with selected *donors*, the missing values are filled using the conditional probabilities. Variance is estimated using a jackknife method (see (Im et al., 2015, 2018) for detail).



(a)

V1	V2	V3	V4	V5
NA	NA	NA	-0.63	4.69
NA	0.71	2.47	-0.08	-0.40
-0.45	NA	2.84	-1.81	3.85
-0.01	NA	0.85	-2.79	3.32
0.46	1.10	1.74	NA	-0.14

(b)

V1	V2	V3	V4	V5
0	0	0	3	4
0	2	2	2	1
1	0	3	2	5
1	0	1	2	4
3	2	2	0	2

(c)

V1	V2	V3	V4	V5
0.74	2.33	3.41	-0.63	4.69
1.65	0.71	2.47	-0.08	-0.40
-0.45	3.17	2.84	-1.81	3.85
-0.01	2.02	0.85	-2.79	3.32
0.46	1.10	1.74	-1.88	-0.14

(d)

Figure 6.5: Example of key procedures for FHDI: (a) entire flow chart; (b) original dataset in which the NA stands for a missing value; (c) categorized dataset; (d) cured dataset

6.3 Statistical Learning and Prediction

6.3.1 Summary of generalized additive model

Generalized additive model (Hastie and Tibshirani, 1990) is a generalized linear model, holding strong flexibility and general applicability. It uses an unspecific smoothing function rather than relying on predefined distributions or parameters. By virtue of the unspecified smoothing function, the predictors do not need to have a set of parameters, where predictors mean independent variables in regression models. GAM is formulated by predicting target of i^{th} sample (denoted by $Y_i \in \mathbb{R}$) with n predictors (denoted by $\mathbf{x}_{ij} \in \mathbb{R}^n$ where $1 \leq j \leq n$). The general form of GAM can be represented as:

$$Y_i = g(\mu_i) = \sum_j f_j(x_{ij}), \quad (6.3)$$

where g is a smooth link function; the expectation of Y_i conditional on \mathbf{x}_i ($\mathbb{E}(Y_i | \mathbf{x}_i)$) is denoted by μ_i ; Y_i is a target response from an exponential family of distribution (e.g., normal, binomial, or gamma distribution); f_j are smooth functions of covariates x_{ji} (Wood, 2006). Essentially, GAM has a non-parametric smooth function for each covariate. Simply explaining, the following description includes a single variable, but generalization for multiple variables is straightforward (Wood, 2006). Let GAM be $\mathbb{E}(Y | x) = f(x)$, and the smoothing function f can be represented as:

$$f(x) = \sum_{j=1}^k b_j(x)\beta_j \quad (6.4)$$

where b_j is the j^{th} basis function and β_j is an unknown parameter. The model can be fit by maximizing the corresponding likelihood. A penalty term is given as $\lambda \int [f''(x)]^2 dx$ where λ is smoothing parameter. If λ is too large, it is an over-smoothed estimated while it is under-smoothed estimated if λ is too small. This error is getting greater in both directions. The λ value is optimized by minimizing generalized cross-validation score (Golub et al., 1979) and selected appropriately via the relevant GAM library. Therefore, there is little need to manually adjust the λ value (Song et al., 2018c).

In sum, GAM requires no prejudice on relations among parameters and holds little restriction to the number of variables and nonlinear distribution of variables. Importantly, GAM's internal

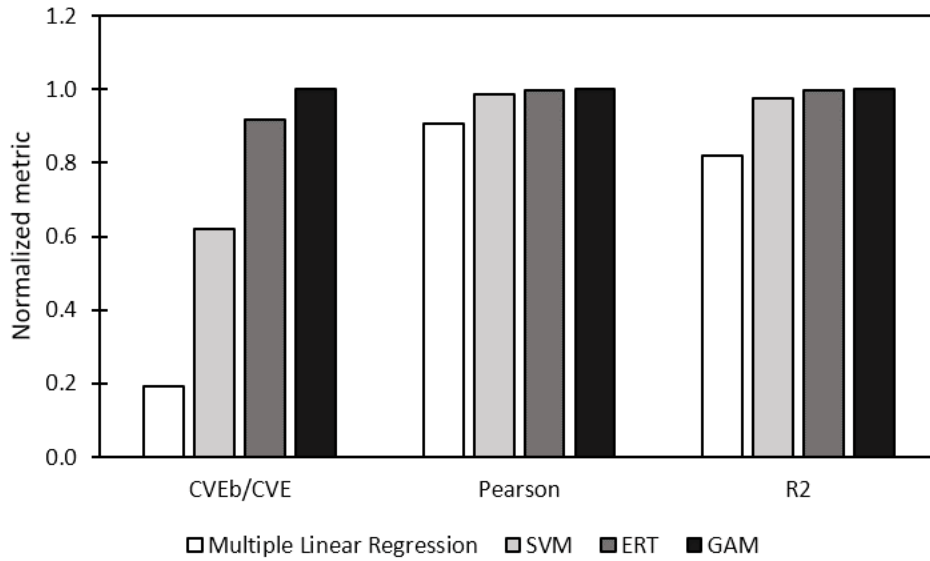


Figure 6.6: Comparison of prediction performance between GAM and other methods. In vertical axes, the higher value indicates the higher prediction accuracy. (cited from Song et al. (2018b))

setting always seeks to balance the fitting accuracy and smoothness, in which the generality and flexibility of GAM are rooted.

6.3.2 Excellent performance of GAM compared to SVM and ERT

In addition to the flexibility of GAM owing to unspecified smooth functions, GAM also performs well in terms of prediction accuracy. In the previous work (Song et al., 2018c), the GAM showed a better performance compared to well-known multiple linear regression and two popular machine learning algorithms (i.e., support vector machine (SVM) and extremely randomized trees (ERT)). The comparison result is shown in Figure 6.6 in which three metric values are normalized by the values of GAM: CVE_b/CVE is the ratio of base cross-validation error (CVE_b) to cross-validation error (CVE), Pearson is the Pearson correlation coefficient, and R^2 is the coefficient of determination (Song et al., 2018b). The result shows GAM outperforms than multiple regression and slightly performs better than SVM and ERT.

Another advantage of GAM compared to ML is that because GAM is a statistical regression model, a prediction result by GAM can be clearly explained based on statistical theories and methodologies while, for ML methods, the pathway from predictors to a response is likely to be unclear due to the arbitrary nature and randomness of ML methods. This advantage of the statistical model makes the prediction process interpretable and allows researchers to build a better predictive model according to their statistical knowledge.

6.3.3 Direct search versus correlation-based predictor selection

To find the best predictors, GAM models are built using multiple predictor combinations. 13 variables are used as predictors and 7 variables are used as responses. The summary of predictor and response variables is shown in Table 6.2. The best predictor combinations are selected using two different approaches: the correlation method and the direct search algorithm suggested by authors (Song et al., 2018b).

For the correlation method, best predictors are selected based on correlation values (Table 6.3). For instance, if a GAM model is built using three predictors, three predictors with the top 3 correlation values are selected. For the direct search method, all possible combinations are examined without any prejudice on relations among predictors and responses or any statistical inference. For example, when 7 predictor variables are used for finding the best predictor combination, there are total 1,716 (i.e. $[13!/7!(13-7)!]$) combinations to be examined. The computation cost, therefore, is very expensive, and so the serial computing will require a long running time. A parallel computing algorithm is developed using *Rmpi* (Yu, 2002) to distribute assigned searching tasks. *Rmpi* is a library for parallelization. The detailed parallel strategy shall be explained in the later section.

The comparison of the prediction performance of GAM model between the two different approaches is shown in Figure 6.7. Root Mean Squared Error (RMSE) values are normalized by the highest RMSE value. When a small number of predictors are selected, the prediction performance using the direct search algorithm is better than that using correlation, and the predictors selected by each method are different. For instance, when 2 predictors are used, the predictors selected are

Table 6.2: Summary of predictor and response variable for GAM model

Role	Vriable	Types	Description
Predictor	Date	Integer (continuous)	8-digit number of date (e.g., 20150723)
	Month	Integer (categorical)	Categorical number for month (e.g., 1 and 12 indicate January and December)
	Day	Integer (categorical)	Categorical number indicating day (i.e., 1 through 31)
	DOW	Integer (categorical)	Categorical number indicating day of week (e.g., 0 and 6 indicate Sunday and Saturday)
	Hour	Integer (categorical)	Categorical number indicating hour (i.e., 0 through 23)
	steelTemp	Float (continuous)	Steel temperature ($^{\circ}$ F) for 1 hour
	concTemp	Float (continuous)	Concrete temperature ($^{\circ}$ F) for 1 hour
	airTemp	Float (continuous)	Air temperature ($^{\circ}$ F) for 1 hour
	strainMedian	Float (continuous)	Median strain value for 1 hour (μ)
	nMeasurement	Integer (continuous)	Count of strain measurement for 1 hour
	smallCar	Integer (continuous)	Traffic count by small size of vehicle for 1 hour
	mediumCar	Integer (continuous)	Traffic count by medium size of vehicle for 1 hour
	largeCar	Integer (continuous)	Traffic count by large size of vehicle for 1 hour
Response	strainMean-Bottom	Float (continuous)	Expected value of the bottom peak strains for 1 hour
	strainMean-Top	Float (continuous)	Expected value of the top peak strains for 1 hour
	strainMin	Integer (continuous)	Minimum peak strain value for 1 hour (μ)
	strainMax	Integer (continuous)	Maximum peak strain value for 1 hour (μ)
	strainSTD	Float (continuous)	Standard deviation of peak strain (μ)
	area	Integer (continuous)	Area under strain distribution

Table 6.3: Correlation among variables

	Month	Day	Hour	DOW	steel-Temp	conc-Temp	air-Temp	strain-Median	Measurement	small-Car	medium-Car	large-Car	Date	area	strain-Max	strain-Mean	strain-Mean-Top	strain-Min	strain-STD
Month	1	0.008	0	0.007	0.327	0.3	0.327	0.478	0.152	-0.053	0.039	0.103	0.396	0.07	0.034	0.029	0.085	0.031	0.035
Day	0.008	1	0	0.013	0.016	0.01	0.014	-0.106	0.07	0.005	-0.001	0.011	-0.03	0.012	0.004	0.005	0.004	0.002	0.005
Hour	0	0	1	0.001	0.152	0.186	0.145	0.106	0.012	0.178	0.1	0.34	-0.001	0.328	0.263	0.26	-0.102	0.278	0.274
DOW	0.007	0.013	-0.001	1	-0.01	0.009	-0.01	0	0.043	-0.06	-0.045	0.114	0.008	-0.087	-0.058	-0.052	0.11	-0.027	-0.031
steel-Temp	0.327	0.016	0.152	-0.01	1	0.984	0.998	0.118	0.085	0.13	0.196	0.131	0.001	0.341	0.277	0.258	-0.289	-0.24	0.285
concTemp	0.3	0.01	0.186	-0.009	0.984	1	0.98	0.169	0.097	0.074	0.18	0.111	0.02	0.287	0.236	0.216	-0.271	-0.211	0.246
air-Temp	0.327	0.014	0.145	-0.01	0.998	0.98	1	0.109	0.1	0.131	0.205	0.119	0.043	0.344	0.28	0.261	-0.261	-0.224	0.286
strain-Median	-0.478	0.106	0.106	0	0.118	0.169	0.109	1	0.069	0.011	0.068	0.134	0.16	0.039	-0.02	-0.024	-0.152	-0.072	-0.011
Measurement	0.152	0.07	0.012	0.043	0.085	0.097	0.1	0.069	1	-0.024	0.046	-0.086	0.306	0.125	0.074	0.076	0.083	0.075	0.072
small-Car	0.053	0.005	0.178	-0.06	0.13	0.074	0.131	0.011	-0.024	1	0.269	0.388	-0.096	0.461	0.398	0.391	-0.292	-0.289	0.392
medium-Car	-0.039	0.001	0.1	0.045	0.196	0.18	0.205	0.068	0.046	0.269	1	0.467	0.151	0.241	0.2	0.198	-0.072	-0.099	0.191
large-Car	0.103	0.011	0.34	0.114	0.131	0.111	0.119	0.134	-0.086	0.388	0.467	1	0.246	0.384	0.267	0.26	-0.258	-0.257	0.262
Date	0.396	-0.03	0.001	0.008	-0.001	0.02	0.043	0.16	0.306	-0.096	0.151	-0.246	1	0.032	0.035	0.047	0.324	0.211	0.022
Area	0.07	0.012	0.328	0.087	0.341	0.287	0.344	-0.039	0.125	0.461	0.241	0.384	0.032	1	0.901	0.894	-0.266	-0.38	0.891
strain-Max	0.034	0.004	0.263	-0.058	0.277	0.236	0.28	-0.02	0.074	0.398	0.2	0.267	0.035	0.901	1	0.994	-0.225	-0.282	0.995
strain-Mean	0.029	0.005	0.26	0.052	0.258	0.216	0.261	-0.024	0.076	0.391	0.198	0.26	0.047	0.894	1	0.991	-0.186	-0.254	0.992
strain-Mean-Top	0.085	0.001	0.102	0.11	-0.289	-0.271	-0.261	-0.152	0.083	-0.292	-0.072	-0.258	0.324	0.266	-0.186	1	-0.199	0.643	-0.237
strain-Mean-Bottom	0.031	0.004	0.278	-0.027	0.267	0.23	0.269	-0.011	0.075	0.373	0.183	0.256	0.033	0.881	0.992	0.991	-0.199	1	0.997
strain-Min	0.013	0.002	-0.182	0.092	-0.24	-0.211	-0.224	-0.072	0.003	-0.289	-0.099	-0.257	0.211	-0.38	0.881	-0.254	0.643	1	-0.293
strain-STD	0.035	0.005	0.274	-0.031	0.285	0.246	0.286	-0.011	0.072	0.393	0.191	0.262	0.022	0.891	0.995	0.992	-0.237	-0.293	1

'hour' and 'air temperature', and 'hour' and 'small car traffic' from the direct search algorithm and the correlation method, respectively. This result shows how the selection of predictors is different between the two methods and the direct search method is better than the correlation method.

6.3.4 Prediction of traffic data using bridge sensor data

In the preceding section, the direct search method was investigated to find the best predictor combination for 6 target responses. The same approach is applied to investigate the application of bridge sensor data to the prediction of traffic data. Here, the previous 6 target responses related to strain are considered as predictors and three traffic variables (i.e., traffic of small, medium and large size of car) are treated as target responses. Best predictors for 3 target responses of the traffic data are shown in Figure 6.8. Usually, the more predictors, the higher prediction accuracy, but the highest accuracy is not always guaranteed when using all predictors. The numbers of the best predictor variables for the small, medium and large size of cars turned out to be 15, 13 and 14 among 16 variables. Those selected predictors are listed in Table 6.4.

Figure 6.9 presents comparisons between measured values and GAM prediction results for three traffic groups. The more points adjacent to the red diagonal line (i.e., line of equality), the better prediction performance. The prediction accuracy of three traffic groups does not seem to be significantly high because a number of points are spread out from the red line. However, this result is still noteworthy because, in the case that traffic is not available, this approach enables researchers to estimate traffic from the proposed prediction model using bridge sensor data.

6.4 Remarks on Various Impacts on Prediction Accuracy

6.4.1 Impact of data curing on prediction

Data measured by sensors typically have missing values due to various reasons such as human-made mistakes, measurement errors, malfunctions of sensors, etc. Missing data may result in low accuracy in statistical inference and machine learning prediction. FHDI has been adopted in this study to cure missing values in the hybrid data set. The original dataset has 10% of missing

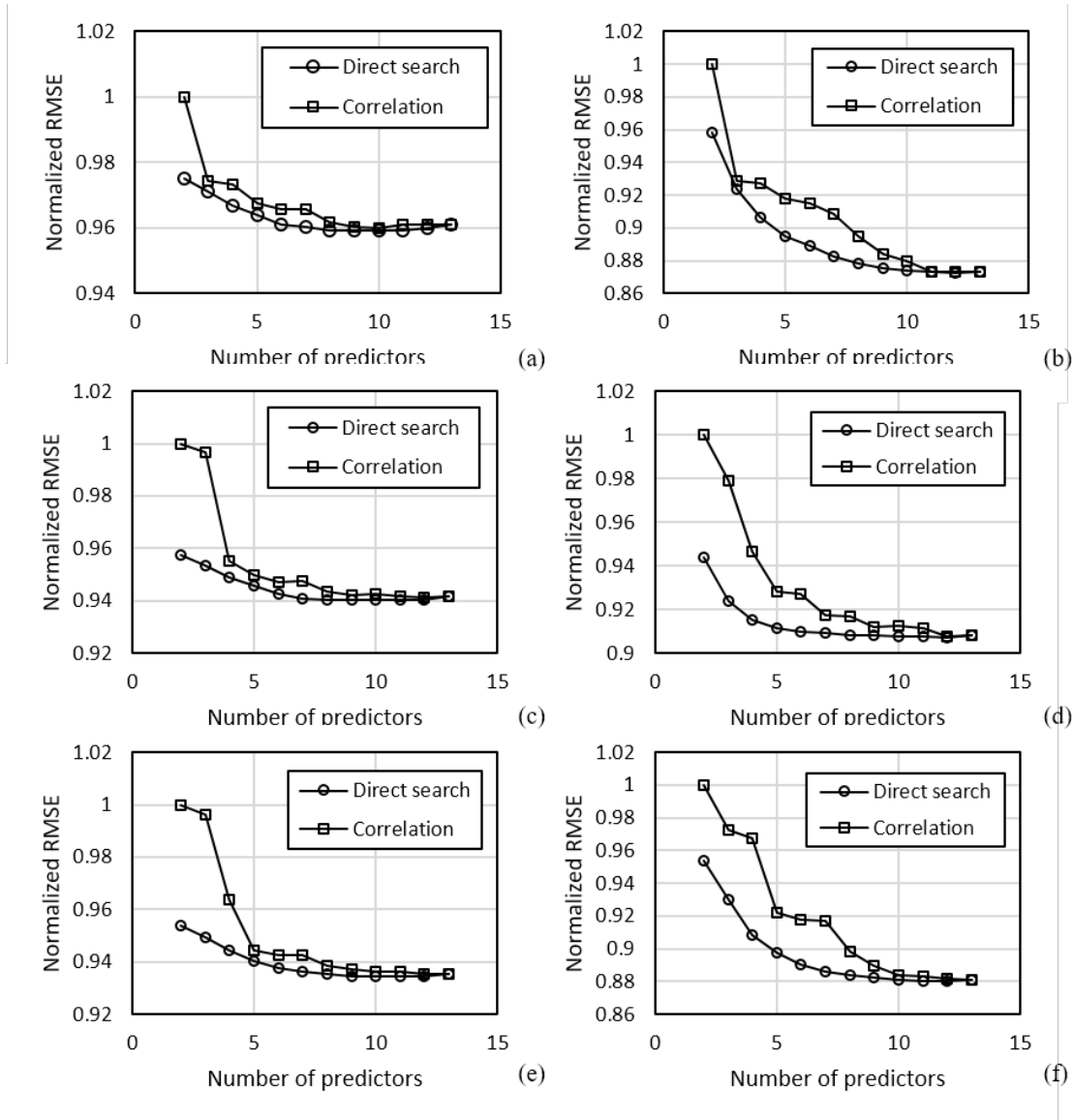


Figure 6.7: The comparison of the best predictor selection between the algorithm used in this study and correlation: (a) mean of top peak strains; (b) mean of bottom peak strains; (c) standard deviation of median strain; (d) minimum strain value of bottom peak; (e) maximum strain value of top peak; (f) area

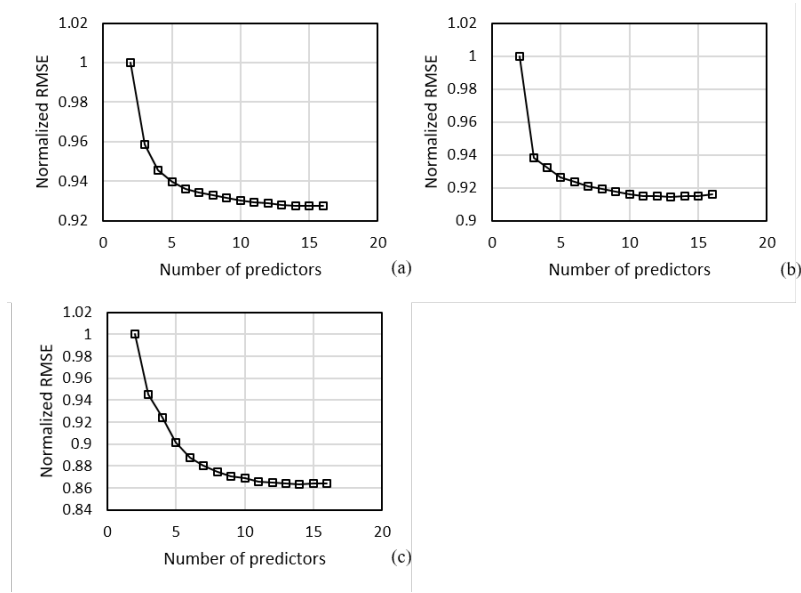


Figure 6.8: The number of the best predictors of traffic data prediction: traffic of (a) small car, (b) medium car and (c) large car

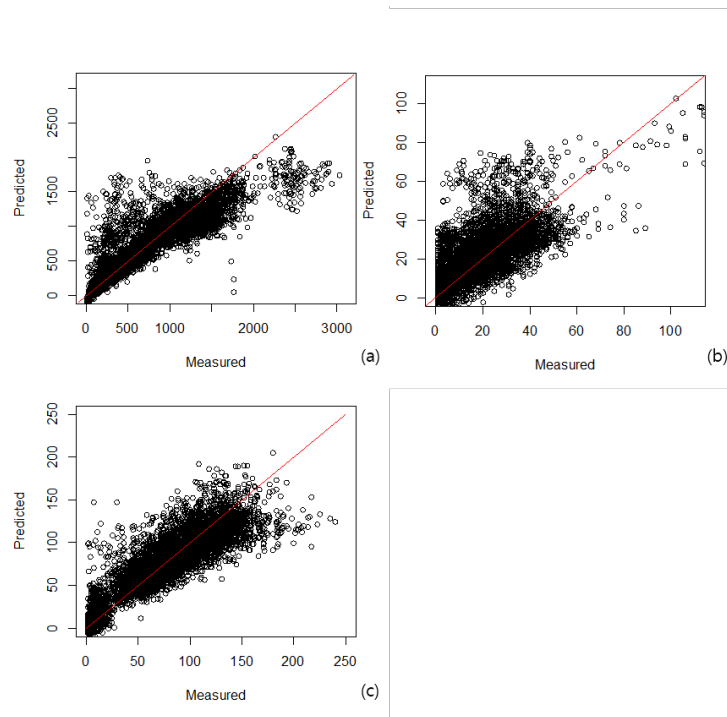


Figure 6.9: GAM prediction vs. measured value of traffic: (a) small car, (b) medium car and (c) large car

Table 6.4: Best predictors selected by the direct search method

Prediction target	# of variable	Predictor (p-value)		
strainMean Top	10	Month(4.91e-9) airTemp(4.80e-7) smallCar(9.15e-11) Date(2.26e-11)	Hour(<2e-16) strainMedian(4.22e-5) mediumCar(0.106)	concTemp(1.09e-6) nMeasurement(<2e-16) largeCar(3.24e-15)
strainMean Bottom	12	Month(<2e-16) DOW(<2e-16) airTemp(<2e-16) smallCar(2.63e-9)	Day(0.02626) steelTemp(<2e-16) strainMedian(<2e-16) mediumCar(0.00224)	Hour(<2e-16) concTemp(3.06e-12) nMeasurement(<2e-16) Date(<2e-16)
strainSTD	10	Month(4.32e-9) airTemp(2.89e-7) smallCar(3.41e-13) Date(2.39e-10)	Hour(<2e-16) strainMedian(2.92e-5) mediumCar(0.191)	concTemp(3.05e-7) nMeasurement(<2e-16) largeCar(9.10e-12)
strainMax	11	Month(5.52e-11) concTemp(1.39e-6) nMeasurement(<2e-16) largeCar(2.34e-10)	Hour(<2e-16) airTemp(2.49e-6) smallCar(8.14e-10) Date(3.78e-10)	DOW(9.81e-15) strainMedian(2.46e-5) mediumCar(0.27)
strainMin	12	Month(5.42e-6) DOW(<2e-16) airTemp(8.12e-7) mediumCar(0.025373)	Day(0.364342) steelTemp(1.22e-12) nMeasurement(<2e-16) largeCar(0.007920)	Hour(<2e-16) concTemp(0.000649) smallCar(0.072332) Date(<2e-16)
area	12	Month(<2e-16) DOW(<2e-16) strainMedian(3.73e-10) mediumCar(0.00458)	Day(5.05e-4) concTemp(<2e-16) nMeasurement(<2e-16) largeCar(<2e-16)	Hour(<2e-16) airTemp(2.41e-15) smallCar(1.06e-8) Date(6.98e-13)
small car traffic	15	Month(<2e-16) DOW(<2e-16) airTemp(8.95e-7) Area(5.41e-13) strainMeanTop(2.50e-15)	Day(4.46e-13) steelTemp(9.69e-7) strainMedian(4.19e-4) strainMax(1.27e-4) strainMin(1.96e-7)	Hour(<2e-16) concTemp(3.75e-5) Date(<2e-16) strainMeanBottom(1.04e-4) strainSTD(4.71e-16)
medium car traffic	13	Month(<2e-16) DOW(<2e-16) airTemp(<2e-16) Date(<2e-16) strainSTD(7.70e-8)	Day(<2e-16) steelTemp(3.17e-12) strainMedian(<2e-16) Area(9.07e-7)	Hour(<2e-16) concTemp(2.39e-12) nMeasurement(0.2495) strainMax(0.0122)
large car traffic	14	Month(<2e-16) DOW(<2e-16) airTemp(2.46e-14) Date(<2e-16) strainMin(0.78)	Day(<2e-16) steelTemp(3.40e-9) strainMedian (<2e-16) Area(<2e-16) strainSTD(1.62e-12)	Hour(<2e-16) concTemp(1.14e-7) nMeasurement(<2e-16) strainMeanTop (6.24e-7)

values. The six target responses in Table 2 are predicted using datasets with and without data curing, respectively, and their prediction performances are compared to investigate the impact of imputation on prediction. Figure 6.10 shows the comparison result. RMSE values, normalized by the values using imputed datasets, are used as the performance metric. The prediction errors are decreased for all 6 cases when using the imputed dataset. Although the amount of prediction accuracy improvement is not significantly large, the improvement is confirmed for all 6 target responses (Figure 6.9). FHDI cures missing values only using observed values and tries to preserve the joint probability of all variables in the original population (Im et al., 2015). In light of the underlying theory of FHDI, the data structure and predictor-target variables' relation may affect how much data-curing improves the prediction accuracy. For instance, data-curing may significantly improve the data-prediction when a dataset has a high missing rate. This study performed another case study with a dataset which has 9,357 samples in 13 variables. We intentionally made three datasets that have different missing rates and then cured the missing values using FHDI. A target response was predicted using GAM. Figure 6.11 shows the case when the data-curing holds a substantial impact on data-prediction. In Figure 6.10, we can confirm the higher missing rates, the larger RMSE.

6.4.2 Impact of traffic information on prediction performance

Another prediction analysis is conducted to see the impact of traffic data on prediction performance. The same target responses are predicted using the datasets with and without traffic information, respectively. The effect of inclusion of traffic data on prediction is investigated by comparing prediction performances. Figure 6.12 presents the comparison result, in which the RMSE values are normalized by the values obtained from prediction using the dataset including traffic. Once again, the lower RMSE indicates the better prediction performance. The result shows the inclusion of traffic data invariably improves the prediction performance for all 6 cases. Although the accuracy improvements may not look significant in the current dataset, the inclusion of traffic data apparently holds positive influence on the data-prediction. This means that for the

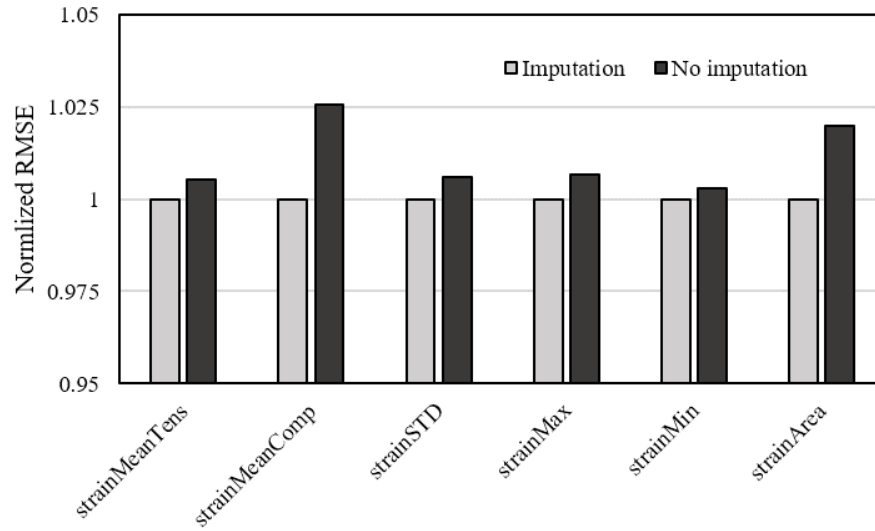


Figure 6.10: Comparison of GAM prediction performances using the dataset with and without imputation

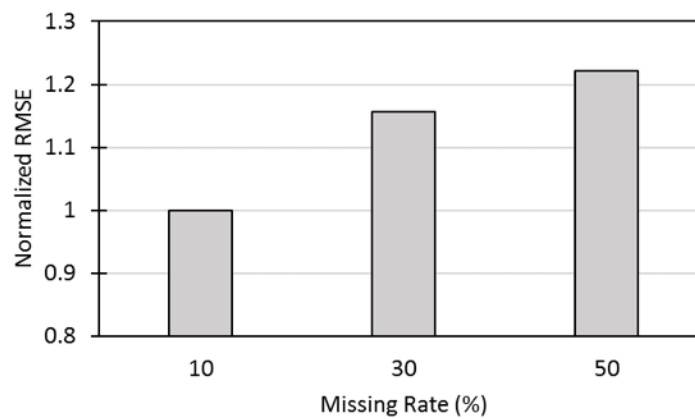


Figure 6.11: Impact of missing rates on prediction accuracy (cited from Song et al. (2018a))

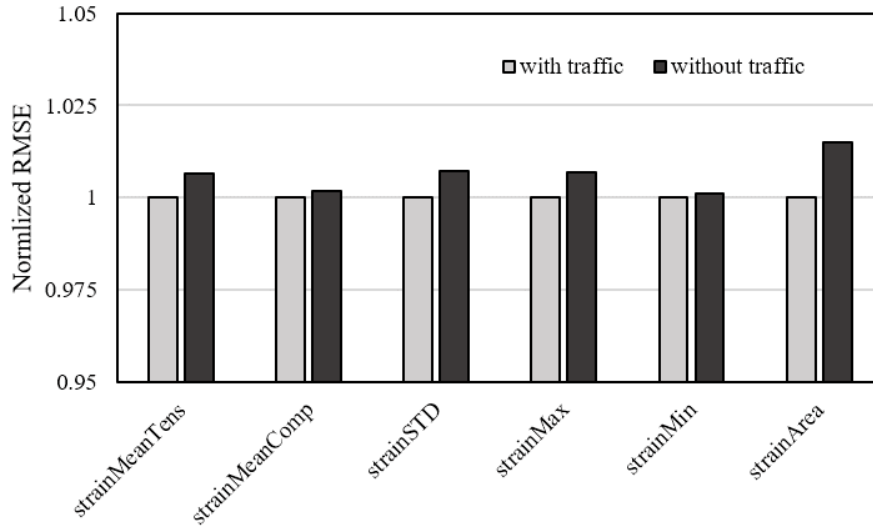


Figure 6.12: Comparison of GAM prediction performances using the dataset with and without traffic data

current dataset the traffic data provide additional meaningful information to bridge sensor data, underpinning the consistent merge of bridge and traffic big data over a longer time period.

Figure 6.13 shows the influence of the inclusion of traffic data on prediction performance depending on different missing rates, in which the target response is *strainMeanComp*. In the case of 40% of missing rate, the normalized RMSE is about 1.06 while that is about 1.00 in the case of 10%, which indicates the inclusion of traffic data has a high impact on the prediction performance in a dataset with a high missing rate.

6.5 Parallelization Strategy

Figure 6.14 shows the job distribution and collection scheme for the parallel computing for the best predictor selection. The master processor only manages whole computing processes (i.e., distributes searching tasks to slave processors and collect the searching results from them).

Slave processors build multiple GAM models using their assigned predictor combinations, predict the target responses, calculate the prediction accuracies using RMSE, and return the RMSE values and the corresponding predictor combinations to the master processor. Finally, the master

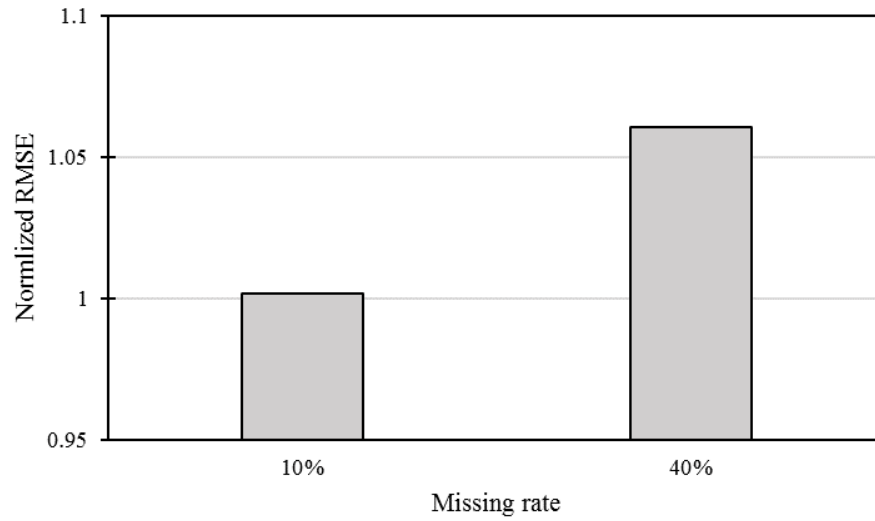


Figure 6.13: The impact of the inclusion of traffic data on prediction of the *strainMeanComp* depending on different missing rates

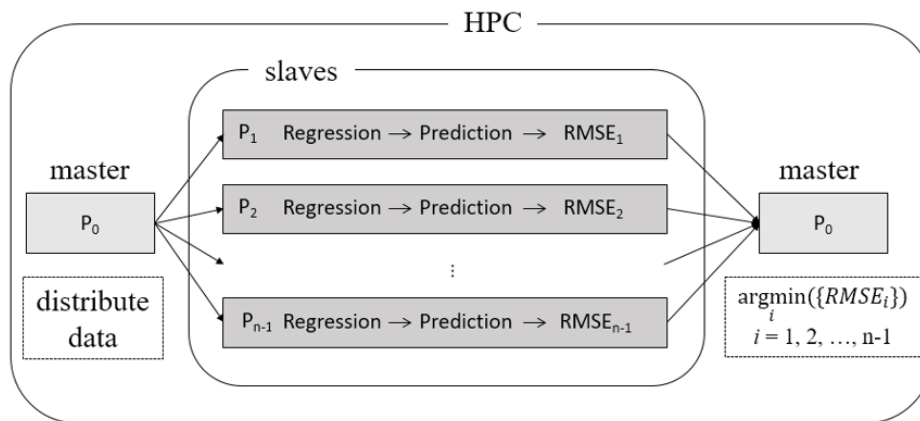


Figure 6.14: Job distribution scheme in the parallel computing system. Jobs are evenly distributed to slaves and then the master collects results from slaves and finds the best predictors

Pseudo code: Algorithm for the parallel work to find the best predictors		
Input: dataR and dataV, datasets for regression and validation		
Output: $(m, i)^*$, the best predictor combination		
	Code	Description
1	[on P ₀]	- P ₀ : master processor
2	Spawn n slaves	- spawn n slave processors
3	[on P ₁ ~P _{$n-1$}]	- P _{i} : slave processor, $i=1, 2, \dots, n-1$
4	import dataR , dataV	- dataR : dataset for regression - dataV : dataset for validation
5	$m = \text{MPI_Comm_rank}()$	- m : id of slave processor
6	$n = \text{MPI_Comm_size}()$	- n : total number of processors
7	$n_combi = n_combi_all / (n-1)$	- n_combi : number of predictor combinations assigned to a slave processor - n_combi_all : total number of predictor combinations
8	for $i=1$ to n_combi	
9	$F_i^m = \text{GAM}(\mathbf{dataR}_i)$	- F_i^m : fitted model using GAM, $m=1, 2, \dots, n-1$
10	$[\mathbf{P}]_i^m = \text{GAM.PRED}(F_i^m, \mathbf{dataV}_i)$	- $[\mathbf{P}]_i^m$: predicted response
11	$\mathbf{M}_i^m = \text{METRIC}([\mathbf{P}]_i^m)$	- \mathbf{M}_i^m : prediction performance metrics
12	end for	
13	Send \mathbf{M}_i^m to P ₀	
14	[on P ₀]	
15	Receive \mathbf{M}_i^m from P ₁ ~P _{$n-1$}	
16	$(m, i)^* = \underset{m, i}{\text{argmin}}(\mathbf{M}_i^m)$	- $(m, i)^*$: the best predictor combination

Figure 6.15: Pseudo code for algorithm of the parallel computing to find the best predictor combination

processor selects the best predictor combination based on the collected results from slaves. The pseudo code of this parallel computing procedure is shown in Figure 6.15. Figure 6.16 shows a speed-up test result in which T_n/T_1 represents the ratio of running time using n slave processors to that using 1 processor. The parallel computing appears to achieve a reasonable scalability.

6.6 Conclusions

In order to promote the active use of bridge and traffic big data for long-term decision-making and strategic planning, this study developed a computational framework that is capable of tackling severe complexity, high dimensionality, missing data problems, and the lack of powerful learning

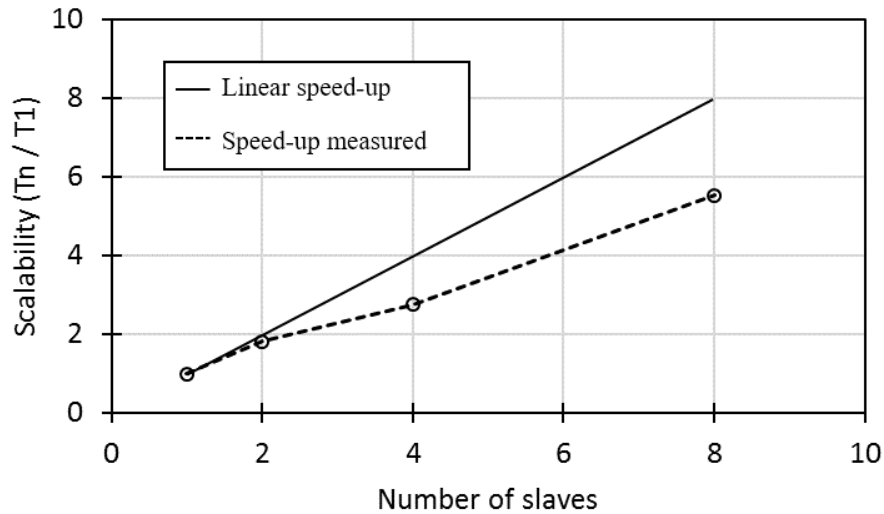


Figure 6.16: Speed-up test using parallel computing

and prediction methods. The developed framework can transform, merge, and squash bridge and traffic big data to improve data-learning and prediction process. The framework adopted a direct search algorithm for a superior predictive power and the fractional hot-deck imputation method for data curing. The framework created a hybrid big data by merging bridge and traffic data and parallel computing algorithms were implemented for scalability and expandability. By using three-year strains and traffic data collected from a target bridge, this study asserts that the direct search algorithm appears to outperform the correlation-based approach in model selection and data prediction. Also, results underpin that data curing and the hybrid big data appear to hold positive impact on improving statistical learning quality and prediction accuracy. All the developed programs will be made publicly available to maximize broader impacts of research community.

6.7 Acknowledgments

This research is jointly supported by the Iowa Department of Transportation, Midwest Transportation Center, U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology, the research funding of the Department of Civil, Construction, and Environmental

Engineering of Iowa State University, Black & Veatch, and NSF CBET 1605275. Special thanks are due to Dr. Raymond Wong, Dr. Jaekwang Kim and Dr. Jongho Im for the guidance on the statistical theories. The parallel computing reported herein is partially supported by the HPC@ISU equipment at ISU, some of which has been purchased through funding provided by NSF under MRI Grant Nos. CNS 1229081 and CRI 1205413.

Bibliography

- Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223.
- Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC Press, Boca Raton, FL, USA.
- Im, J., Cho, I., and Kim, J. (2018). *FHDI: Fractional Hot Deck and Fully Efficient Fractional Imputation*. R package version 1.2.2.
- Im, J., Kim, J.-K., and Fuller, W. A. (2015). Two-phase sampling approach to fractional hot deck imputation. In *Proceedings of the Survey Research Methods Section*, pages 1030–1043.
- Jang, S., Jo, H., Cho, S., Mechitov, K., Rice, J. A., Sim, S.-H., Jung, H.-J., Yun, C. B., Spencer Jr, B. F., and Agha, G. (2010). Structural health monitoring of a cable-stayed bridge using smart sensor technology: deployment and evaluation. *Smart Structures and Systems*, 6(5-6):439–459.
- Kim, J. K. and Fuller, W. (2004). Fractional hot deck imputation. *Biometrika*, 91(3):559–578.
- Ko, J. and Ni, Y. (2005). Technology developments in structural health monitoring of large-scale bridges. *Engineering structures*, 27(12):1715–1725.
- Le, T. and Jeong, H. (2017). Nlp-based approach to semantic classification of heterogeneous transportation asset data terminology. *Journal of Computing in Civil Engineering*, 31(6):04017057.
- Li, H.-N., Li, D.-S., and Song, G.-B. (2004). Recent applications of fiber optic sensors to health monitoring in civil engineering. *Engineering structures*, 26(11):1647–1657.
- Li, Z., Chan, T. H., and Zheng, R. (2003). Statistical analysis of online strain response and its application in fatigue assessment of a long-span steel bridge. *Engineering structures*, 25(14):1731–1741.
- Lv, Y., Duan, Y., Kang, W., Li, Z., and Wang, F.-Y. (2015). Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873.
- Ntotsios, E., Papadimitriou, C., Panetsos, P., Karaiskos, G., Perros, K., and Perdikaris, P. C. (2009). Bridge health monitoring system based on vibration measurements. *Bulletin of Earthquake Engineering*, 7(2):469.
- Perera, L. P. and Mo, B. (2016). Data compression of ship performance and navigation information under deep learning. In *ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering*, pages V007T06A086–V007T06A086. American Society of Mechanical Engineers.
- Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. John Wiley & Sons.

- Song, I., Cho, I., Im, J., and Kim, J. (2018a). Impacts of fractional hot-deck imputation on machine learning of engineering data. *in preparation*.
- Song, I., Cho, I., and Tessitore, T. (2017). Advanced statistical learning and prediction of complex runway incursion. In *Airfield and Highway Pavements 2017*, pages 38–50.
- Song, I., Cho, I., Tessitore, T., Gurcsik, T., and Ceylan, H. (2018b). Data-driven prediction of runway incursions with uncertainty quantification. *Journal of Computing in Civil Engineering*, 32(2):04018004.
- Song, I., Cho, I. H., and Wong, R. K. W. (2018c). An advanced statistical approach to data-driven earthquake engineering. *Journal of Earthquake Engineering*, 0(0):1–25.
- Wood, S. (2006). *Generalized additive models: an introduction with R*. CRC Press.
- Yu, H. (2002). Rmpi: parallel statistical computing in r. *R News*, 2(2):10–14.

CHAPTER 7. CONCLUSIONS

This dissertation focuses on developing a systematic computational framework for infrastructure databases using advanced statistical methods, such as the generalized additive model (GAM) and fractional hot-deck imputation(FHDI), and parallel computing technology. The GAM is a flexible non-linear statistical model due to its unspecified smooth function in which covariates enter the model without any prejudice or assumption of variables. All manuscripts provided herein used this novel statistical model for accurate prediction. The speed-up test results demonstrate the parallel computing is useful for a large amount of data and remarkably reduces the computing time. FHDI shows its reliable ability of prediction accuracy improvement. The following paragraphs discuss the detailed findings of the studies.

The first study developed a computational framework to utilize GAM to accurately predict runway incursion (RI) in the major US airports. Relevant information, such as the geometric information of airports, operational data, and visibility data were collected from heterogeneous databases in the Federal Aviation Administration. The data collected are transformed into a concise dataset for data analysis. Using the GAM with a direct search (DS) algorithm, the best predictor variables were identified for RI prediction. Results show that all variables are not always necessary for accurate prediction and five variables were selected: (1) the number of taxi operations, (2) the number of general operations, (3) hours of high impact visibility, (4) hours of slight impact visibility, and (5) sum of hours of high, moderate, and slight impact visibility. The principal component analysis (PCA) method was compared to the DS and it turned out that DS outperforms the PCA-based variable selection. The comparison between GAM and ANN also illustrates the superior prediction power of GAM. This study reveals the clear causal pathway between salient variables and provides the relevant importance of predictor variables, which will help stakeholders to arrive at a practical decision.

The second study expounded upon GAM that can facilitate a data-driven approach in the earthquake engineering field. Particularly, reinforced concrete (RC) shear wall data are used for statistical learning and prediction. The important variables are selected by using the DS method. Validations to real-world earthquake engineering data exhibit a promising capability of GAM. The prediction performances of GAM are compared to the high-precision simulation results. Results show that the statistical prediction holds a reasonable level of accuracy. In terms of running times, the statistical approach appears to be superior to the simulation approach.

The third study investigated efficient variable selection methods and identified the relative importance of predictor variables for GAM prediction using field survey pavement and simulated airport pavement data. The direct search method can find the best predictor variables, but it takes a long time depending on the size of data and number of predictor variables. However, the backward selection based on AIC can provide acceptable prediction accuracy with a much smaller amount of time than that of the direct search approach. Age, thickness, joint spacing, and overlay type variables turn out to be relatively significant for GAM prediction in the field survey data, and variables of thickness and modulus of pavement turn out to play an important role in the simulated airport pavement data. The impact of family distribution on GAM prediction was also investigated. The results show that Gamma distribution appears to be reliable for most cases.

The fourth study examined the impact of FHDI on statistical learning and ML regressions using four engineering databases. To this end, the different response rates from 10% to 50% and a wide range of FHDI's two parameters have been examined. Multiple regression methods are adopted including GAM, SVM, ERT, and ANN to investigate the quantitative impacts of FHDI on the regression prediction. Normalized RMSE is used to measure the prediction accuracy of each case. Results show that FHDI outperforms a simple naive method in terms of prediction accuracy improvement. According to the parametric study, it turned out k of 30 or 35 and M of 5 are optimal parameter setting for FHDI implementation using general engineering data.

The fifth study developed a systematic computational framework for collecting, transforming, and squashing bridge sensor and traffic big data. Advanced statistical methods, the FHDI and

GAM, are adopted for seamless data curing and accurate prediction. Three-year strain, temperature data, measured by sensors installed in the target bridge, are used to predict the bridge's structural behaviors. Results show that the direct search method is superior to the correlation-based variable selection approach, and that the hybrid data, combining bridge and traffic data, hold a positive impact on the prediction accuracy improvement.

By virtue of the "additive" nature of GAM, the prediction accuracy will be able to be improved as community-level databases continue to evolve, which will enable researchers and stakeholders to better understand the underlying relationship among variables in databases and devise a better decision based on the improved results. The investigation of more sophisticated methods for efficient variable selection can be a topic for further research. The current version of FHDI program is serial. For a small dataset with a few variables, this program can complete the implementation in a reasonable time; however, this time will exponentially increase as the data size become bigger and the dimensions of variables increases. Hence, the development of a parallel version of FHDI can be the next research topic.