Multimodal Machine Learning for Climate Adaptation

by

Cynthia Zeng

BSc Mathematics, Imperial College London, 2017

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by:	Cynthia Zeng Department of Operations Research May 1, 2024
Certified by:	Dimitris J. Bertsimas Associate Dean for the Master of Business Analytics Professor of Operations Research, Thesis Supervisor
Accepted by:	Georgia Perakis John C Head III Dean (Interim), MIT Sloan School of Management Professor of Operations Management, Operations Research & Statistics

Multimodal Machine Learning for Climate Adaptation

by

Cynthia Zeng

Submitted to the Sloan School of Management
on May 1, 2024 in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

ABSTRACT

Climate change stands as one of the most urgent challenge of our generation. Devastating floods in Pakistan, heartbreaking earthquakes in Turkey, and unprecedented wildfires in Canada. For over a century, meteorology has traditionally relied on solving dynamical equations; however, machine learning (ML) is now emerging as a transformative force. This thesis explores how to use machine learning and optimization methods to address issues around climate change adaptation and sustainable development.

The first part of the thesis centers around the development of multimodal machine learning frameworks for extreme weather forecasting. The multimodal ML approach integrates diverse sources and modalities of data, including text-based language, images, tabular time series, etc. We showcase the effectiveness of such an approach through two distinct case studies in extreme weather forecasting: in Chapter 2, a short-term hurricane forecast with a 12-hour lead time, and in Chapter 3, a long-term flood risk assessment model. Our contribution include the development of a generalizable multimodal ML framework to facilitate a wide range of prediction tasks in the field of meteorology and beyond. Notably, our hurricane forecasting models demonstrate performance comparable to the National Hurricane Center's top models for 24-hour intensity and track forecasts.

ML-driven weather forecasting models offer two distinct advantages over traditional dynamical models: significant reductions in computational time thus enabling real-time, location-specific predictions, and the ability to develop long-term risk models for proactive disaster mitigation rather than reactive responses. Therefore, in the second part of the thesis, we delve into two application domains to envision the transformative force in addressing climate change induced challenges. In Chapter 4, we introduce an Adaptive Robust Optimization (ARO) framework for designing insurance policies, combining historical and anticipatory risks obtained by machine learning models. In Chapter 5, we develop a real-time machine learning framework for wind forecasting, aimed at adjusting factory production levels to minimize air pollution and its impact on surrounding urban areas. In partnership with OCP Group, the world's largest phosphate producer, our algorithm is now fully integrated into operational systems and reduces hazardous emission impact by 33-47% annually.

Thesis supervisor: Dimitris J. Bertsimas
Title: Associate Dean for the Master of Business Analytics
Professor of Operations Research

Acknowledgments

First and foremost, words cannot fully express my gratitude towards my advisor, Professor Dimitris Bertsimas. I vividly remember our first meeting in your office, an ordinary October afternoon in 2018, which has forever altered the course of my life. You have enlightened me that mathematics can make our world a better place. I am incredibly fortunate to have you as my advisor; you have taught me how to be a researcher, a team player, and most importantly, a compassionate human being.

I am also immensely grateful to my thesis committee members, Professor Jónas Jónasson and Professor Nicholas Trichakis, for your invaluable feedback and guidance throughout my PhD journey. I am deeply humbled by your achievements within and beyond academia, and I would be extremely fortunate to achieve even a fraction of the meaningful impact you have made through your research.

Special gratitude goes to my academic mentors who have invested in my success during my PhD journey: Professor Thomas Magnanti, Professor Bart van Parys, Professor Daniel Kuhn, Professor Vivek Farias. I am thankful to my undergraduate advisor, Professor Gunnar Pruessner from Imperial College, who believed in me from day one and encouraged me to embark on this wonderful academic journey. Additionally, I must thank my mentor and former manager at BlackRock, Ursula Gritsch, for encouraging me to pursue a doctoral degree during my time in industry.

I extend my gratitude to Professor Georgia Perakis and Professor Patrick Jaillet, the co-directors of the Operations Research Center (ORC) for fostering such a welcoming and dynamic environment. My appreciation also goes to Laura Rose and Andrew Carvalho for their unwavering support of us students at the ORC.

The ORC is a special place where collaboration and mentorship often blossom into friendships. I am extremely fortunate to have worked with and learned from incredible collaborators - Léonard Boussioux, Kimberley Villalobos, Irra Na, Yu Ma, Holly Wiberg, Michael Li, Louis Soenksen, and my industry collaborators at the OCP Group - Karim Bendioub, Yassine Azhari, Aimad Aneddame, and Walid Daou, who have been instrumental in deploying our models. To the wonderful Master's and undergraduate students I have had the opportunity to mentor and work with: Tiana Cui, Calvin Wang, Peggy Wang, Oscar Zhang, and Riley Oh. To my ORC PhD cohort — Kim, Léonard, Shuvo, Amine, Moise, Leann, Adam, Ken and Kayhan — thank you for sharing this incredible PhD journey together. To the wonderful friends I have made at the ORC — Célia, Agni, Holly, Lea, Ivan, Tamar, Jean, Vassilis, Driss, Ryan, Raluca, Tianyi, and the Grand Canyon crew, Manuel, Rares, Asterios — I cherish our continued friendship going forward.

Acknowledgments are also due to my MIT and Harvard friends who have made Cambridge a truly special place: Adriana, Manon, Catherine, Hallee, Margarita, Delaney, Marianne, Amélie, Annan, Pete, Oscar, Farri, Francesco, Beth, Lucio, Pierre, Dimitra, Carmen, Nicholas, Yaroslav, Yves. To friends I have known from Imperial College London - Catherine, Monika, Merissa, Hlér, Balint, Callum, Giovanni, Alwin, Wendelin and Sofie, Borislav, Hubert. The fun crowd in St Maxime - Sofia, Emmanuel, Guido, Philipp. Of course, my childhood friends who have shared my journey moving from Beijing to London to Boston - Audrey, Kylie and Annabel. These friendships are treasures I will always hold dear. To the Charles river, the sunsets, and Rachmaninoff's piano concertos for accompanying me during those lonely COVID days.

I am also extremely grateful to my mentors and family friends - Anke and Brian Ma Siy, Sir George Iacobescu, Professor Yi Cui, Peggy Dulany, Elliott Donelley, Susanne Robstock, Mark Wiseman, Dr. He, Hannnelore Tschiderer, Professor David Pan. You have inspired me through true artisanship and professionalism. Thank you for enlightening conversations and for standing next to me at some major crossroads in my life.

Finally, I must express my deepest gratitude to my family, to whom I dedicate this thesis and attribute every single accomplishment to. To my mother, stepfather, father, stepmother, grandmother, my grandparents who are no longer with us. For your endless love and support throughout my life, and for the sacrifices made to pave the foundations upon which your children hope to thrive. To my sister Helena and my brother Gordon, I am extremely grateful and fortunate to share this journey called life with you. I hope we continue to serve as the source of strength, courage and inspiration for one another. To our dog Afu, you are an important member of the family.

This thesis is partially supported by the OCP grant “Machine learning for weather forecasting at OCP Safi Site under Grant No. 6941242. Any opinions, findings , conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of OCP.

Contents

Title page **1**

Abstract **3**

Acknowledgments **5**

List of Figures **13**

List of Tables **17**

1 Introduction **21**

1.1 Organization 22

1.1.1 Multimodal Machine Learning 22

1.1.2 Applications around Climate Change Adaptation 24

1.2 Structure and Contributions 24

2 Hurricane Forecasting: A Novel Multimodal Machine Learning Framework **27**

2.1 Introduction 28

2.2 Data 30

2.2.1 Historical Storm Data Set 30

2.2.2 Reanalysis Maps 31

2.2.3 Operational Forecast Models 33

2.3 Methodology 35

2.3.1 Feature Extraction 36

2.3.2 Forecasting Models 39

2.3.3 Summary of Models 40

2.4 Experiments 41

2.4.1 Evaluation Metrics 41

2.4.2 Training, Validation and Testing Protocol 41

2.4.3 Computational Resources 42

2.5 Results 42

2.5.1 Standalone machine learning models produce a comparable performance to operational models. 42

2.5.2 Machine learning models bring additional insights to consensus models. 44

2.5.3 A multimodal approach leads to more accurate forecasts than using single data sources.
. . . . 45

2.6 Limitations and Extensions 46

2.6.1 The Use of Reanalysis Maps 46

2.6.2 Incorporate Additional Data 46

2.6.3 Longer-Term Forecasts 46

2.7 Conclusion 47

**3 Global Flood Prediction:
A Multimodal Machine Learning Approach 49**

3.1 Introduction 49

3.2 Data 51

3.3 Methodology 52

3.3.1 Statistical Feature Processing 53

3.3.2 Text Feature Processing 54

3.4 Training and Testing Protocol 55

3.5 Results 56

3.6 Conclusion 57

3.6.1 Future Work 58

3.7 Acknowledgements 59

4 Catastrophe Insurance Pricing:
An Adaptive Robust Optimization Approach 61

4.1 Introduction 62

4.2 Literature Review 64

4.3 Optimization Framework for Catastrophe Insurance 66

4.3.1 Robust Optimization 66

4.3.2 Nominal Formulation 66

4.3.3 Robust Optimization Formulation 67

4.3.4 Choices of demand damping function f(p) 74

4.3.5 Adaptive Robust Optimization Formulation 74

4.4 Case Study for US National Flood Insurance 81

4.4.1 Data 81

4.4.2 Optimization Model Parameter Estimation 82

4.4.3 Machine Learning Model 83

4.5 Results 84

4.5.1 Sensitivity Analysis 86

4.5.2 Effectiveness 87

4.5.3 Efficiency 89

4.6 Conclusion 91

5 Reducing Air Pollution through Machine Learning 95

5.1 Introduction 96

5.2 Methodology 98

5.2.1 Case Study on Safi 98

5.2.2 The Previous Operational Procedure at Safi 100

5.2.3 Scenario Definition 101

5.2.4 Predictive Methodology 102

5.2.5 Prescriptive Methodology 105

5.2.6 Training Protocol 106

5.3 Results 107

5.3.1 Predictive Methodology Results 107

5.3.2 Prescriptive Methodology Results 109

5.4 Conclusion 112

6 Conclusion **115**

A Appendix for Chapter 2 **117**

A.1 Encoder-Decoder Architectures 117

A.1.1 Overall Architecture and Mechanisms 117

A.1.2 Technical Details on the CNN-Encoder GRU-Decoder Network 118

A.1.3 Technical Details on the Transformer-Decoder Architecture 120

A.2 Tucker Decomposition for Tensors 121

A.3 Experiment Details 123

A.3.1 Testing Methodology 123

A.3.2 The Specific Protocol for HUML-ensemble 123

A.3.3 Hyperparameter Tuning 124

A.3.4 Metrics 126

B Appendix for Chapter 4 **127**

B.1 Optimization Model Parameter Estimation 127

B.1.1 Computing L_i^{CLT} 127

B.1.2 Demand Damping 128

B.2 Training Protocol of Machine Learning Models 129

B.2.1 Data Processing 129

B.2.2 Training and Testing Protocols 130

B.2.3 Detailed Prediction Results 130

B.2.4 Computational resources 131

References **133**

List of Figures

1.1 Illustration of blind men touching an elephant. 23

2.1 Representation of the nine reanalysis maps repeatedly extracted for each time step, corresponding to three different features (geopotential height z, zonal component of the wind u, meridional component of the wind v) at three pressure levels (225, 500, 700 hPa). Each map is of size $25° \times 25°$, centered on the TC center location, and each pixel corresponds to the average field value at the given latitude and longitude degree. 34

2.2 Representation of our multimodal machine learning framework using the two data sources: statistical and reanalysis maps. During Step 1, we extract embeddings from the reanalysis maps. In particular, we use encoder-decoder architectures or tensor decomposition to obtain a one-dimensional representation. During Step 2, we concatenate the statistical data with the features extracted from the reanalysis maps. During Step 3, we train one XGBoost model for each of the prediction tasks: intensity in 24 h, latitude displacement in 24 h, and longitude displacement in 24 h. 36

2.3 Schematic of our CNN-encoder GRU-decoder network for an 8-time step TC sequence. At each time step, we utilize the CNN to produce a one-dimensional representation of the reanalysis maps. Then, we concatenate these embeddings with the corresponding statistical features to create a sequence of inputs fed sequentially to the GRU. At each time step, the GRU outputs a hidden state passed to the next time step. Finally, we concatenate all the successive hidden states and pass them through three fully connected layers to predict intensity or track with a 24-hour lead time. We finally extract our spatial-temporal embeddings as the output of the second fully connected layer. 38

2.4 Schematic of our CNN-encoder Transformer-decoder network for an 8-time step TC sequence. At each time step, we utilize the CNN to produce a one-dimensional representation of the reanalysis maps. Then, we concatenate these embeddings with the corresponding statistical features to create a sequence of inputs fed as a whole to the Transformer. The Transformer outputs a new 8-timestep sequence that we average (pool) feature-wise and then feed into one fully connected layer to predict intensity or track with a 24-hour lead time. We finally extract our spatial-temporal embeddings as the output of the pooling layer. 39

3.1 Example ‘Geography’ section of the Boston Wikipedia page. 52

3.2 Three-step framework to combine statistical data with text-based data. The transformer-based text data embedding extraction contains three types of architectures. 53

4.1 Number of major disasters globally since 1900, maintained by the EM-DAT database [63]. A disaster is defined as an event which overwhelms local capacity, necessitating a request to the national or international level for external assistance. Disasters include: flood, storm, earthquake, drought, landslide, extreme temperature, wildfire, volcanic activity, mass movement (dry), glacial lake outburst, fog, etc. 62

4.2 Surplus (or loss) computed during 2012 to 2022 across all states when vary different levels of γ_2. Two dotted lines demonstrate the level of surplus from two baseline models: historical surplus calculated from the actual premiums collected, cma surplus is computed using the cumulative moving average. . . 88

4.3 Scatter plot visualizing the efficient frontier, showing how different values of γ_2 affect the number of insolvent states (x-axis) and the total surplus (or deficit) (y-axis) computed as the total premium charged minus actual loss over the testing period. Note that CMA and Hist are plotted as static points because their values do not change with varying γ_2 values. 91

4.4 Scatter plot visualizing the efficient frontier, showing how different values of γ_2 affect the number of insolvent states (x-axis) and the total absolute deviation (y-axis) computed as the absolute value of the difference between premium charged and actual losses over the testing period. Note that CMA and Hist are plotted as static points because their values do not change with varying γ_2 values. The plot demonstrates that ARO achieves higher efficiency with lower absolute deviation at high γ_2 values, highlighting its better performance under these conditions. 92

5.1 Predictive and prescriptive approach to plant operations scheduling. 99

5.2 Predictive and prescriptive approach to plant operations scheduling. 101

5.3 Wind direction and wind speed determine the dissemination of pollutants. Winds coming from the South with low speeds are the most dangerous conditions. 103

5.4 Optimal policy trees trained using a health cost of \$10,000 (left) and \$15,000 (right). The trees illustrate an interpretable decision-making process to arrive at certain recommended decisions (prescription options). `Prescribe x1` corresponds to maintaining production rate while `Prescribe x2` corresponds to reducing plant operations and injecting odor control chemicals. 111

5.5 Screenshot of the software platform used by the plant operators in Safi. Our model predictions for the next 3 hours are displayed on the upper left, while the baseline model is displayed on the upper right. Below, the operator can check the real-time 1-min measurements of the previous 30 minutes to adapt decisions. 112

A.1 Representation of our CNN-encoder. We use 3 convolutional layers, with batch normalization, ReLU and MaxPool in between. We use fully connected (dense) layers to obtain in the end a one-dimensional vector x_{ti}^{emb}. 120

A.2 Illustration of the tensor decomposition of a 3 dimensional tensor. Tensor $\mathcal{A}$ is the original tensor, which is approximated through Tucker decomposition using a core tensor tensor $\mathcal{S}$ and three linear projection maps along each axis $U^{(1)}, U^{(2)}, U^{(3)}$. 122

B.1 (a) Demand damping estimation for Louisiana state (LA). 128

B.2 (b) Demand damping estimation for New York state (NY). 128

B.3 Different piece-wise linear demand damping curves corresponding to different rate of decline. 129

B.4 Map illustrating machine learning (ML) predicted risks for all states surpassing 90th-percentile flooding risk within a 5-year time frame from 2016. Regions are color-coded, with darker shades indicating a higher probability of predicted risks. 131

List of Tables

2.1 Number of TCs meeting our selection criteria from the dataset. We show for each basin and storm category: from Tropical Storm (TS) to Hurricanes of category 1 to 5. We also report the total number of 3-hour interval cases we used from each basin. 30

2.2 List of features included in our statistical data. 32

2.3 Summary of all operational forecast models included in our benchmark. . . . 34

2.4 Summary of the various versions of the Hurricast framework for which we report results. Models differ in architecture and data used and are named based on these two characteristics. 40

2.5 Mean absolute error (MAE), forecast skill with respect to CLP5, and standard deviation of the error (Error sd) of standalone Hurricast models and operational forecasts on the same test set between 2016 and 2019, for 24-hour lead time track forecasting task. Bold values highlight the best performance in each category. 43

2.6 Mean absolute error (MAE), forecast skill with respect to Decay-SHIPS, and standard deviation of the error (Error sd) of standalone Hurricast models and operational forecasts on the same test set between 2016 and 2019, for 24-hour lead time intensity forecasting task. Bold values highlight the best performance in each category. 43

2.7 Mean absolute error (MAE), forecast skill with respect to CLP5, and standard deviation of the error (Error sd) of consensus models compared with NHC's official model OFCL on the same test set between 2016 and 2019 for track forecasting task. Bold values highlight the best performance in each category. 45

2.8 Mean absolute error (MAE), forecast skill with respect to Decay-SHIPS, and standard deviation of the error (Error sd) of consensus models compared with NHC's official model OFCL on the same test set between 2016 and 2019 for intensity forecasting task. Bold values highlight the best performance in each category. 45

3.1 Distribution of each natural disaster as a percentage of total disaster incidents in the dataset across from all years. 52

3.2 A list of all models experimented within the scope of this chapter, models defer in data sources and architectures. 55

3.3 Out-of-sample performance for the next 1,2,5 years of flood risk prediction task. Baseline model predicts the same outcome as current year outcome. Multimodal models employs statistical features and text embeddings extracted using various architectures. We record the number of total features employed (N) in each approach given in brackets. We report ROCAUC score, accuracy, F1 score, and balanced accuracy. 58

4.1 Consolidated out-of-sample accuracy for percentile threshold predictions at 90%, 95%, and 99%, across different time horizons using logistic regression (logreg) and XGBoost (xgb) models. 85

4.2 Summary of models for insurance premium calculation 86

4.3 The number of insolvent states during the testing period based on different methods. In total we test over 52 states, which are the 50 US states with additionally Puerto Rico and the US virgin islands. 89

4.4 Condensed table showing surplus/deficit (S/D) and absolute deviation (A/D) across different γ_2 levels for ARO, RO1, RO2, CMA, and Hist. 91

5.1 Scenario definitions based on wind speed and direction, accounting for five wind speed buckets and three wind direction buckets. Scenarios are differentiated as either favorable (S1, S2, S2b, S3a) or dangerous (S3b, S4). 102

5.2 Categorization of scenarios as favorable and dangerous based on wind speed and direction. 102

5.3 Table recording all the features and processing techniques. The number of features obtained accounts for concatenating the past 48-hour values. 104

5.4 Reward matrix for training the Optimal Policy Trees based upon forecasted and actual weather conditions. 106

5.5 Hyperparameters searched for our models. 107

5.6 Beta test results on the test set for wind speed prediction for all models. We record the MAE and expected shortfall at 85% level for different lead times ranging from 1 hour to 48 hours. 108

5.7 Beta test results on the test set for wind direction prediction for all models. We record the MAE and expected shortfall at 85% level for different lead times ranging from 1 hour to 48 hours. 109

5.8 Performance of three families of models for recommending actions. We include the baseline model, the Elastic Net ensemble, and a series of optimal policy trees trained with different health costs associated with false negatives. . . . 110

B.1 Statistics of top 10 most costly states in the US, including maximum annual claim loss, mean, standard deviation and median. 127

B.2 Hyperparameters searched for our models. 130

B.3 Consolidated out-of-sample accuracy for percentile threshold predictions at 90%, 95%, and 99%, across different time horizons using logistic regression (logreg) and XGBoost (xgb) models. 132

Chapter 1

Introduction

Climate change stands as one of most urgent challenges of our generation. Globally, we are experiencing more extreme weather conditions in recent years: devastating floods in Pakistan, heartbreaking earthquakes in Turkey, and unprecedented wildfires in Canada. Mitigating measures to slow down the changing climate are critical, but how to adapt to a more extreme climate regime is a pressing and critical problem our society confronts.

Central to climate adaptation efforts is the capacity to accurately predict weather conditions, thereby enhancing preparedness. This pursuit dates back to the dawn of civilization. Circa 340 B.C., the Greek philosopher Aristotle documented in his work *Meteorologica* early theories on the formation of rain and clouds through the observation of weather patterns. Fast forward to the early 20th century, with improved scientific understanding of the physical laws governing earth's atmosphere, the field of dynamical modeling was born and has ever since been the primary method underlying modern meteorology. Today, meteorology is yet undergoing another transformation with the integration of machine learning (ML), drawing upon ancient pattern recognition methodologies enhanced by vast datasets and advanced data processing techniques.

This thesis navigates the question of how machine learning and optimization methods can be employed to foster adaptation, build resilience, and support sustainable development in the context of climate change. We showcase various works where machine learning and optimization techniques have been applied to tackle climate adaptation and sustainability challenges. Specifically, we have developed multimodal machine learning models that integrate diverse data formats — ranging from imagery and textual data to tabular datasets — for weather forecasting, covering both short-term hurricane forecasts and long-term extreme weather risk models. The promising predictive capabilities of these ML models have the potential to revolutionize numerous industries. For instance, we have designed an adaptive robust optimization framework for disaster insurance, incorporating ML-derived risks. Furthermore, we have derived a data-driven operating scheme for weather-dependent industrial operations in order to mitigate the impact of airborne pollutants from factories on surrounding urban areas. Through these examples, the thesis highlights the transformative power of machine learning in redefining industries through data-drieven decision-making processes.

1.1 Organization

Broadly speaking the thesis is organized into two categories: the methodological component of multimodal machine learning and its the applications in addressing issues around climate adaptation and sustainable development. The former proceeds by exploring the synthesis of data from a variety formats and sources applied to forecast the weather, spanning from tabular-based time series, image-based satellite data, and text-based language data. The latter focuses on decision-making applications utilizing the novel machine learning driven extreme weather forecasting models.

1.1.1 Multimodal Machine Learning

Multimodal learning is an intuitive way of making inference. Drawing on the analogy of an ancient Indian parable depicted in Figure 1.1, where each blind man touches a different part of an elephant and only through the holistic integration of their perceptions can they form an accurate picture of the animal. This analogy underscores the essence of multimodal machine learning, which integrates diverse insights from various sources and modalities to enhance prediction processes. This section of the thesis seeks to explore how multimodal machine learning can be leveraged to improve the forecasting capabilities of extreme weather events.

Broadly speaking, meteorological data is categorized into three main types: imagery based satellite observation data, tabular time series historical data, and text based descriptive data. In this part of the thesis, we showcase the effectiveness of a multimodal approach through two distinct case studies in extreme weather forecasting: a short-term hurricane forecast with a 12-hour lead time and a long-term flood risk assessment model. We delve into various methods of integrating subsets of these data types and applying diverse data processing techniques. These works affirm the feasibility and promise of leveraging machine learning for predicting extreme weather events over both short and long durations. Moreover, they highlight the superiority of multimodal approaches over single-modal machine learning approach. The papers that compose this section are Hurricane Forecasting: A Multimodal Machine Learning Approach [1] and Global Flood Prediction: A Multimodal Machine Learning Approach [2]. As climate change leads to more frequent natural disasters, we hope these works would advance the role of machine learning in developing accurate and reliable risk models.

Figure 1.1: Illustration of blind men touching an elephant.

1.1.2 Applications around Climate Change Adaptation

Machine learning weather forecasting models offer two key advantages over traditional dynamical models: significant reductions in computational time — from hours to mere seconds — thus enabling real-time, location-specific predictions, also known as "nowcasting", and the ability to develop long-term risk models for proactive disaster mitigation rather than reactive responses. These long-term risk models have the potential to revolutionize many sectors, including infrastructure planning, insurance, urban development, and sustainable energy management, among others.

Therefore, the latter part of this thesis is dedicated to exploring the application of innovative machine learning-driven weather models to address issues around climate change adaptation and sustainable development. The first work introduces an adaptive robust optimization framework for catastrophe insurance that leverages long-term machine learning risk assessments. The second work outlines a real-time machine learning framework for wind forecasting, aimed at adjusting factory production levels to minimize air pollution and its impact on surrounding urban areas. This project was conducted in collaboration with the OCP group and has been implemented in their Safi manufacturing plant since December 2022. The papers that compose this section of the thesis are Catastrophe Insurance: An Adaptive Robust Optimization Approach [3] and Reducing Air Pollution Through Machine Learning [4].

1.2 Structure and Contributions

A chapter by chapter description of the thesis is as follows:

Chapter 2 Introduces the multimodal machine learning model to hurricane forecasting. The task comprises of 12-hour ahead forecasting for two tasks: intensity and track. Results suggest that machine learning approach produce highly comparable results to leading dynamical models. The main advantage being short deployment time - once models are trained, deployment take seconds with much less computational requirement.

This is joint work with Léonard Boussioux, Théo Guénais, and Dimitris Bertimas, and appears in Weather and Forecasting [1].

Chapter 3 Extends the multimodal machine learning approach to obtain long-term flood risk models, and further incorporates language-based text description data. We combine geographical information about locations through text data with tabular time series. The work exhibits a generalizable approach to deploy machine learning models for long-term alert system for extreme weathers, such as draughts, wildfires, etc.

This is joint work with Dimitris Bertsimas, and appears as a workshop paper in ICLR [2].

Chapter 4 Develops an adaptive robust optimization framework for catastrophe insurance scheme. We construct robust uncertainty sets using machine learning driven risk models and historical data. Results show that among tested optimization models, ARO models with conservative parameter values to achieve superiority in both effectiveness and efficiency. Overall, this optimization approach offers versatility and generalizability, making it adaptable to a variety of natural disaster scenarios, such as wildfires, droughts, etc.

This is joint work with Dimitris Bertsimas, and is under review at Management Science [3].

Chapter 5 Develops a near-term predictive-and-prescriptive framework to regulate industrial operations, that emits air-borne pollutants, through machine learning enhanced near-term wind forecasting models. We have successfully implemented our framework in collaboration with OCP Group's phosphate production site near the city of Safi. Results show that our machine learning algorithm significantly reduced forecasting errors, leading to reduced air pollution impact as well as increased planning visibility.

This is joint work with Léonard Boussioux, Dimitris Bertsimas, and our industry collaborators at OCP Group. The work is under review at MSOM [4].

Chapter 6 Summarizes the major threads of the thesis and provides some closing remarks.

Chapter 2

Hurricane Forecasting: A Novel Multimodal Machine Learning Framework

This chapter describes a novel machine learning (ML) framework for tropical cyclone intensity and track forecasting, combining multiple ML techniques and utilizing diverse data sources. Our multimodal framework, called Hurricast, efficiently combines spatial-temporal data with statistical data by extracting features with deep-learning encoder-decoder architectures and predicting with gradient-boosted trees. We evaluate our models in the North Atlantic and Eastern Pacific basins on 2016-2019 for 24-hour lead time track and intensity forecasts and show they achieve comparable mean absolute error and skill to current operational forecast models while computing in seconds. Furthermore, the inclusion of Hurricast into an operational forecast consensus model could improve over the National Hurricane Center's official forecast, thus highlighting the complementary properties with existing approaches. In summary, our work demonstrates that utilizing machine learning techniques to combine different data sources can lead to new opportunities in tropical cyclone forecasting.

2.1 Introduction

A tropical cyclone (TC) is a low-pressure system originating from tropical or subtropical waters and develops by drawing energy from the sea. It is characterized by a warm core, organized deep convection, and a closed surface wind circulation about a well-defined center. Every year, tropical cyclones cause hundreds of deaths and billions of dollars of damage to households and businesses [5]. Therefore, producing an accurate prediction for TC track and intensity with sufficient lead time is critical to undertake life-saving measures.

The forecasting task encompasses the track, intensity, size, structure of TCs, and associated storm surges, rainfall, and tornadoes. Most forecasting models focus on producing track (trajectory) forecasts and intensity forecasts, i.e., intensity measures such as the maximum sustained wind speed in a particular time interval. Current operational TC forecasts can be classified into dynamical models, statistical models, and statistical-dynamical models [6]. Dynamical models, also known as numerical models, utilize powerful supercomputers to simulate atmospheric fields' evolution using dynamical and thermodynamical equations [7]. Statistical models approximate historical relationships between storm behavior and storm-specific features and, in general, do not explicitly consider the physical process [8], [9]. Statistical-dynamical models use statistical techniques but further include atmospheric variables provided by dynamical models [10]. Lastly, ensemble models combine the forecasts made by multiple runs of a single model [6]. Moreover, consensus models typically combine individual operational forecasts with a simple or weighted average [6], [11]–[13].

In addition, recent developments in Deep Learning (DL) enable Machine Learning (ML) models to employ multiple data processing techniques to process and combine information from a wide range of sources and create sophisticated architectures to model spatial-temporal relationships. Several studies have demonstrated the use of Recurrent Neural Networks (RNNs) to predict TC trajectory based on historical data [14]–[16]. Convolutional Neural Networks (CNNs) have also been applied to process reanalysis data and satellite data for track forecasting [17]–[19] and storm intensification forecasting [20], [21].

This chapter introduces a machine learning framework called Hurricast (HUML) for both intensity and track forecasting by combining several data sources using deep learning architectures and gradient-boosted trees.

Our contributions are three-fold:

1. We present novel multimodal[1] machine learning techniques for TC intensity and track predictions by combining distinct forecasting methodologies to utilize multiple individual data sources. Our Hurricast framework employs XGBoost models to make predictions using statistical features based on historical data and spatial-temporal features extracted with deep learning encoder-decoder architectures from atmospheric reanalysis maps.

2. Evaluating in the North Atlantic and Eastern Pacific basins, we demonstrate that our machine learning models produce comparable results to currently operational models for 24-hour lead time for both intensity and track forecasting tasks.

3. Based on our testing, adding one machine learning model as an input to a consensus model can improve the performance, suggesting the potential for incorporating machine learning approaches for hurricane forecasting.

The chapter is structured as follows: Section 2.2 describes the data used in the scope of this study; Section 2.3 explains the operational principles underlying our machine learning models; Section 2.4 describes the experiments conducted; Section 2.5 deals with conclusions from the results and validates the effectiveness of our framework. Finally, Section 2.6 discusses limitations and future work needed for the potential operational deployment of such ML approaches.

[1] Multimodality in machine learning refers to the simultaneous use of different data formats, including, for example, tabular data, images, time series, free text, audio.

2.2 Data

In this study, we employ three kinds of data dated since 1980: historical storm data, reanalysis maps, and operational forecast data. We use all storms from the seven TC basins since 1980 that reach 34 kt maximum intensity at some time, i.e., are classified at least as a tropical storm, and where more than 60 h of data are available after they reached the speed of 34 kt for the first time. Table 2.1 summarises the TCs distribution in each basin included in our data.

Table 2.1: Number of TCs meeting our selection criteria from the dataset. We show for each basin and storm category: from Tropical Storm (TS) to Hurricanes of category 1 to 5. We also report the total number of 3-hour interval cases we used from each basin.

Basin	**TC Category**						**Total TC**	**Total Cases**
	TS	1	2	3	4	5		
Eastern North Pacific (EP)	109	112	57	59	100	14	451	20,970
North Atlantic (NA)	108	96	46	42	46	17	355	18,468
North Indian (NI)	36	13	10	6	8	1	74	2,540
South Atlantic (SA)	1	1	0	0	0	0	2	16
Southwest Indian (SI)	179	96	73	71	28	0	447	25,538
Southern Pacific (SP)	117	76	38	45	15	1	292	13,319
Western North Pacific (WP)	422	240	158	128	29	1	978	53,148
All Basins	972	634	382	351	226	34	2,599	133,999

2.2.1 Historical Storm Data Set

We obtained historical storm data from the National Oceanic and Atmospheric Administration through the post-season storm analysis dataset IBTrACS [22]. Among the available features, we have selected time, latitude, longitude, and minimum pressure at the center of the TC, distance-to-land, translation speed of the TC, direction of the TC, TC type (disturbance, tropical, extra-tropical, etc.), basin (North-Atlantic, Eastern Pacific, Western Pacific, etc), and maximum sustained wind speed from the WMO agency (or from the regional agency when not available). Our overall feature choice is consistent with previous statistical forecasting approaches [10], [19], [23]. In this chapter, we will refer to this data as *statistical data* (see Table 5.3).

The IBTrACS dataset interpolates some features to a 3-hour frequency from the original 6-hour recording frequency. It provides a spline interpolation of the position features (e.g., latitude and longitude) and a linear interpolation of the features not related to position (wind speed, pressure reported by regional agencies). However, the WMO wind speed and pressure were not interpolated by IBTrACS and we interpolated them linearly to match the 3-hour frequency.

We processed statistical data through several steps before inputting it into machine learning models. First, we treated the categorical features using the one-hot encoding technique: for a specific categorical feature, we converted each possible category as an additional binary feature, with 1 indicating the sample belongs to this category and 0 otherwise. We encoded the basin and the nature of the TC as one-hot features. Second, we encoded cyclical features using cosine and sine transformations to avoid singularities at endpoints. Features processed using this smoothing technique include date, latitude, longitude, and storm direction[2].
We also engineer two additional features per time-step to capture first-order dynamical effects: the latitude and longitude displacements in degrees between two consecutive steps.

Finally, the maximum sustained wind speed feature reported can have different averaging policies depending on the specific reporting agency: 1-minute for US basins and 10-minute for other WMO Regional Specialized Meteorological Centres. We adjust all averaging time periods to 1-minute by dividing the 10-minute values by 0.93 as recommended by [24].

2.2.2 Reanalysis Maps

In our work, we used the extensive ERA5 reanalysis data set [25] developed by the European Centre for Medium-Range Weather Forecasts (ECWMF). ERA5 provides hourly estimates

[2] For example, we encoded the latitude value by $\cos(\pi \cdot \frac{\text{lat}}{180})$ and $\sin(\pi \cdot \frac{\text{lat}}{180})$ and the date value by $\cos(2\pi \cdot \frac{\text{date}}{365})$ and $\sin(2\pi \cdot \frac{\text{date}}{365})$.

Table 2.2: List of features included in our statistical data.

Feature	Range	Unit	Type	Processing	Description
Latitude	[-90.000, 90.000]	deg north	numerical	spline interpolation by IBTrACS, standardize	Latitude of the center of the hurricane.
Longitude	[-180.000, 180.000]	deg east	numerical	spline interpolation by IBTrACS, standardize	Longitude of the center of the hurricane.
WMO Wind	[10, 165]	knots	numerical	linear interpolation, conversion to 1-min, standardize	Maximum sustained wind speed from the WMO agency for the current location.
WMO Pressure	[880, 1022]	mb	numerical	linear interpolation, standardize	Wind pressure from the WMO agency for the current location.
Distance to Land	[0, 4821]	km	numerical	standardize	Distance to land from the current position. The IBTrACS land mask includes islands larger than 1400 km^2.
Storm Speed	[0, 69]	knots	numerical	standardize	Translation speed of the system as calculated from the positions in latitude and longitude.
Storm Direction	[0, 360]	deg	numerical	cosine & sine encoding	Translation direction of the system, as calculated from the positions, pointing in degrees east of north.
Storm Displacement Latitude	[-2.68, 3.13]	deg	numerical	standardize	Engineered feature, indicating latitude change since the last time step (3 hours ago).
Storm Displacement Longitude	[-3.83, 4.28]	deg	numerical	standardize	Engineered feature, indicating longitude change since the last time step (3 hours ago).
Basin	[NA, EP, WP, NI, SI, SP, SA]	N/A	categorical	one-hot encoding	Basins include: NA - North Atlantic, EP - Eastern North Pacific, WP - Western North Pacific, NI - North Indian, SI - South Indian, SP - Southern Pacific, SA - South Atlantic
Storm Type	[DS, TS, ET, SS, MX]	N/A	categorical	one-hot encoding	Storm types include: DS - Disturbance, TS - Tropical, ET - Extratropical, SS - Subtropical, NR - Not reported, MX - Mixture (contradicting nature reports from different agencies)

of a large number of atmospheric, land, and oceanic climate variables. The data cover the Earth on a 30km grid and resolve the atmosphere using 137 levels from the surface up to a height of 80km.

We extracted (25° × 25°) maps centered at the storm locations across time, given by the IBTrACS dataset described previously, of resolution $1^\circ \times 1^\circ$, i.e., each cell corresponds to one degree of latitude and longitude, offering a sufficient frame size to capture the entire storm. We obtained nine reanalysis maps for each TC time step, corresponding to three different features (geopotential height z, zonal component of the wind u, meridional component of the wind v) at three pressure levels (225, 500, 700 hPa), as illustrated in Figure 2.1. We chose the three features to incorporate physical information which would influence the TC evolution, and this choice is motivated by previous literature in applying ML techniques to process reanalysis maps [19], [20], [26].

As a remark, we acknowledge two main limitations from using reanalysis maps for TC forecasting. First, since they are reanalysis products, they are not available in real-time and thus significantly hinder operational use. Second, they have deficiencies in representing tropical cyclones [27]–[29]; for instance, with large TC sizes particularly being underestimated [29].

2.2.3 Operational Forecast Models

We obtained operational forecast data from the Automated Tropical Cyclone Forecasting (ATCF) data set, maintained by the National Hurricane Center (NHC) [30], [31]. The ATCF data contains historical forecasts by operational models used by the NHC for its official forecasting for tropical cyclones and subtropical cyclones in the North Atlantic and Eastern Pacific basins. To compare the performance of our models with a benchmark, we selected the strongest operational forecasts with a sufficient number of cases concurrently available: including DSHP, GFSO, HWRF, FSSE, and OFCL for the intensity forecast; CLP5, HWRF, GFSO, AEMN, FSSE, and OFCL for the track forecast (see detailed list in Table 2.3). We extracted the forecast data using the Tropycal Python package [32].

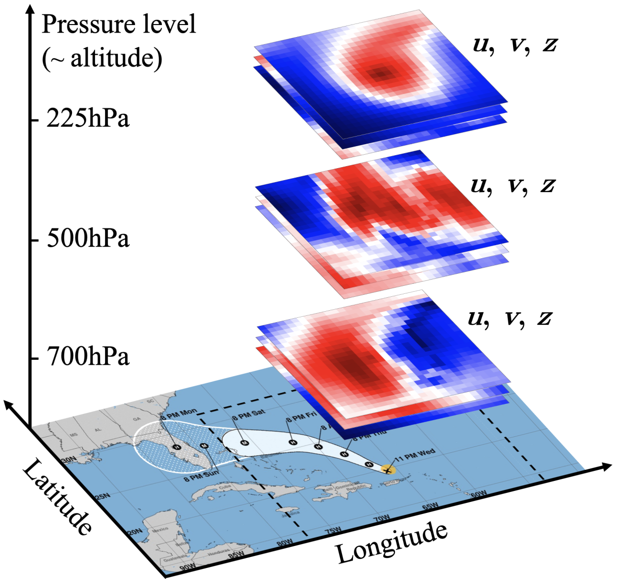


Figure 2.1: Representation of the nine reanalysis maps repeatedly extracted for each time step, corresponding to three different features (geopotential height z, zonal component of the wind u, meridional component of the wind v) at three pressure levels (225, 500, 700 hPa). Each map is of size $25° \times 25°$, centered on the TC center location, and each pixel corresponds to the average field value at the given latitude and longitude degree.

Table 2.3: Summary of all operational forecast models included in our benchmark.

Model ID	Model name or type	Model type	Forecast
CLP5	CLIPER5 Climatology and Persistence	Statistical (baseline)	Track
Decay-SHIPS	Decay Statistical Hurricane Intensity Prediction Scheme	Statistical-dynamical	Intensity
GFSO	Global Forecast System model	Multi-layer global dynamical	Track, Intensity
HWRF	Hurricane Weather Research and Forecasting model	Multi-layer regional dynamical	Track, Intensity
AEMN	GFS Ensemble Mean Forecast	Ensemble	Track
FSSE	Florida State Super Ensemble	Corrected consensus	Track, Intensity
OFCL	Official NHC Forecast	Consensus	Track, Intensity

2.3 Methodology

Our Hurricast framework makes predictions based on time-series data with different formats: three-dimensional vision-based reanalysis maps and one-dimensional historical storm data consisting of numerical and categorical features. The problem of simultaneously using different types of data is broadly known as multimodal learning in the field of machine learning.

Overall, we adopt a three-step approach to combine the multiple data sources. We first extract a one-dimensional feature representation (embedding) from each reanalysis maps sequence. Second, we concatenate this one-dimensional embedding with the statistical data to form a one-dimensional vector. Third, we make our predictions using gradient-boosted tree XGBoost models [33] trained on the selected features.

At a given time step (forecasting case), we perform two 24-hour lead time forecasting tasks: intensity prediction, i.e., predicting the maximum sustained wind speed at a 24-hour lead time; and displacement prediction, i.e., the latitude and longitude storm displacement in degrees between given time and forward 24-hour time. Figure 2.2 illustrates the three-step pipeline.

To perform the feature extraction in Step 1, we have experimented with two computer vision techniques to obtain the reanalysis maps embeddings: (1) encoder-decoder neural networks and (2) tensor decomposition methods. The former is a supervised learning method; for each input, we use an associated prediction target to train the network. On the other hand, tensor decomposition is an unsupervised method; there is no specific labeled prediction target, and instead, embeddings are drawn directly from the patterns within the data.

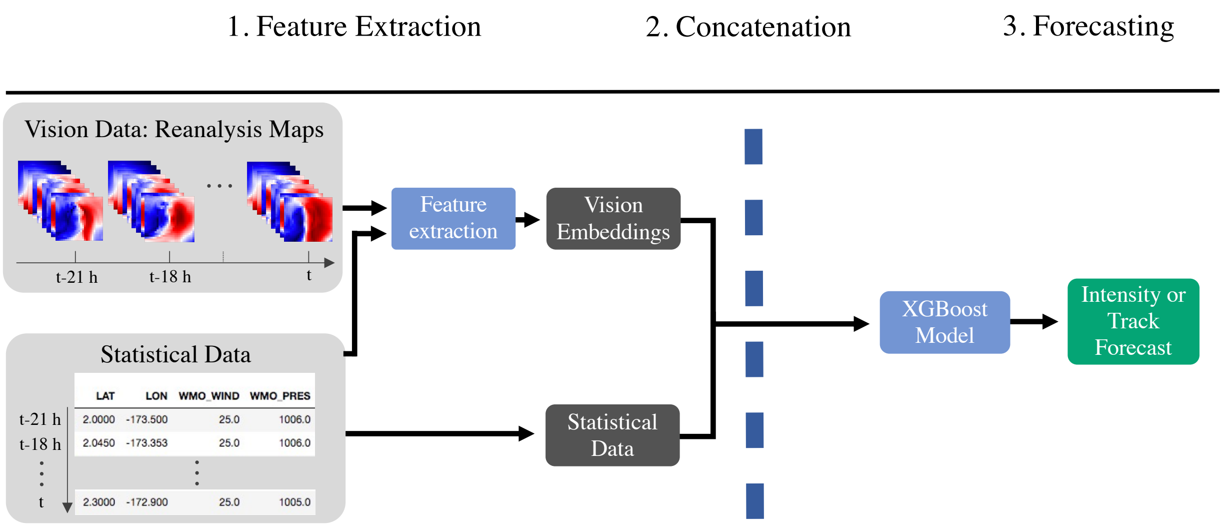


Figure 2.2: Representation of our multimodal machine learning framework using the two data sources: statistical and reanalysis maps. During Step 1, we extract embeddings from the reanalysis maps. In particular, we use encoder-decoder architectures or tensor decomposition to obtain a one-dimensional representation. During Step 2, we concatenate the statistical data with the features extracted from the reanalysis maps. During Step 3, we train one XGBoost model for each of the prediction tasks: intensity in 24 h, latitude displacement in 24 h, and longitude displacement in 24 h.

2.3.1 Feature Extraction

Encoder - Decoder Architectures

The encoder-decoder neural network architecture refers to a general type of deep learning architecture consisting of two components: an encoder, which maps the input data into a latent space; and a decoder, which maps the latent space embeddings into predictions. It is well-suited to deal with multimodal data as different types of neural network layers can be adapted to distinct modalities.

In our work, the encoder component consists of a Convolutional Neural Network (CNN), a successful computer vision technique to process imagery data [34]–[36].

We compare two decoder variations. The first one relies on Gated Recurrent Units (GRU) [37], a well-suited recurrent neural network to model temporal dynamic behavior in sequential data. The second one uses Transformers [38], a state-of-the-art architecture for sequential data. While the GRU model the temporal aspect through a recurrence mechanism, the Transformers utilize attention mechanisms and positional encoding [38], [39] to model long-range dependencies.

First, we train the encoder-decoder architectures using standard backpropagation to update the weights parameterizing the models [40], [41]. We use a mean squared error loss with either an intensity or track objective and add an $L2$ regularization penalty on the network's weights. We then freeze the encoder-decoder's weights when training is completed.

To perform feature extraction from a given input sequence of reanalysis maps and statistical data, we pass them through the whole frozen encoder-decoder, except the last fully-connected layer (see Figures 2.3 and 2.4). The second fully connected layer after the GRU or the pooling layer after the Transformer output a vector of relatively small size, e.g., 128 features, to compress information and provide predictive features. This vector constitutes our one-dimensional reanalysis maps embedding that we extract from the initial 45,000 ($8 \times 9 \times 25 \times 25$) features forming the spatial-temporal input. The motivation is that since the encoder-decoder acquired intensity or track prediction skills during training, it should capture relevant reanalysis maps information in the embeddings. Using these internal features as input to an external model is a method inspired by transfer learning and distillation, generally efficient in visual imagery [42]–[45].

Figures 2.3 and 2.4 illustrate the encoder-decoder architectures. More details on all components are given in Appendix.

Tensor Decomposition

We also explored tensor decomposition methods as a means of feature extraction. The motivation of using tensor decomposition is to represent high-dimensional data using low dimension features. We use the Tucker decomposition definition throughout this work, also known as the higher-order singular value decomposition. In contrast to the aforementioned

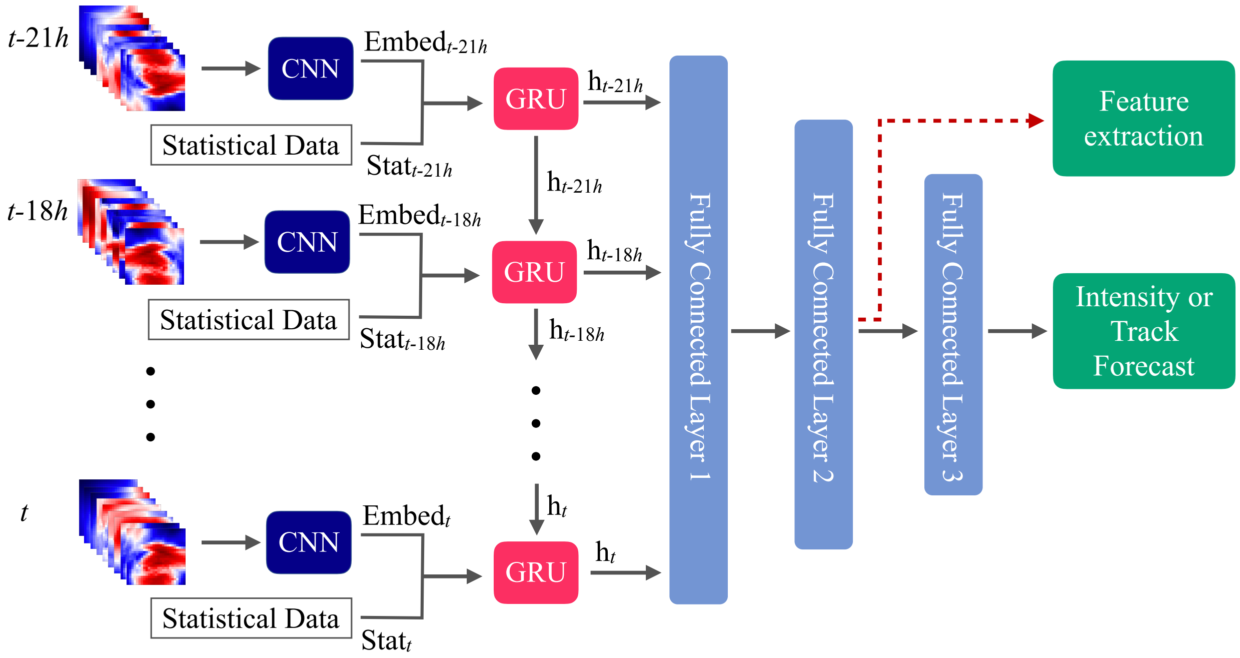


Figure 2.3: Schematic of our CNN-encoder GRU-decoder network for an 8-time step TC sequence. At each time step, we utilize the CNN to produce a one-dimensional representation of the reanalysis maps. Then, we concatenate these embeddings with the corresponding statistical features to create a sequence of inputs fed sequentially to the GRU. At each time step, the GRU outputs a hidden state passed to the next time step. Finally, we concatenate all the successive hidden states and pass them through three fully connected layers to predict intensity or track with a 24-hour lead time. We finally extract our spatial-temporal embeddings as the output of the second fully connected layer.

neural network-based feature processing techniques, tensor decomposition is an unsupervised extraction technique, meaning features are not learned with respect to specific prediction targets.

At each time step, we treated past reanalysis maps over past time steps as a four-dimensional tensor of size $8 \times 9 \times 25 \times 25$ (corresponding to 8 past time steps of 9 reanalysis maps of size 25 pixels by 25 pixels). We used the core tensor obtained from the Tucker decomposition as extracted features after flattening it. We decomposed the tensor using the multilinear singular value decomposition (SVD) method, which is computationally efficient [46].

The size of the core tensor, i.e., the Tucker rank of the decomposition, is a hyperparameter to be tuned. Based on validation, the Tucker rank is tuned to size $3 \times 5 \times 3 \times 3$. More details

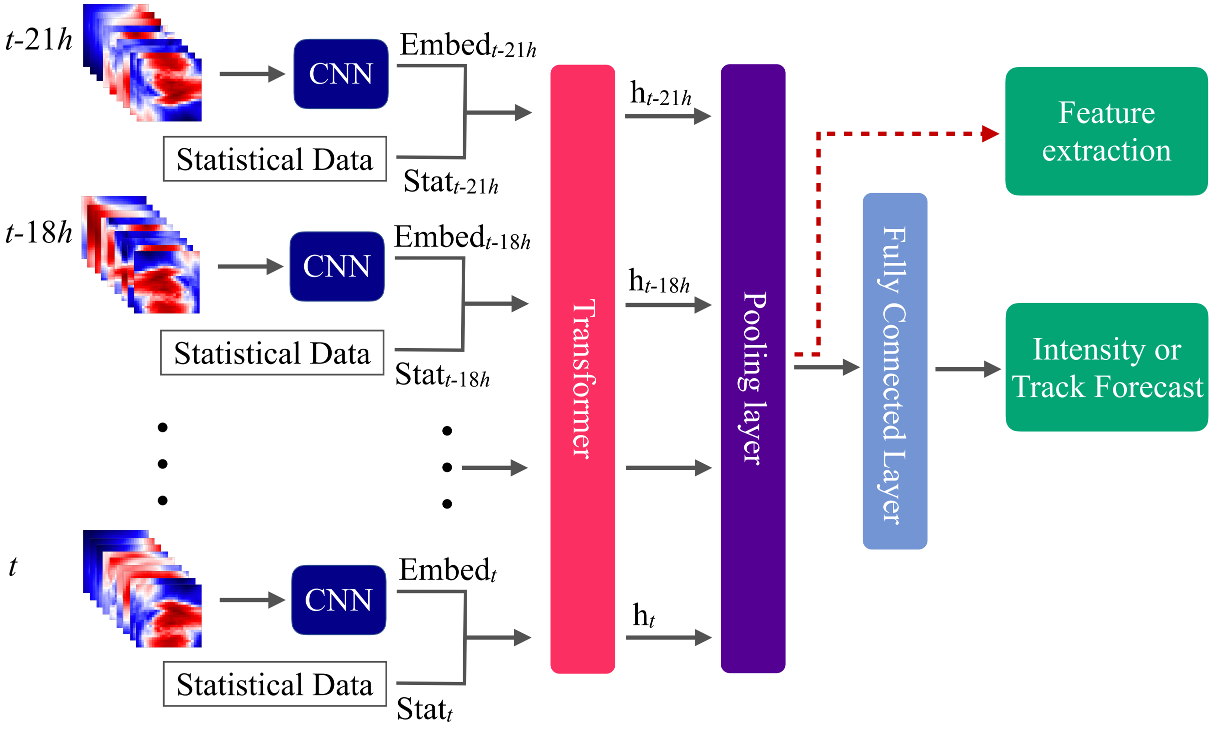


Figure 2.4: Schematic of our CNN-encoder Transformer-decoder network for an 8-time step TC sequence. At each time step, we utilize the CNN to produce a one-dimensional representation of the reanalysis maps. Then, we concatenate these embeddings with the corresponding statistical features to create a sequence of inputs fed as a whole to the Transformer. The Transformer outputs a new 8-timestep sequence that we average (pool) feature-wise and then feed into one fully connected layer to predict intensity or track with a 24-hour lead time. We finally extract our spatial-temporal embeddings as the output of the pooling layer.

on tensor decomposition methodology can be found in the Appendix.

2.3.2 Forecasting Models

During step 2, we concatenated features from relevant data sources to form a one-dimensional input vector corresponding to each forecasting case.

First, we reshaped the statistical data sequence corresponding to the fixed window size of past observations into a one-dimensional vector. Then, we concatenated it to the one-dimensional reanalysis maps embeddings obtained with one of the feature extraction techniques.

During step 3, we used XGBoost models for the track and intensity forecasts. XGBoost is a gradient-boosted tree-based model widely used in the machine learning community for superior modeling skills and efficient computation time. We compared several other machine learning models during the experimentation phase, including Linear Models, Support Vector Machines, Decision Trees, Random Forests, Feed-forward Neural Networks, and found XGBoost to be generally the most performing.

2.3.3 Summary of Models

This section lists all the forecast models tested and retained and summarizes the methodologies employed in Table 2.4.

Table 2.4: Summary of the various versions of the Hurricast framework for which we report results. Models differ in architecture and data used and are named based on these two characteristics.

N°	Name	Data Used	ML Methods
1	HUML-(stat, xgb)	Statistical	XGBoost
2	HUML-(stat/viz, xgb/td)	Statistical, Vision embeddings	XGBoost, Feature extraction with tensor decomposition
3	HUML-(stat/viz, xgb/cnn/gru)	Statistical, Vision embeddings	XGBoost, Feature extraction with CNN, GRU
4	HUML-(stat/viz, xgb/cnn/transfo)	Statistical, Vision embeddings	XGBoost, Feature extraction with CNN, Transformers
5	HUML-ensemble	Models 1-4 forecasts	ElasticNet
6	HUML/OP-average	Operational forecasts, HUML-(stat/viz, xgb/cnn/transfo)	Simple average

Models 1-4 are variations of the three-step framework described in Figure 2.2, with the variation of input data source or processing technique. Model 1, HUML-(stat, xgb), has the simplest form, utilizing only statistical data. Models 2-4 utilize statistical and vision data and are referred to as multimodal models. They differ on the extraction technique used on the reanalysis maps. Model 2, HUML-(stat/viz, xgb/td), uses vision features extracted with tensor decomposition technique. In contrast, Models 3 and 4 utilize vision features extracted with the encoder-decoder, with GRU and Transformer decoders, respectively. Model 5, HUML-ensemble is a weighted consensus model of Models 1 to 4. The weights given to each model are optimized using ElasticNet. Model 6 is a simple average consensus of a few operational forecasts models used by the NHC and our Model 4, HUML-(stat/viz, xgb/cnn/transfo). We use Model 6 to explore whether the Hurricast framework can benefit current operational forecasts by comparing its inclusion as a member model.

2.4 Experiments

2.4.1 Evaluation Metrics

To evaluate our intensity forecasts' performance, we computed the mean absolute error (MAE) on the predicted 1-minute maximum sustained wind speed in 24 hours, as provided by the NHC for the North Atlantic and Eastern Pacific basins, defined as:

$$\text{MAE} := \frac{1}{N}\sum_{i=1}^{N}\left|y_i^{\text{true}} - y_i^{\text{pred}}\right|,$$

where N is the number of predictions, y_i^{pred} the predicted forecast intensity with a 24-hour lead time and y_i^{true} the ground-truth 1-min maximum sustained wind speed value given by the WMO agency.

We computed the mean geographical distance error in kilometers between the actual position and the predicted position in 24 hours to evaluate our track forecasts' performance, using the Haversine formula. The Haversine metric (see Appendix for the exact formula) calculates the great-circle distance between two points — i.e., the shortest distance between these two points over the Earth's surface.

We also report the MAE error standard deviation and the forecast skills, using Decay-SHIPS and CLP5 as the baselines for intensity and track, respectively.

2.4.2 Training, Validation and Testing Protocol

We separated the data set into training (80% of the data), validation (10% of the data), and testing (10% of the data). The training set ranges from 1980 to 2011, the validation set from 2012 to 2015, and the test set from 2016 to 2019. Within each set, we treated all samples independently.

The test set comprises all the TC cases between 2016 and 2019 from the NA and EP basins where the operational forecast predictions are concurrently available as benchmarks. We compare all models on the same cases.

We use data from all basins during training and validation, but we only report performance on the North Atlantic and Eastern Pacific basins, where we have operational forecast data available.

The precise validation-testing methodology and hyperparameter tuning strategy are detailed in Appendix.

2.4.3 Computational Resources

Our code is available at https://github.com/leobix/hurricast. We used Python 3.6 [47] and we coded neural networks using Pytorch [48]. We trained all our models using one Tesla V100 GPU and 6 CPU cores. Typically, our encoder-decoders trained within an hour, reaching the best validation performance after 30 epochs. XGBoost models trained within two minutes. When making a new prediction at test time, the whole model (feature extraction + XGBoost) runs within a couple of seconds, which shows practical interest for deployment. The bottleneck lies in the acquisition of the reanalysis maps only. We further discuss this point in Section 2.6.2.6.1.

2.5 Results

2.5.1 Standalone machine learning models produce a comparable performance to operational models.

For 24-hour lead time track forecasting, as shown in Table 2.5, the best Hurricast model, HUML-(stat/viz, xgb/cnn/transfo), has a skill with respect to CLP5 of 40% on the EP basin. In comparison, HWRF has a skill of 45% and GFSO 46%. On the NA basin, HUML-(stat/viz, xgb/cnn/transfo) has a skill of 46%, compared to 63% for HWRF and 65% for GFSO.

For 24-hour lead time intensity forecasting, as shown in Table 2.6, the multimodal Hurricast models have a better MAE and lower standard deviation in errors than Decay-SHIPS, HWRF, and GFSO in the EP basin. In particular, our best model, HUML-(stat/viz, xgb/cnn/transfo), outperforms Decay-SHIPS by 12% and HWRF by 3% in MAE. In the NA basin, HUML-(stat/viz, xgb/cnn/transfo) underperforms Decay-SHIPS by 2% and HWRF by 7% but has a lower error standard deviation.

These results highlight that machine learning approaches can emerge as a new methodology to currently existing forecasting methodologies in the field. In addition, we believe there is potential for improvement if given more available data sources.

Table 2.5: Mean absolute error (MAE), forecast skill with respect to CLP5, and standard deviation of the error (Error sd) of standalone Hurricast models and operational forecasts on the same test set between 2016 and 2019, for 24-hour lead time track forecasting task. Bold values highlight the best performance in each category.

Model Type	Model Name	Eastern Pacific Basin Comparison on 837 cases			North Atlantic Basin Comparison on 899 cases		
		MAE (km)	Skill (%)	Error sd (km)	MAE (km)	Skill (%)	Error sd (km)
Hurricast (HUML) Methods	HUML-(stat, xgb)	81	33	47	144	28	108
	HUML-(stat/viz, xgb/td)	81	33	47	140	30	108
	HUML-(stat/viz, xgb/cnn/gru)	**72**	**40**	**43**	111	45	79
	HUML-(stat/viz, xgb/cnn/transfo)	**72**	**40**	**43**	**109**	**46**	**71**
Standalone Operational Forecasts	CLP5	121	0	67	201	0	149
	HWRF	67	45	42	75	63	**49**
	GFSO	65	46	45	**71**	**65**	54
	AEMN	**60**	**50**	**37**	73	64	55

Table 2.6: Mean absolute error (MAE), forecast skill with respect to Decay-SHIPS, and standard deviation of the error (Error sd) of standalone Hurricast models and operational forecasts on the same test set between 2016 and 2019, for 24-hour lead time intensity forecasting task. Bold values highlight the best performance in each category.

Model Type	Model Name	Eastern Pacific Basin Comparison on 877 cases			North Atlantic Basin Comparison on 899 cases		
		MAE (kt)	Skill (%)	Error sd (kt)	MAE (kt)	Skill (%)	Error sd (kt)
Hurricast (HUML) Methods	HUML-(stat, xgb)	10.6	9.4	10.5	10.7	−4.9	9.3
	HUML-(stat/viz, xgb/td)	10.6	9.4	10.4	10.6	−3.9	9.2
	HUML-(stat/viz, xgb/cnn/gru)	**10.3**	**12.0**	10.0	10.8	−5.9	9.2
	HUML-(stat/viz, xgb/cnn/transfo)	**10.3**	**12.0**	**9.8**	**10.4**	**−2.0**	**8.8**
Standalone Operational Forecasts	GFSO	15.7	−34.2	14.7	14.2	-39.2	14.1
	Decay-SHIPS	11.7	0.0	**10.4**	10.2	0.0	9.3
	HWRF	**10.6**	**9.4**	11.0	**9.7**	**4.9**	**9.0**

2.5.2 Machine learning models bring additional insights to consensus models.

Consensus models often produce better performance than individual models by averaging out errors and biases. Hence we conducted testing for two consensus models: HUML-ensemble is the weighted average of all individual Hurricast variations; HUML/OP-consensus is a simple average of HUML-(stat/viz, xgb/cnn/transfo) and the other standalone operational models included in our benchmark.

As shown in Tables 2.7 and 2.8, HUML-ensemble consistently improves upon the best performing Hurricast variation in terms of MAE, showcasing the possibility of building practical ensembles from machine learning models.

Moreover, OP-average consensus is the equal-weighted average of available operational forecasts. We constructed the HUML/OP-average consensus with the additional inclusion of the HUML-(stat/viz, xgb/cnn/transfo) model. Results show that the inclusion of our machine learning model brings value into the consensus for both track and intensity tasks. In addition, HUML/OP-average produces lower MAE and standard deviation under our testing scope than the NHC's official forecast OFCL for 24-hour lead time.

In particular, in our 24-hour lead time testing scope, in terms of intensity MAE, HUML/OP-average outperforms OFCL by 8% on the EP basin and 2% on the NA basin. In track MAE, HUML/OP-average outperforms OFCL by 7% on the EP basin and 14% on the NA basin.

As a remark, we do not consider the computational time lag of operational model forecasts in our experiments. Computational time varies and can take several hours for dynamical models. Nevertheless, these results highlight the complementary benefits of machine learning models to operational models.

Table 2.7: Mean absolute error (MAE), forecast skill with respect to CLP5, and standard deviation of the error (Error sd) of consensus models compared with NHC's official model OFCL on the same test set between 2016 and 2019 for track forecasting task. Bold values highlight the best performance in each category.

Model Type	Model Name	Eastern Pacific Basin Comparison on 837 cases			North Atlantic Basin Comparison on 899 cases		
		MAE (km)	Skill (%)	Error sd (km)	MAE (km)	Skill (%)	Error sd (km)
Hurricast Methods	HUML-(stat/viz, xgb/cnn/transfo)	72	40	43	109	46	**71**
	HUML-ensemble	**68**	**44**	**41**	**107**	**47**	76
Operational Forecasts	FSSE	56	54	47	**69**	**66**	**53**
	OFCL	**54**	**55**	**33**	71	65	56
Consensus Models	OP-average consensus	55	55	37	64	68	48
	HUML/OP-average consensus	**50**	**59**	**32**	**61**	**70**	**42**

Table 2.8: Mean absolute error (MAE), forecast skill with respect to Decay-SHIPS, and standard deviation of the error (Error sd) of consensus models compared with NHC's official model OFCL on the same test set between 2016 and 2019 for intensity forecasting task. Bold values highlight the best performance in each category.

Model Type	Model Name	Eastern Pacific Basin Comparison on 877 cases			North Atlantic Basin Comparison on 899 cases		
		MAE (kt)	Skill (%)	Error sd (kt)	MAE (kt)	Skill (%)	Error sd (kt)
Hurricast Methods	HUML-(stat/viz, xgb/cnn/transfo)	10.3	12.0	**9.8**	10.4	-2.0	**8.8**
	HUML-ensemble	**10.2**	**12.8**	9.9	**10.2**	**0.0**	8.9
Operational Forecasts	FSSE	**9.7**	**17.1**	**9.5**	**8.5**	**16.7**	**7.8**
	OFCL	10.0	14.5	10.1	**8.5**	**16.7**	8.1
Consensus Models	OP-average consensus	9.6	17.9	9.7	8.5	16.7	7.9
	HUML/OP-average consensus	**9.2**	**21.4**	**9.0**	**8.3**	**18.6**	**7.6**

2.5.3 A multimodal approach leads to more accurate forecasts than using single data sources.

As shown in Tables 2.5 and 2.6, for both track and intensity forecasts, multimodal models achieve higher accuracy and lower standard deviation than the model using only statistical data.

The deep-learning feature extraction methods outperform the tensor-decomposition-based approach. This is not surprising as our encoder-decoders trained with a supervised learning objective, which means extracted features are tailored for the particular downstream prediction task. Tensor decomposition is, however, advantageously label-agnostic but did not extract features with enough predictive information to improve the performance.

2.6 Limitations and Extensions

2.6.1 The Use of Reanalysis Maps

A significant limitation of reanalysis maps is the computation time for construction, as they are assimilated based on observational data. Thus, although our models can compute forecasts in seconds, the dependence on reanalysis maps is a bottleneck in real-time forecasting. Therefore, a natural extension for effective deployment is to train our models using real-time observational data or field forecasts from powerful dynamical models such as HWRF. Since dynamical models are constantly updated with improved physics, higher resolution, and fixed bugs, reforecast products (e.g., [49]) should be well-suited for training our encoder-decoders. Nevertheless, we hope our framework could provide guidance and reference to build operational machine learning models in the future.

2.6.2 Incorporate Additional Data

Under the scope of this work, we used nine reanalysis maps per time step, corresponding to the geopotential height (z), the zonal (u) and meridional (v) component of the wind fields from three pressure levels. One natural extension is to include additional features, such as the sea-surface temperature, the temperature, and the relative humidity, and include information from more vertical levels to potentially improve model performance.

In addition, one could include more data sources, such as satellite and radar data. Notably, we highlight the flexibility of our framework that can easily incorporate new data: we can adopt different feature extraction architectures and then append or substitute extracted features in the XGBoost forecasting model accordingly.

2.6.3 Longer-Term Forecasts

We conducted our experiments for 24-hour lead time predictions to demonstrate the potential of ML techniques in hurricane forecasting tasks. However, experiments on longer-term forecasts are needed before deploying such approaches. For example, the official NHC forecast provides guidance for up to 5 days. Nevertheless, our framework can be extended to longer lead-time forecasts. In particular, we recommend extending the input window size (from current 24-hour) as our models can process arbitrary long input sequences.

2.7 Conclusion

This study demonstrates a novel multimodal machine learning framework for tropical cyclone intensity and track forecasting utilizing historical storm data and meteorological reanalysis data. We present a three-step pipeline to combine multiple machine learning approaches, consisting of (1) deep feature extraction, (2) concatenation of all processed features, (3) prediction. We demonstrate that a successful combination of deep learning techniques and gradient-boosted trees can achieve strong predictions for both track and intensity forecasts, producing comparable results to current operational forecast models, especially in the intensity task. We acknowledge that the unavailability of real-time reanalysis data poses a challenge for operational use, and suggest future work to extend our framework with other operational data sources.

We demonstrate that multimodal encoder-decoder architectures can successfully serve as a spatial-temporal feature extractor for downstream prediction tasks. In particular, this is also the first successful application of a Transformer-decoder architecture in tropical cyclone forecasting.

Furthermore, we show that consensus models that include our machine learning model could benefit the NHC's official forecast for both intensity and track, thus demonstrating the potential value of developing machine learning approaches as a new branch methodology for tropical cyclone forecasting.

Moreover, once trained, our models run in seconds, showing practical interest for real-time forecast, the bottleneck lying only in the data acquisition. We propose extensions and guidance for effective real-world deployment.

In conclusion, our work demonstrates that machine learning can provide valuable additions to the field of tropical cyclone forecasting. We hope this work opens the door for further use of machine learning in meteorological forecasting.

Acknowledgements

This chapter is based on joint work with Lèonard Boussioux, Thèo Guénais, and Dimitris Bertsimas [1]. We thank the review team of the Weather and Forecasting journal for insightful comments that improved the chapter substantially. We thank Louis Maestrati, Sophie Giffard-Roisin, Charles Guille-Escuret, Baptiste Goujaud, David Yu-Tung Hui, Ding Wang, Tianxing He for useful discussions. We thank Nicolò Forcellini, Miya Wang for proof-reading. The work was partially supported from a grant to MIT by the OCP Group. The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing high-performance computing resources that have contributed to the research results reported within this chapter.

Chapter 3

Global Flood Prediction: A Multimodal Machine Learning Approach

This chapter presents a novel multimodal machine learning approach for multi-year global flood risk prediction, combining geographical information and historical natural disaster dataset. Our multimodal framework employs state-of-the-art processing techniques to extract embeddings from each data modality, including text-based geographical data and tabular-based time-series data. Experiments demonstrate that a multimodal approach, that is combining text and statistical data, outperforms a single-modality approach. Our most advanced architecture, employing embeddings extracted using transfer learning upon DistilBert model, achieves 75%-77% ROCAUC score in predicting the next 1-5 year flooding event in historically flooded locations. This chapter demonstrates the potentials of using machine learning for long-term planning in natural disaster management.

3.1 Introduction

A disastrous flood in 2022 left one third of the land in Pakistan underwater for over four months, affecting 33 million people in the country and causing over 30 billion US dollars of damage [50]. Globally, floods cost billions of dollars each year and inflict massive damage to human life, infrastructure, agriculture, and industrial activities. Most concerningly, studies suggest climate change impacts lead to drastically increasing flooding risks globally in both frequency and scale [51], [52]. Therefore, it is crucial to develop both short-term and long-term predictions for flood events to mitigate damage.

Most established models for flood prediction use physical models to simulate hydrological dynamics. [53] provides a technical review of large-scale hydrodynamical models employed in various continents. The most advanced models take into consideration terrain data, water flow data, river networks [54]. To combine insights from individual models and reduce errors, most forecasting agencies, such as the pan-European Flood Awareness System (EFAS), employ an ensemble of predictions across many individual hydrological models to produce probabilistic forecasts [55].

Physical models dominate short-term flood prediction space; however, they lack forecasting capabilities for a longer horizon due to escalating simulation errors. To address this need, machine learning can emerge as a powerful tool to offer a predictive perspective. [56] provides an extensive literature review on the recent ML approaches. Most early works of machine learning approaches are based on a single modality of data, such as rainfall and water level data [57]–[59], or remote-sensing dataset such as satellite and radars to capture real-time high resolution rain gauges [54], [60]. Multimodal machine learning, referring to models that employ more than one modality of data such as tabular, imagery, text, or other formats, have been recently applied for flood detection purposes. For instance, [61] combines hydrological information with twitter data to detect and monitor flood.

This chapter presents a multimodal machine learning approach combining for global multi-year flood prediction. To the best of our knowledge, this is the first machine learning flood prediction model at the global scale and on a multi-year horizon. In addition, it is the first time text-based data has been applied to flood prediction. Our main contributions are three-fold:

1. A novel multimodal framework to incorporate text-based geographical information to complement time-series statistical features for global flood prediction. We employ

state-of-the art large natural language processing techniques, including fine-tuning and transfer learning on pre-trained BERT models.

2. Our experiments show strong results for multi-year flood risk forecasting, with the strongest model achieving 75%-77% ROCAUC score in the next 1-5 year flooding prediction. In addition, we show that multimodal models, combining text with statistical data, outperform single-modal models using only statistical data.
3. Our framework can be generalised to other natural disaster forecasting tasks such as the wildfires, earthquakes, droughts, and extreme weather events. Thus, this chapter suggests a promising direction in long-term preparation for natural disaster management.

3.2 Data

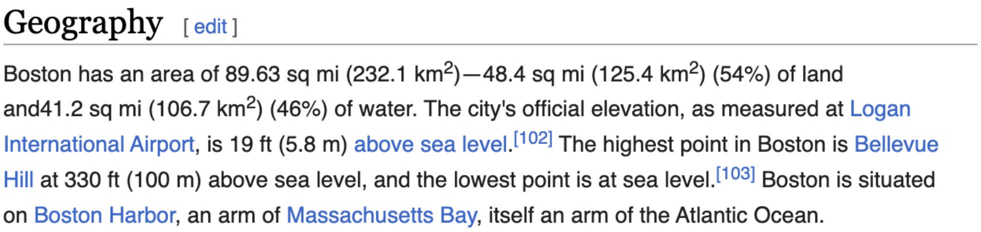
Historical Flood Data. We use the Geocoded Disasters (GDIS) dataset, which includes geocoded information on 9,924 unique natural disasters occurred globally between 1960 and 2018 [62]. In addition, we linked this dataset with the EM-DAT dataset to add additional economic information such as damage estimation [63]. Natural disasters include floods, storms (typhoons, monsoons etc.), earthquakes, landslides (wet and dry), droughts, volcanic activity and extreme temperatures events. Floods account for 43% of all the incidents in this dataset, followed by storms at 29%, and earthquakes at 11%. Detailed distribution can be found in table 3.1 below.

In this project, we restrict forecasting locations to those with historical flooding event. We use the date, latitude, longitude, location (given as the name of the location), and if available, damage cost from this dataset. We divide the earth into 1° by 1° grid, corresponding to about 100km by 100km squares. Using the latitude and longitude information, we compute a 'grid id' for each natural disaster from the GDIS dataset. Overall, there are 2852 unique grid locations in the dataset with a recorded historical natural disaster.

Table 3.1: Distribution of each natural disaster as a percentage of total disaster incidents in the dataset across from all years.

Natural Disaster	Percentage
Flood	0.430
Storm	0.290
Earthquake	0.110
Landslide	0.060
Extreme temperature	0.050
Drought	0.040
Volcanic activity	0.020

Geographical Data. To incorporate the geographical information of each location, we use open-source Wikipedia website's Geographical section, which contain text-based geographical description of certain areas, as shown in Figure 3.1 as an example for the 'Boston' Wikipedia page. To obtain the geographical information, we use the 'location' data from the GDIS dataset for each grid id, then use the Wikipedia-API to obtain the text from the Geographical section for each location [64]. To deal with the noise in the data, since some locations have different names on Wikipedia, we search over synonyms for each location. For those location Wikipedia pages without Geography section, we use the Summary section. Among 2852 unique grid ids, we collected text-based information for 2775 grid ids, and fill the remainder grid ids as 'missing'.



Geography [edit]

Boston has an area of 89.63 sq mi (232.1 km^2)—48.4 sq mi (125.4 km^2) (54%) of land and41.2 sq mi (106.7 km^2) (46%) of water. The city's official elevation, as measured at Logan International Airport, is 19 ft (5.8 m) above sea level.[102] The highest point in Boston is Bellevue Hill at 330 ft (100 m) above sea level, and the lowest point is at sea level.[103] Boston is situated on Boston Harbor, an arm of Massachusetts Bay, itself an arm of the Atlantic Ocean.

Figure 3.1: Example 'Geography' section of the Boston Wikipedia page.

3.3 Methodology

The overall goal is to predict next 1 to 5 years of flood risk using a multimodal approach. The framework adopts a three-step approach to combine distinct data formats and sources.

Figure 3.2 illustrates the overall three-step framework. More details of the training and testing protocol can be found in the Appendix.

1. We gather different sources and modalities of data, which are a) tabular-based historical natural disaster data and b) text-based geographical data from Wikipedia pages.

2. We perform feature processing individually for each data modality, and obtain a one-dimensional feature representation (embeddings) respectively.

3. We concatenate feature embeddings from different modalities and perform feature sections, before making next-N-year flood event predictions using gradient boosted tree (XGBoost) models for binary classification task. Prediction target 1 indicates a flood in the next N years, 0 otherwise.

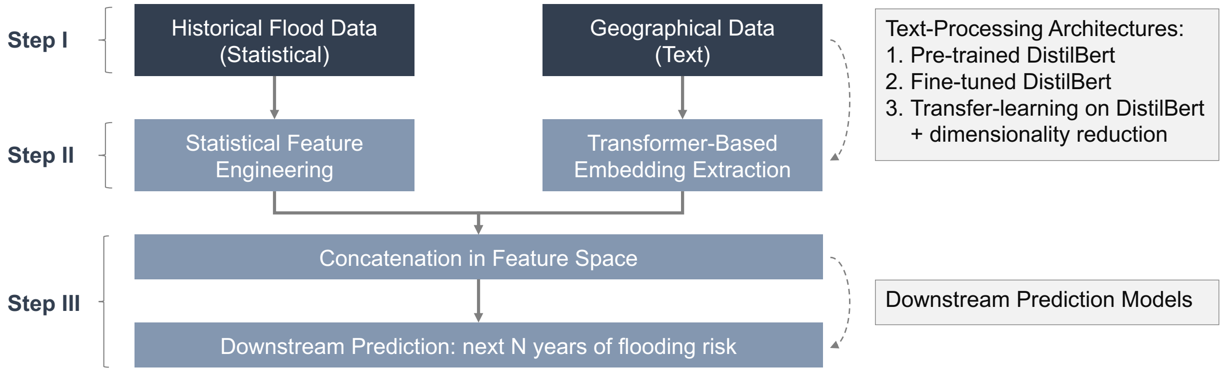

Step I
Step II
Step III
Historical Flood Data (Statistical)
Geographical Data (Text)
Statistical Feature Engineering
Transformer-Based Embedding Extraction
Concatenation in Feature Space
Downstream Prediction: next N years of flooding risk
Text-Processing Architectures:
1. Pre-trained DistilBert
2. Fine-tuned DistilBert
3. Transfer-learning on DistilBert + dimensionality reduction
Downstream Prediction Models

Figure 3.2: Three-step framework to combine statistical data with text-based data. The transformer-based text data embedding extraction contains three types of architectures.

3.3.1 Statistical Feature Processing

We use the GDIS dataset to process historical statistics of natural disasters. In particular, for each grid id, we aggregate statistical features into yearly basis uisng only the current year's natural disaster statistics. In particular, we summarize the 'count' 'binary' and 'damage cost' feature during the year for each natural disaster: 'flood', 'storm', 'earthquake', 'extreme temperature', 'landslide', 'volcanic activity', 'drought', 'mass movement (dry)'. The 'damage cost' feature corresponds to the insurance amount claimed by the natural disaster, which is intended as a proxy to reflect the severity of the natural disaster. In total, the statistical features contain 24 features. Additionally, we record the ‘year’ feature as numerical feature.

3.3.2 Text Feature Processing

For each location, we use the Geography section from the Wikipedia page using the location name. This information is given as text, and each location is associated with a paragraph of geographical information description. Under the scope of this chapter, we experiment with pre-trained large language model DistilBert, a distilled version of the BERT model, which offers good performance whilst faster to train and fine-tune [65]. The two main challenges are: a) DistilBert model is trained on a large set of generic texts, whilst we would like to adapt it to encode geographical information specifically; b) feature extraction is performed on a token-by-token basis, whilst we require embeddings corresponding to a paragraph of sentences. In summary, we experiment with three distinct architectures.

1. The original DistilBert. As proposed by [66], we use the second last layer of hidden states and taking the average of embedding tokens across from all words in the sentence to obtain the paragraph embedding.

2. Fine-tuned version of the DistilBert model. We fine-tune the DistilBertForSequenceClassification model using binary classification labels with 1 indicating the location has more than two historical floods, and 0 indicating the location has less or equal to two historical floods. The motivation is to fine-tune DistilBert embeddings specifically for flood prediction. Then we pool token embeddings by taking the average of the second last layer.

3. Transfer learning and dimensionality reduction. We add an additional linear layer of dimension (796, 32) with a sigmoid activation function. The classification labels are the same as in the second approach, and we use the 32 vector as extracted embeddings. During the training process, parameters from the pre-trained model are frozen, and the training only learns parameters from the linear layer. Similarly as above, we compute paragraph embeddings by taking the average of the 32-vector embeddings for each token.

Model	Description
Baseline	Predicts the next N years of flood outcome as the same current year flood outcome. I.e., if there is a flood occurring at the forecasting year, then the model predicts 1 for the forecasting year.
Statistical	Using only statistical features processed using the GDIS dataset, and we experiment with XGBoost and Logistic Regression as downstream classi
DistilBert Avg	Using the pre-trained DistilBert model, and taking the average embedding of the second last layer across all tokens to form paragraph embedding, contactenated with statistical features.
Finetune Avg	Using the fine-tuned version of DistilBert model, taking the average embedding of the second last layer across all tokens to form paragraph embedding, contactenated with statistical features.
Transfer Learning	Using the transfer learning and dimensionality reduction architecture, taking the average embedding of the 32-dimension embedding across all tokens to form paragraph embedding, contatenated with statistical features.

Table 3.2: A list of all models experimented within the scope of this chapter, models defer in data sources and architectures.

3.4 Training and Testing Protocol

In Step II, for the fine-tuning and transfer learning of transformer-based feature extraction models, we split the text dataset (which contains 2852 locations with associated Wikipedia text data) into training and validation set with 70% randomly selected samples as the training set. Models are trained using SGD with Adam optimiser. Both fine-tuning and transfer learning are trained on 3 epochs.

In Step III, for the training and testing of the downstream binary classification task of flooding risk, we separate the data into 70% training and 30% testing. For each model, we perform 3-fold cross validation on the grid search to perform hyperparameter tuning with AUC score as the scoring metric. we record the following evaluation merics: accuracy, balanced accuracy, ROCAUC score, and F1 score.

Finally, prediction targets are constructed using the GDIS datasets, which records all major flood between 1960 - 2019 globally. For each grid id for a particular year, we process the current year information from the GDIS dataset combined with geographical text-based information, and predict next 1-5 years of flooding risk. Since there is no historical dependence in each sample, we shuffle and split the entire dataset into training (70%) and testing (30%).

The training and fine-tuning of DistilBert models are conducted on Google Colab with 1 GPU computing power. The training and parameter search on classification tasks are conducted using the MIT SuperCloud cluster with 1 GPU computing power [67].

As a remark, due to the rarity of natural disaster occurrence, we face a significant data imbalance challenge: the majority of the grids would not have a flood incidence and, thus, the positive prediction case is less than 0.1% for the entire dataset. To address this issue, we filter to select grid ids with at least 2 historical flood incidents, and perform prediction tasks on those selected grid ids. This filtering criterion is based on the assumption that some grid locations are not prone to flooding risk. Among 2852 unique grids, 881 grids are selected.

3.5 Results

Table 3.2 lists out all models deferring in architectures and data sources experimented under the scope of this chapter. Table 3.3 contains out-of-sample binary classification performance from various models for the next 1,2,5 year flood prediction horizon on the selected 818 grid locations. In summary, a multimodal approach demonstrates the strongest performance, achieving 70% - 75% ROCAUC score. Training and testing sets are randomly selected at 70% and 30%.

We construct a deterministic baseline model which predicts the next N years of flood outcome as the same current year flood outcome. This approach aims to mark previously flooded region as high risk, which is similar to the flood risk mapping procedure employed by agencies such as FEMA.

Due to high class imbalance, metrics such as ROCAUC and balanced accuracy scores are more objective than accuracy scores in evaluating prediction capabilities. We observe that a single-modality model employing only statistical features outperforms the baseline model by around 35% in ROCAUC score and around 25% in balanced accuracy, underperforms the baseline by around 23% in accuracy score. Among multimodal approaches, the strongest architecture combines statistical features with text features obtained using transfer learning upon DistilBert model. This architecture improves upon the baseline model by around 42% in ROCAUC score, 25% in balanced accuracy,and underperforms the baseline by around 13% in accuracy score. Finally, other multimodal architectures, such as using directly pre-trained DistilBert or finetuned DistilBert does not improve the performance from a single-modality approach.

Among multimodal models which employ additional text data, features obtained using transfer learning and dimensionality reduction layer gives the most performing results, improve around 5% across various evaluation metrics from the baseline statistical model.

In general, we observe that adding pre-trained DistilBert embeddings does not show an imporvement on the model performance. Finetuning is helpful in obtaining more meaningful embedding towards the flood prediction task.

3.6 Conclusion

This chapter presents a multimodal machine learning framework for global flood risk forecasting combining statistical natural disaster dataset with text-based geographical information. We have employed state-of-the-art natural language processing tools to encode geographical information given by text-based data from Wikipedia. And the multimodal framework proposed can successfully extract information from text data to complement statistical data. In

Horizon	**Metric**	**Baseline**	**Statistical**	**Multimodal**		
			(N=26)	**DistilBert (N=795)**	**Finetune (N=795)**	**Transfer (N=61)**
1-year	rocauc	0.544	0.742	0.734	0.758	0.772
	f1	0.545	0.519	0.527	0.554	0.558
	acc	0.895	0.707	0.747	0.783	0.783
	acc balanced	0.544	0.681	0.640	0.664	0.675
2-year	rocauc	0.534	0.726	0.724	0.756	0.764
	f1	0.536	0.502	0.525	0.559	0.560
	acc	0.889	0.664	0.742	0.782	0.781
	acc balanced	0.534	0.676	0.627	0.664	0.668
5-year	rocauc	0.539	0.715	0.726	0.749	0.767
	f1	0.541	0.501	0.522	0.545	0.557
	acc	0.892	0.668	0.724	0.758	0.764
	acc balanced	0.539	0.664	0.641	0.658	0.682

Table 3.3: Out-of-sample performance for the next 1,2,5 years of flood risk prediction task. Baseline model predicts the same outcome as current year outcome. Multimodal models employs statistical features and text embeddings extracted using various architectures. We record the number of total features employed (N) in each approach given in brackets. We report ROCAUC score, accuracy, F1 score, and balanced accuracy.

particular, transfer learning based on DistilBert models shows strong results in forecasting capabilities. As a remark, this chapter only experiments with flood prediction, the general framework could be applied to other natural disaster forecasting tasks such as the wildfires, earthquakes, droughts, extreme weather events. This work shows promising experimental results in the strong performance of a machine learning approach towards flood risk forecasting. With further work and more experimentation, a machine learning approach could emerge as a powerful means for medium-to-long term natural disaster risks forecasting.

3.6.1 Future Work

The demonstrated effectiveness of the multimodal machine learning approach in flood prediction not only highlights its current successes but also paves the way for expansive future research and development. The potential areas for this continued exploration include:

- Bringing additional insights from hydrological models and other physically based models. Since flooding is by nature a physical phenomenon, I would like to incorporate further climate and hydrological features into the model. I intend to apply time-series techniques to process these temporal features.

- Incorporate radar and remote sensing data to include more granular climate-related features. In intend to employ computer vision processing techniques to process imagery based data, and investigate imagery as an additional modality to the data sources.

- Interpretability and explainability of the models and features. I intend to conduct some analysis into understanding which features are related to flooding risks, which could serve to improve the credibiltiy of machine learning models and provide valuable insights for monitoring purposes.
- Finally, a useful extension of work is to use global dataset to train local models to perform near-term prediction. Long-term models are useful for resource allocation purposes. Near-term forecasting models can be valuable for local agencies to alert stakeholders and undertake life-saving actions.

3.7 Acknowledgements

We extend our gratitude to Oscar Zhang and Riley Oh for their valuable contributions during his research project, to Faraz Faruqi for insightful discussions, and to Farri Gaba for his meticulous proofreading.

Chapter 4

Catastrophe Insurance Pricing: An Adaptive Robust Optimization Approach

The escalating frequency and severity of natural disasters, exacerbated by climate change, underscore the critical role of insurance in facilitating recovery and promoting investments in risk reduction. This chapter introduces a novel Adaptive Robust Optimization (ARO) framework tailored for the calculation of catastrophe insurance premiums, with a case study applied to the United States National Flood Insurance Program (NFIP). To the best of our knowledge, it is the first time an ARO approach has been applied to for disaster insurance pricing. Our methodology is designed to protect against both historical and emerging risks, the latter predicted by advanced machine learning models, thus directly incorporating amplified risks induced by climate change. Using the US flood insurance data as a case study, optimization models demonstrate effectiveness in covering losses and produce surpluses, with a smooth balance transition through parameter fine-tuning. Among tested optimization models, results show ARO models with conservative parameter values to achieve superiority in both effectiveness and efficiency. Overall, the optimization framework offers versatility and generalizability, making it adaptable to a variety of natural disaster scenarios, such as wildfires, droughts, etc. this chapter not only advances the field of insurance premium modeling but also serves as a vital tool for policymakers and stakeholders in building resilience

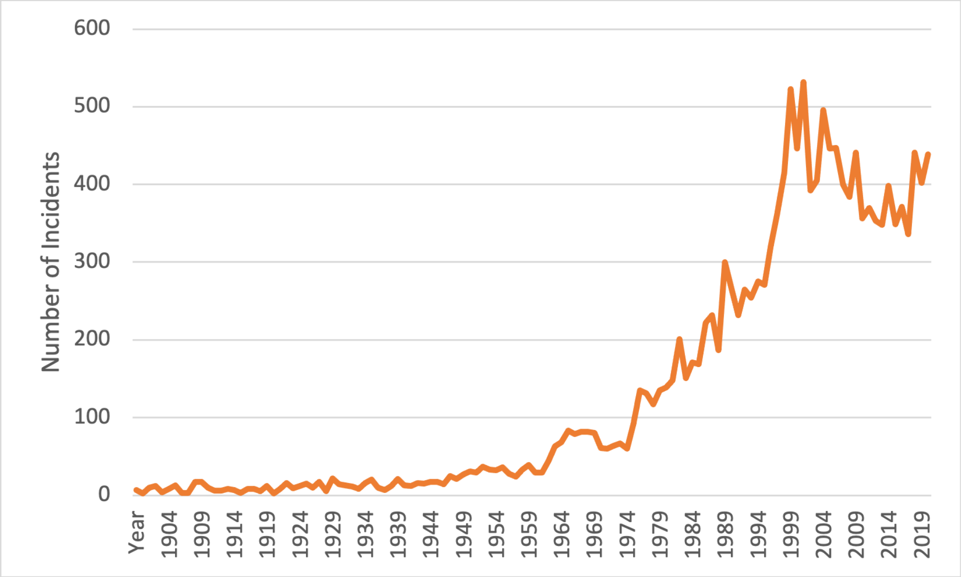


Figure 4.1: Number of major disasters globally since 1900, maintained by the EM-DAT database [63]. A disaster is defined as an event which overwhelms local capacity, necessitating a request to the national or international level for external assistance. Disasters include: flood, storm, earthquake, drought, landslide, extreme temperature, wildfire, volcanic activity, mass movement (dry), glacial lake outburst, fog, etc.

to the growing risks of natural catastrophes.

4.1 Introduction

Global climate change causes serious consequences in climate variability and weather extremes, which could lead to more frequent and costly natural disasters worldwide [68]. As shown in Figure 4.1, the number of disasters worldwide has increased tremendously during the last decades. Enhanced risks for catastrophic events, such as tropical cyclones, draughts, floods, heatwaves, can inflict severe damage and losses on individuals, businesses, communities, and the entire society. It is crucial to mitigate disaster risks and facilitate climate change adaptation [69]. The International Panel on Climate Change (IPCC) has emphasized the need for financial instruments for disaster risk management and climate change adaptation [70]. Catastrophe insurance emerges as a crucial risk management tool, offering financial support in recovery and incentivizing investments for mitigating efforts.

Catastrophe insurance, also known as disaster insurance, focuses on large-scale, low-frequency events that have the potential to cause widespread damage. This form of insurance typically covers disasters such as hurricanes, earthquakes, floods, terrorist acts, and pandemics. The rarity of such catastrophic events complicates the insurance process, as traditional actuarial methods fall short, often due to a lack of comprehensive historical data. Compounding this challenge, climate change is leading to more regular and destructive climate-related catastrophes, which traditional reliance on historical data alone tends to underestimate. In contrast to conventional insurance policies that spread risks across insured individuals, catastrophe insurance confronts a temporal problem of matching of regular influx of annual premiums with the irregular and unforeseeable distribution of payouts for losses. In the United States, catastrophe insurance has historically been managed predominantly through national programs. The National Flood Insurance Program (NFIP), for example, is the principal provider for flood insurance in the country, covering more than 95% of the underwriting risks [71]. However, the NFIP's actuarial effectiveness has been the subject of scrutiny, with the program operating at a significant deficit—19 billion US dollars as of 2023—highlighting the need for reform in the structuring of such insurance schemes [72]. The entry of private insurers into the catastrophe coverage market has been deemed crucial, yet the inherent complexities associated with rare events have led to minimal participation from private entities [73]. Indeed, many private insurers are withdrawing from regions deemed “uninsurable", due to escalating risks from climate change [74].

In this work, we present an Adaptive Robust Optimization (ARO) framework for catastrophe insurance premium pricing designed to protect against uncertain losses. To the best of our knowledge, this is the first work using an ARO approach to set disaster insurance premiums. We develop the framework and implement it to flood insurance using the National Flood Insurance Program (NFIP) data. The main contributions are three-fold:

- We introduce a Robust Optimization (RO) framework for pricing catastrophe insurance premiums, incorporating both historical data and machine learning predictions. We propose two distinct uncertainty sets to model losses: one retrospective, grounded in historical loss distributions, and the other prospective, utilizing risk estimates derived from machine learning predictions. We further extend to Adaptive Robust Optimization (ARO) framework linking premiums with realized losses.

- We apply our ARO framework to US flood insurance using data from the National Flood Insurance Program from 1975 to 2022. Using training data, we parameterize optimization models and train our own machine learning risk models, and evaluate model performance using out-of-sample testing data. Optimization models demonstrate capabilities in effectively covering losses whilst offering the adaptability to policy-makers' risk tolerance and level of conservatism. In particular, we recommend policy makers to choose an ARO model, with conservative parameter values, to achieve superior performance in both effectiveness and efficiency in covering losses.
- We highlight the adaptability and generalizability of our framework, suggesting the potential application of an ARO approach to pricing a wide range of catastrophic events, such as wildfires, droughts, extreme weather events.

The structure of the paper is as follows. In Section 2, we review the relevant literature. In Section 3 we introduce the problem and outline the Robust Optimization (RO) and Adaptive Robust Optimizaiton (ARO) framework. In Section 4, we demonstrate the application of our framework through a case study on flood insurance in the United States. We explain the model parameter estimation and details on the machine learning risk prediction model. In Section 4, we discuss results from our models against two baselines: historical NFIP premiums and Cumulative Moving Average (CMA) scheme. Finally, we draw conclusions in Section 5.

4.2 Literature Review

This work is related to the catastrophe modeling literature, often abbreviated at CAT modeling. It is a pivotal method for assessing and managing risks associated with extreme events employed by insurance companies. [75] offers a comprehensive analysis of how catastrophe models are employed for risk assessment and management purposes. Most CAT models are proprietary, with AIR Worldwide, Risk Management Solutions (RMS), and EQECAT being the major players in the private sector; and the open-source HAZUS model developed by FEMA [76]. One major challenge of the CAT modeling approach lies in the lack of historical data due to the rarity of the events, and thus standard actuarial techniques fail to capture tail-event risks. In addition, catastrophe modeling depends on scenario simulations, which can be both numerous and lead to largely varying outcomes. How to combine different"what-if" scenarios remains a challenge to decision makers [77]. Our paper illustrates the promise of an optimization-based approach to model uncertain losses and climate risks directly, thus offering transparency and offer greater robustness against rare events.

Our work is also related to the literature of disaster management. Many works have focused the problem of general resource allocations to different programs or regions under a given budget constraint [78]–[80]. [81] discusses fund allocation for flash flood reduction, and [82] further incorporates the use of insurance premiums as a source of funding. As highlighted by [77], the critical issue in insurance policy lies in the need for decision-making robustness in the face of climate change's uncertainties. There is also growing work on using robust optimization in disaster relief management to deal with uncertainties [83], [84]. Few works discuss the use of optimization for catastrophe insurance pricing directly, [85] applies stochastic optimization to the Dutch Flood insurance scheme, and [86] discusses an integrated catastrophe management approach by incorporating CAT models with stochastic optimization methods. Our study fills in the gap by proposing a robust optimization framework by modeling uncertain losses and integrating machine learning forecast risks to address the increasing unpredictability of weather events driven by climate change.

Finally, this chapter broadly belongs to the climate finance literature, and see [87] for a comprehensive overview. [88], [89] examine the public policy implications in funding climate change adapation. It is critical to design new tools to financially manage weather risks. [90] discusses the role of insurance sector in decreasing the vulnerability of human and natural systems. [91], [92] propose the use of municipal bonds to finance natural disasters. [93] considers insuring climate change induced risks across broad business spectrum. Our paper adds to the literature by proposing a new design of the weather-related insurance contract to manage weather risk.

4.3 Optimization Framework for Catastrophe Insurance

In this section, we introduce the problem and present the Robust Optimization (RO) and Adaptive Robust Optimizaiton (ARO) frameworks.

4.3.1 Robust Optimization

Robust Optimization (RO) is a useful methodology for handling optimization problems with uncertain data [94]. It has been applied to address uncertainties in various fields, including operations research, engineering, and finance. Consider a generic linear programming problem

$$\min_{\mathbf{x}} \left\{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b} \right\},$$

where $\mathbf{c} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^m \times \mathbb{R}^n$. Robust Optimization addresses the problem where data (c, $\mathbf{A}, \mathbf{b}$) are uncertain, but are known to reside in an uncertainty set $\mathcal{U}$. Based on prior information or assumptions, we construct the uncertainty set $\mathcal{U}$ to express the uncertainties in data. We are addressing a family of problems for each realization of $(\mathbf{c}, \mathbf{A}, \mathbf{b}) \in \mathcal{U}$. Therefore, we can reformulate the problem into its Robust Counter part

$$\min_{\mathbf{x}} \left\{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b} \quad \forall (\mathbf{c}, \mathbf{A}, \mathbf{b}) \in \mathcal{U} \right\}.$$

4.3.2 Nominal Formulation

We consider setting insurance premiums for N locations for the insurance period of T years, which we denote with variable $p_{i,t}$, where $i = 1, \ldots, N,\ t = 1, \ldots, T$. We are given historical losses for each location for each year in the past T_0 years, which we denote with $\bar{l}_{i,t}$, where $i = 1, \ldots, N,\ t = 1, \ldots, T_0$. We assume that the future losses for each location i n the insurance period of T years with $l_{i,t},\ i = 1, \ldots, N,\ t = 1, \ldots, T$. Note that this quantity is unknown, but we assume the knowledge of it to introduce a simple deterministic LP formulation to introduce the basic requirements and set up the generic framework. In the next subsections, we expand on how to model this uncertain quantity through Robust Optimization framework.

We formulate an LP model to set insurance premium price. The objective function is to minimize the overall premium collected, or the least required premium needed. In addition, to model the consumer behavior that as we increase premium price less consumers are willing to purchase the insurance product, we introduce a damping function $f : \mathbb{R} \rightarrow \{0, 1\}$, a monotonically decreasing function representing decline in demand due to higher premiums. Details of the choice of the damping function will be discussed later in Section 4.2.2. The overall objective is as follows

$$\min_{p_{i,t}} \sum_{i=1}^{N} \sum_{t=1}^{T} f(p_{i,t}) * p_{i,t}. \tag{4.1}$$

We require the premium price to cover projected losses with an additional buffer amount, denoted by δ, which is set at constant for each location. Thus we require

$$\sum_{t=1}^{T} f(p_{i,t}) * p_{i,t} - \sum_{t=1}^{T} f(p_{i,t}) * l_{i,t} \geq \delta, \quad i \in [N]. \tag{4.2}$$

In addition, we impose a constraint to require premiums collected over consecutive years to vary slowly, in order to prevent drastic changes in insurance premiums

$$|p_{i,t} - p_{i,t-1}| \leq \gamma_1, \quad i \in [N], \quad t \in [T]. \tag{4.3}$$

Variables $p_{i,t}$ should be positive, for each location i for each period t

$$p_{i,t} \in \mathbb{R}^{+}, \quad i \in [N], \quad t \in [T]. \tag{4.4}$$

4.3.3 Robust Optimization Formulation

In the nominal formulation, we assume the knowledge of projected loss for each of the future period. However, this quantity is unknown and highly uncertain. In this subsection, we expand on how to construct uncertainty sets to describe this quantity.

The overall optimization formulation is the same as before, except for constraint 4.3, where we require the inequality to hold for all losses in the uncertainty set

$$\sum_{t=1}^{T} f(p_{i,t}) * p_{i,t} - \sum_{t=1}^{T} f(p_{i,t}) * l_{i,t} \geq \delta, \quad i \in [N], \quad \forall l_{i,t} \in \mathcal{U}. \tag{4.5}$$

We propose two uncertainty sets to model the uncertainties of the future losses: with Central Limit Theorem (CLT) and with Machine Learning driven risks.

Uncertainty Set from Central Limit Theorem

Based on the assumption that for each specific location, future flooding losses follow the distribution of historical losses. We adopt the central limit theorem (CLT) to form the uncertainty set as discussed in [95]. Formally, for a fixed location i, $L_{i,t}, t \in [T]$ are independent, identically distributed random variables with mean $\bar{l}_i$ and standard deviation $\bar{\sigma}_i$, where $\bar{l}_i, \bar{\sigma}_i$ are historical mean and standard deviation for location i. We assume the uncertain quantities $L_{i,t}$ take values such that

$$\left| \sum_{t=1}^{T} L_{i,t} - T \cdot \bar{l}_i \right| \leq \gamma_2 \cdot \sigma_i \sqrt{n}, \tag{4.6}$$

where γ_2 is a small constant to denote how close future losses distribution should deviation from the normal distribution. In other words, we describe the uncertain quantities $L_{i,t}$ as values in the uncertainty set

$$\mathcal{U}_i^{CLT} = \left\{ (l_{i,1}, \ldots, l_{i,T}) : \frac{|\sum_{t=1}^{T} l_{i,t} - \bar{l}_i|}{\sigma_i \sqrt{T}} \leq \gamma_2 \right\}, \tag{4.7}$$

where $\bar{l}_i, \sigma_i$ can be computed for each location i using historical data. The larger we set γ_2, the more conservative the optimization model, and higher premiums will be. As a remark, we derive one uncertainty set for each location using the historical mean and standard deviation for that location to acomodate different flooding risk profiles. We derive the uncertainty sets separately for each location $\mathcal{U}_i^{CLT}$.

Uncertainty Set from Machine Learning Risk Models

In addition, suppose we have some information on how future losses should be, which could be informed by risk models. We can formulate such model predictions as uncertainty sets. One way of obtaining such risk prediction is through machine learning models, as recent advances in ML models demonstrate capabilities for models to predict accurate multi-year forecasts. In this work, we build machine learning models to obtain flooding risks, see greater details in the next section. Nevertheless, one can obtain such risk predictions from physical-based models, or other alternative approaches.

We define a major flood event to be flood incurring loss over a threshold loss level Θ. Suppose we have model predictions for the risk that at location i, the probability of having one flood event in the next k years to be $q_{i,k}$. Then we can express the model prediction as

$$\mathbb{P}\left(\sum_{t=1}^{k} z_{i,t} = 1\right) = q_{i,k}, \tag{4.8}$$

where $z_{i,t} \in \{0,1\}$ are binary variables denoting if there is a major flood at location i at time t. By modeling such, we assume having more than one major event is negligible for reasonable k. Since model predictions are probabilistic predictions which can have errors, and we want to be conservative and protect against suffering potential huge losses. Thus, assuming actual incidence of having a major flood is close to the model predictions, we can express $z_{i,t}$ as random variables taking values in an uncertainty set as

$$\mathcal{U} = \left\{ z_{i,t} : |\sum_{t=1}^{k} z_{i,t} - q_{i,k}| \leq \epsilon, \quad \sum_{t=1}^{k} z_{i,t} \leq 1 \right\}, \tag{4.9}$$

where the constant ϵ is a parameter to indicate how close we believe the model predictions are to actual probabilities. The larger the value, the less confident and more conservative our model will be. Therefore, linking variables $z_{i,t}$ to $l_{i,t}$, and assuming the future losses will be bounded by the expected value coming from suffering a major flood, we model the uncertainty set of the future loss for each location i as follows

$$\mathcal{U}_i^{ML} = \left\{ z_{i,t}, l_{i,t} : \sum_{t=1}^{k} l_{i,t} \leq \Theta \cdot \sum_{t=1}^{k} z_{i,t}, \quad |\sum_{t=1}^{k} z_{i,t} - q_{i,k}| \leq \epsilon, \quad \sum_{t=1}^{k} z_{i,t} \leq 1 \right\}. \tag{4.10}$$

The Robust Counterpart

Combining uncertainty sets from Central Limit Theorem and from Machine Learning risk models, the robust optimization formulation of the problem is as follows

$$\begin{aligned}
&\min_{p_{i,t}} \sum_{i=1}^{N} \sum_{t=1}^{T} f(p_{i,t}) * p_{i,t} \\
&\quad \sum_{t=1}^{T} f(p_{i,t}) * p_{i,t} - \sum_{t=1}^{T} f(p_{i,t}) * l_{i,t} \geq \delta, \quad \forall l_{i,t}, z_{i,t} \in \mathcal{U}_i^{CLT}, l_{i,t} \in \mathcal{U}_i^{ML}, \quad i \in [N], \\
&\quad ||p_{i,t} - p_{i,t-1}|| \leq \gamma_1, \quad i \in [N], t \in [T], \\
&\quad p_{i,t} \in \mathbb{R}^+, z_{i,t} \in \{0,1\}, \quad i \in [N], t \in [T].
\end{aligned} \tag{4.11}$$

Since the uncertain occurs only in one constraint, we can write Problem 4.11 as a min-max problem

$$\begin{aligned}
\min_{p_{i,t}} \max_{l_{i,t}, z_{i,t} \in \mathcal{U}} & \sum_{i=1}^{N} \sum_{t=1}^{T} f(p_{i,t}) * p_{i,t} \\
& \sum_{t=1}^{T} f(p_{i,t}) * p_{i,t} - \sum_{t=1}^{T} f(p_{i,t}) * l_{i,t} \geq \delta, \quad i \in [N], \\
& ||p_{i,t} - p_{i,t-1}|| \leq \gamma_1, \quad i \in [N], t \in [T], \\
& l_{i,t} \in \mathcal{U}_i^{CLT}, \quad i \in [N] \\
& l_{i,t}, z_{i,t} \in \mathcal{U}_i^{ML}, \quad i \in [N], \\
& p_{i,t} \in \mathbb{R}^+, z_{i,t} \in \{0,1\} \quad i \in [N], t \in [T].
\end{aligned} \tag{4.12}$$

Next, we consider the inner problem

$$\begin{aligned}\max_{l_{i,t}, z_{i,t} \in \mathcal{U}} & \sum_{i=1}^{N} \sum_{t=1}^{T} f(p_{i,t}) * p_{i,t} \\ & \sum_{t=1}^{T} f(p_{i,t}) * p_{i,t} - \sum_{t=1}^{T} f(p_{i,t}) * l_{i,t} \geq \delta, \quad i \in [N], \\ & l_{i,t} \in \mathcal{U}^{CLT}, \\ & l_{i,t}, z_{i,t} \in \mathcal{U}^{ML}, \\ & p_{i,t} \in \mathbb{R}^{+}, z_{i,t} \in \{0,1\} \quad i \in [N], t \in [T].\end{aligned} \tag{4.13}$$

For the inner problem, we can treat $p_{i,t}$ as constants, thus the inenr problem can be simplified to

$$\begin{aligned}\max_{l_{i,t}, z_{i,t} \in \mathcal{U}} & \sum_{i=1}^{N} \sum_{t=1}^{T} l_{i,t} \\ & l_{i,t} \in \mathcal{U}^{CLT}, \\ & l_{i,t}, z_{i,t} \in \mathcal{U}^{ML}, \\ & p_{i,t} \in \mathbb{R}^{+}, z_{i,t} \in \{0,1\} \quad i \in [N], t \in [T].\end{aligned} \tag{4.14}$$

Note that the uncertainty set is composed of two separate uncertainty sets $\mathcal{U}^{CLT}$ and $\mathcal{U}^{ML}$. We can thus decompose the inner problem into two subproblems, and the objective of the original problem takes the maximum of the two subproblems. Solving each subproblem separately, we can then plug back the analytical solution from each subproblem back to the original problem.

Proposition. *The overall min-max problem is equivalent to*

$$\begin{aligned}
&\min_{p_{i,t}} \sum_{i,t}^{T} f(p_{i,t}) * p_{i,t} \\
&\sum_{t=1}^{T} f(p_{i,t}) * p_{i,t} - \frac{1}{T} \sum_{t=1}^{T} f(p_{i,t}) * L_i^{CLT} \geq \delta, \quad \forall i \in [N], \\
&\sum_{t=1}^{k} f(p_{i,t}) * p_{i,t} - \frac{1}{k} \sum_{t=1}^{k} f(p_{i,t}) * L_i^{ML} \geq \delta, \quad \forall i \in [N], \\
&||p_{i,t} - p_{i,t-1}|| \leq \gamma_1, \quad \forall i \in [N], \quad t \in [T],
\end{aligned} \tag{4.15}$$

where

$$L_i^{CLT} = T \cdot \bar{l}_i + \gamma_2 \cdot \sigma_i \sqrt{T}, \tag{4.16}$$

$$L_i^{ML} = \Theta \cdot \min\{1, q_{i,k} + \epsilon\}. \tag{4.17}$$

As a remark, we take the average "damping" over losses $l_{i,t}$ because we solve for the optimal $\sum_t l_{i,t}$.

Proof. Consider the first subproblem

$$\begin{aligned}
&\max_{l_{i,t}} \sum_{i=1}^{N} \sum_{t=1}^{T} l_{i,t} \\
&|\sum_{t=1}^{T} l_{i,t} - T \cdot \bar{l}_i| \leq \gamma_2 \cdot \sigma_i \sqrt{T} \quad \forall i \in [N], \\
&l_{i,t} \in \mathbb{R}^+,
\end{aligned} \tag{4.18}$$

looking at the constraint, we can take out $|\cdot|$ since we are maximizing over $l_{i,t}$, and rearranging terms

$$\sum_{t=1}^{T} l_{i,t} \leq T \cdot \bar{l}_i + \gamma_2 \cdot \sigma_i \sqrt{T} \quad \forall i \in [N], \tag{4.19}$$

and optimality is achieved at equality, we can solve for i.e., $l^*_{i,t}$ s.t.

$$\sum_{t=1}^{T} l_{i,t} = T \cdot \bar{l}_i + \gamma_2 \cdot \sigma_i \sqrt{T} \quad \forall i \in [N]. \tag{4.20}$$

Hence we can compute for all location $i \in [N]$, and denote the analytical solution to the first subproblem as L_i^{CLT}

$$L_i^{CLT} := \sum_{t=1}^{T} l^*_{i,t} = T \cdot \bar{l}_i + \gamma_2 \cdot \sigma_i \sqrt{T} \sqrt{(T_1 - T_0)}. \tag{4.21}$$

Consider now the second subproblem

$$\begin{aligned} \max_{l_{i,t} z_{i,t}} & \sum_{i=1}^{N} \sum_{t=1}^{T} l_{i,t} \\ & \sum_{t=1}^{k} l_{i,t} \leq \Theta \cdot \sum_{t=1}^{k} z_{i,t}, \\ & |\sum_{t=1}^{k} z_{i,t} - q_{i,k}| \leq \epsilon, \\ & \sum_{t=1}^{k} z_{i,t} \leq 1, \\ & l_{i,t} \in \mathbb{R}^+, z_{i,t} \in \{0, 1\}. \end{aligned} \tag{4.22}$$

Without loss of generality, we relax $z_{i,t}$ to take continuous value $z_{i,t} \in [0, 1]$, because we can treat $\sum_i^k z_{i,t}$ as one variable taking continuous values in $[0, 1]$. Thus looking at the constraints concerning $z_{i,t}$, and take out $| \cdot |$ since we are maximizing over $z_{i,t}$, we get $z^*_{i,t}$ achieve optimality at

$$\sum_{t=1}^{k} z^*_{i,t} = \min\{1, q_{i,k} + \epsilon\}, \tag{4.23}$$

which gives $l^*_{i,t}$ at optimality at

$$\sum_{t=1}^{k} l^*_{i,t} = \Theta \cdot \sum_{t=1}^{k} z^*_{i,t}. \tag{4.24}$$

Combining with the solution from the first subproblem given by equation 4.21 we have the sum of future loss for each location given as follows

$$\sum_{t=1}^{T} l_{i,t}^{*} = T \cdot \bar{l}_i + \gamma_2 \cdot \sigma_i \sqrt{T}, \tag{4.25}$$

$$\sum_{t=1}^{k} l_{i,t}^{*} = \Theta \cdot \min\{1, q_{i,k} + \epsilon\}, \tag{4.26}$$

where the first equality bounds the entire future period T, the second equality bounds the period depending on machine learning model's risk forecast horizon k. As a remark, we can have multiple risk models for different forecasting horizons. □

4.3.4 Choices of demand damping function f(p)

Recall that to model the behavioral aspect that as insurance premium increase there is less demand for it, we introduce the damping function $f(p) : \mathbb{R} \to \{0, 1\}$ into the constraint. To preserve the convexity property of the overall problem, we model the behavior using piecewise linear functions under the scope of this work. In the next section, we explain in greater detail on the estimation using NFIP data, as well as sensitivity analysis on the choice of the function.

As a remark, as we damp demand, we correspondingly damp the covered loss in the constraint. Since the inner problem gives the the overall loss over the forecasting period T, i.e., L_i^{CLT} gives the maximum loss deduced from the uncertainty set over period T, we have taken the corresponding damping term to be the average over T periods, i.e., $\frac{1}{T} \sum_{t=1}^{T} f(p_{i,t})$.

4.3.5 Adaptive Robust Optimization Formulation

This section discusses how to the RO framework with adaptive robust optimization techniques, enabling premium adjustments based on actual loss experiences. We let the premiums depend on realized losses using affine decision rules, as proposed in Chapter 7 of [95]. This approach not only refines premium pricing accuracy but also ensures a responsive and equitable insurance mechanism against the backdrop of unpredictable catastrophic events. In particular, we let premiums depend on loss from the previous time step as follows

$$p_{i,t} = \begin{cases} \alpha_{i,1}, & \text{for } t = 1 \\ \alpha_{i,t} + \beta_{i,t} \cdot l_{i,t-1}, & \text{for } t = 2, \ldots, T \end{cases} \tag{4.27}$$

where premium for location i at time period t is determined by a linear combination of parameters. Specifically, for the first time period, the premium is set to a base value $\alpha_{i,1}$; for subsequent periods, the premium is adjusted based on the loss $l_{i,t-1}$, experienced in the previous period, with $\alpha_{i,t}$ and $\beta_{i,t}$ new variables to be optimized over.

Throughout this section, we consider the simplified robust optimization problem from the original problem, by dropping the demand-damping to allow the derivation of a robust counter part. As discussed later in the sensitivity analysis in Section 4.5.1, the demand damping term does not materially influence the premium outcome. Thus the problem we consider is given as follows

$$\begin{aligned} \min_{p_{i,t}} & \sum_{i=1}^{N} \sum_{t=1}^{T} p_{i,t} \\ & \sum_{t=1}^{T} p_{i,t} - \sum_{t=1}^{T} l_{i,t} \geq \delta, \quad \forall l_{i,t} \in \mathcal{U}_i^{CLT}, \quad i \in [N], \\ & ||p_{i,t} - p_{i,t-1}|| \leq \gamma_1, \quad i \in [N], \quad t \in [T], \\ & p_{i,t} \in \mathbb{R}^+, z_{i,t} \in \{0,1\}, \quad i \in [N], \quad t \in [T]. \end{aligned} \tag{4.28}$$

Recall that this optimization problem aims to minimize the total premiums over all locations and time periods while ensuring that the cumulative premium exceeds the cumulative losses by at least a margin of δ. Note that in this formulation, we impose slowly varying constraint on $\alpha_{i,t}$, instead of $p_{i,t}$ as in the previous formulation, because $p_{i,t}$ is now depending on uncertain variables $l_{i,t}$.

Similar as before, noticing that the constraints for each location i is independent of other locations, and thus minimizing the aggregated premium is the same as minimizing the premium for each location. Therefore we can decompose the problem by solving for each location i the ARO problem independently. In addition, noticing that the objective function now depends on uncertain variable $l_{i,t}$, we therefore use the epigraph formulation as discussed in Chapter 2 of the book [95] to move all variables containing uncertain variables into constraints. We arrive at the following ARO formulation for one location i

$$\min_{\alpha_{i,t},\beta_{i,t}} \ \Omega \tag{4.29}$$

$$\sum_{t=1}^{T} p_{i,t} \leq \Omega, \quad \forall l_{i,t} \in \mathcal{U}_i^{CLT}, \tag{4.30}$$

$$\sum_{t=1}^{T} p_{i,t} - \sum_{t=1}^{T} l_{i,t} \geq \delta, \quad \forall l_{i,t} \in \mathcal{U}_i^{CLT}, \tag{4.31}$$

$$||\alpha_{i,t} - \alpha_{i,t-1}|| \leq \gamma_3, \quad t = 2, \dots, T, \tag{4.32}$$

$$||\beta_{i,t} - \beta_{i,t-1}|| \leq \gamma_4, \quad t = 2, \dots, T, \tag{4.33}$$

$$p_{i,t} \geq 0, \quad t = 1, \dots, T, \quad \forall l_{i,t} \in \mathcal{U}_i^{CLT}., \tag{4.34}$$

where $p_{i,t}$ is a quantity that depends on the uncertain losses given by equation 4.27. Next, we derive the robust counter part by taking the RC for each constraint, that depends on uncertain variable $l_{i,t}$, independently.

First, consider constraint give by inequality 4.30, substituting $p_{i,t}$ with $\alpha_{i,t}, \beta_{i,t}$ which now depends on the uncertain variable $l_{i,t}$. We rewrite the constraint as follows

$$\sum_{t=1}^{T} \alpha_{i,t} + \sum_{t=2}^{T} \beta_{i,t} \cdot l_{i,t-1} \leq \Omega, \quad \forall l_{i,t} \in \mathcal{U}_i^{CLT}, \tag{4.35}$$

which is is equivalent to

$$\sum_{t=1}^{T} \alpha_{i,t} + \max_{l_{i,t} \in \mathcal{U}_i^{CLT}} \left\{ \sum_{t=2}^{T} \beta_{i,t} \cdot l_{i,t-1} \right\} \leq \Omega. \tag{4.36}$$

Consider now the inner maximization problem using the explicit expression for $\mathcal{U}_i^{CLT}$, we have a LP problem in $l_{i,t}$

$$\begin{aligned}
&\max_{l_{i,t}} \sum_{t=2}^{T} \beta_{i,t} \cdot l_{i,t-1} \\
&\sum_{t=1}^{T} l_{i,t} \leq \bar{T}_i + \gamma_2 \cdot \sigma_i \cdot \sqrt{T}, \\
-&\sum_{t=1}^{T} l_{i,t} \leq -\bar{T}_i + \gamma_2 \cdot \sigma_i \cdot \sqrt{T}.
\end{aligned} \tag{4.37}$$

The inner problem thus satisfies strong duality. Notice that the uncertainty set is given in polyhedron form, we can thus take the dual by introducing dual variables s_1^1, s_2^1, and arrive at the following form

$$\begin{aligned}
&\min_{s_1^1, s_2^1} \sum_{j=1}^{2} c_j s_j^1 \\
&s_1^1 - s_2^1 \geq \beta_t, \quad \forall t = 2, \ldots, T, \\
&s_1^1 - s_2^1 \geq 0, \\
&s_1^1, s_2^1 \geq 0.
\end{aligned} \tag{4.38}$$

By strong duality, the inner maximization problem has the same objective value of the dual minimization problem. Using the dual expression, constraint 4.30 becomes

$$\begin{aligned}
\sum_{t=1}^{T} \alpha_{i,t} + \min_{s_1^1, s_2^1} \sum_{j=1}^{2} c_j s_j \leq \Omega& \\
s_1^1 - s_2^1 \geq \beta_{i,t}, \quad \forall t = 2, \ldots, T,& \\
s_1^1 - s_2^1 \geq 0,& \\
s_1^1, s_2^1 \geq 0.&
\end{aligned}$$

Note that we can take away the minimization term because if any feasible s_1^1, s_2^1 satisfies this constraint, the minimum also does.

Second, consider constraint give by inequality 4.31, which ensurs premiums cover losses

$$\sum_{t=1}^{T} p_{i,t} - \sum_{t=1}^{T} l_{i,t} \geq \delta, \quad \forall l_{i,t} \in \mathcal{U}_i^{CLT}. \tag{4.39}$$

Multiplying both sides by -1, we have

$$\sum_{t=1}^{T} l_{i,t} - \sum_{t=1}^{T} p_{it} \leq -\delta. \tag{4.40}$$

Substituting $l_{i,t}$ with $\alpha_{i,t}$ and $\beta_{i,t}$

$$\sum_{t=1}^{T} l_{i,t} - \left[\sum_{t=1}^{T} \alpha_{i,t} + \sum_{t=2}^{T} \beta_{i,t} l_{i,t-1} \right] \leq -\delta, \tag{4.41}$$

re-arranging terms to collect all the uncertain terms $l_{i,t}$

$$-\sum_{t=1}^{T} \alpha_{i,t} + \left[l_{i,1} + \sum_{t=2}^{T} (1 - \beta_{i,t}) l_{i,t} \right] \leq -\delta, \tag{4.42}$$

we finally arrive at

$$-\sum_{t=1}^{T} \alpha_{i,t} + \max_{l_{i,t} \in \mathcal{U}_i^{CLT}} \left\{ l_{i,1} + \sum_{t=2}^{T} (1 - \beta_{i,t}) l_{i,t} \right\} \leq -\delta. \tag{4.43}$$

Consider now the inner maximization problem, which is once again a LP in $l_{i,t}$

$$\begin{aligned} \max_{l_{i,t}} \quad & l_{i,1} + \sum_{t=2}^{T} (1 - \beta_{i,t}) \cdot l_{i,t} \\ \sum_{t=1}^{T} l_{i,t} & \leq \bar{T}_i + \gamma_2 \cdot \sigma_i \cdot \sqrt{T}, \\ -\sum_{t=1}^{T} l_{i,t} & \leq -\bar{T}_i + \gamma_2 \cdot \sigma_i \cdot \sqrt{T}. \end{aligned} \tag{4.44}$$

Similar as before, by strong duality, the dual is given as a minimization problem in dual variables s_1^2, s_2^2

$$\begin{aligned}
&\min_{s_1^2,s_2^2} \sum_{j=1}^{2} c_j s_j^2 \\
&\quad s_1^2 - s_2^2 \geq 1 - \beta_{i,t}, \quad \forall t = 2, \dots, T, \\
&\quad s_1^2 - s_2^2 \geq 1, \\
&\quad\; s_1^2, s_2^2 \geq 0.
\end{aligned} \tag{4.45}$$

Therefore constraint 4.31 becomes

$$\begin{aligned}
&-\sum_{t=1}^{T} \alpha_{i,t} + \sum_{j} c_j s_j^2 \leq -\delta, \\
&s_1^2 - s_2^2 \geq 1 - \beta_{i,t}, \quad \forall t = 2, \dots, T, \\
&s_1^2 - s_2^2 \geq 1, \\
&s_1^2, s_2^2 \geq 0.
\end{aligned} \tag{4.46}$$

Finally, we consider the positivity constraint as given by inequality 4.34

$$p_{i,t} \geq 0, \quad \forall t \in [T], \quad \forall l_{i,t} \in \mathcal{U}^{CLT} \tag{4.47}$$

substituting $p_{i,t}$ with $\alpha_{i,t}, \beta_{i,t}$, the constraint is equivalent to

$$\alpha_{i,t} + \beta_{i,t} \cdot l_{i,t-1} \geq 0, \quad \forall t \in [T], \quad \forall l_{i,t} \in \mathcal{U}^{CLT} \tag{4.48}$$

Since for each time t, we have one separate constraint. Therefore, considering this constraint for some t

$$\alpha_{i,t} + \beta_{i,t} \cdot l_{i,t-1} \geq 0, \quad \forall l_{i,t} \in \mathcal{U}^{CLT}.$$

Multiplying both sides by -1 and rearranging terms, we have

$$-\alpha_{i,t} + \max_{l_{i,t} \in U} \{-\beta_{i,t} \cdot l_{i,t-1}\} \leq 0.$$

Similar as before, we consider the inner maximization problem

$$\max_{l_{i,t} \in U} -\beta_{i,t} \cdot l_{i,t-1}, \tag{4.49}$$

which has strong duality, with dual given for each t in dual variables $s_1^{3,t}, s_2^{3,t}$ as

$$\begin{aligned}
&\min_{s_1^{3,t}, s_2^{3,t}} \sum_j c_j s_j^{3,t} \\
&s_1^{3,t} - s_2^{3,t} \geq -\beta_{i,t}, \\
&s_1^{3,t} - s_2^{3,t} \geq 0, \\
&s_1^{3,t}, s_2^{3,t} \geq 0.
\end{aligned} \tag{4.50}$$

Thus the positivity constraint for some t becomes

$$\begin{aligned}
&\alpha_{i,t} + \sum_j c_j s_j^{3,t} \leq 0 \\
&s_1^{3,t} - s_2^{3,t} \geq -\beta_{i,t}, \\
&s_1^{3,t} - s_2^{3,t} \geq 0, \\
&s_1^{3,t}, s_2^{3,t} \geq 0.
\end{aligned} \tag{4.51}$$

Thus, the overall robust counter part of the ARO formulation is given as follows

$$\min_{\alpha_{i,t}, \beta_{i,t}} \Omega \tag{4.52}$$

with epigraph constraint

$$\begin{aligned}
&\sum_{t=1}^{T} \alpha_{i,t} + \min_{s_1^1, s_2^1} \sum_{j=1}^{2} c_j s_j \leq \Omega, \\
&s_1^1 - s_2^1 \geq \beta_{i,t}, \quad \forall t = 2, \ldots, T, \\
&s_1^1 - s_2^1 \geq 0, \\
&s_1^1, s_2^1 \geq 0.
\end{aligned} \tag{4.53}$$

With loss coverage constraint

$$\begin{aligned}
&-\sum_{t=1}^{T}\alpha_{i,t} + \sum_{j} c_j s_j^2 \leq -\delta, \\
&s_1^2 - s_2^2 \geq 1 - \beta_{i,t}, \quad \forall t = 2, \ldots, T, \\
&s_1^2 - s_2^2 \geq 1, \\
&s_1^2, s_2^2 \geq 0.
\end{aligned} \tag{4.54}$$

With positivity constraint

$$\begin{aligned}
&\alpha_{i,t} + \sum_{j} c_j s_j^{3,t} \leq 0, \\
&s_1^{3,t} - s_2^{3,t} \geq -\beta_{i,t} \\
&s_1^{3,t} - s_2^{3,t} \geq 0, \\
&s_1^{3,t}, s_2^{3,t} \geq 0.
\end{aligned} \tag{4.55}$$

With slowly varying constraint

$$\begin{aligned}
&||\alpha_{i,t} - \alpha_{i,t-1}|| \leq \gamma_3, \quad t = 2, \ldots, T, \\
&||\beta_{i,t} - \beta_{i,t-1}|| \leq \gamma_4, \quad t = 2, \ldots, T.
\end{aligned} \tag{4.56}$$

4.4 Case Study for US National Flood Insurance

In this section, we demonstrate the application of our framework through a case study on flood insurance in the United States. We explain the model parameter estimation and details on the training of machine learning risk predictions.

4.4.1 Data

We used two redacted datasets from the National Flood Insurance Program (NFIP) on claims and policies respectively. Both data sets are created and maintained by Federal Emergency Management Agency (FEMA). The claims transaction data provide details on NFIP claims transactions across from all states in the United States [96]. This dataset consist of 2570089 lines of claim transactions ranging, dated from 1970. Due to limited data availability in the early years, we included data from 1975 to 2022.

For this study, we aggregate data into state level on an annual basis, and have used the following features: date ('dateOfLoss'), state, claim amount ('amountPaidOnBuildingClaim'). Additionally, data from 'MP' (Northern Mariana Islands), 'AS' (American Samoa), 'GU' (Guam), and 'DC' (District of Columbia) have been omitted due to their limited data records. As a result, our cleaned dataset encompasses information from 52 jurisdictions over 48 years, including 50 U.S. states, alongside two territories recognized as island states: the U.S. Virgin Islands and Puerto Rico, enhancing the geographical breadth of our study..

The policy premium data contains 228664 lines of data, covering 54 states and 6704 unique zip codes, and contains policies from 2009 to 2022 [97]. We use the policy data as a benchmark to compare model performance between last ten years of testing period from 2013 to 2022. Similar to the claims data, we aggregated data into state level on an annual basis, and have used the following features: state ('propertyState'), date ('policyTeminationDate') and premium ('totalInsurancePremiumOfThePolicy').

In this work, we consider setting insurance premium at state level on an annual basis. We consider setting the premium for the last 10 years, from 2013 to 2022. We trained machine learning models using data from 1975 to 2012, to derive risks in the testing period from 2013 to 2022. We used machine learning derived risks as parameter input for the optimization model, and drive premiums using the RO framework.

4.4.2 Optimization Model Parameter Estimation

Recall from equation 4.16, to compute L_i^{CLT} we need to compute the historical mean and variance for each state. We use the training data from 1975 to 2012 to estimate optimization model parameter to avoid data spoilage in the testing data set. We compute the historical mean and standard deviation for all states on an annual basis. Table B.1 in the appendix exhibits the historical mean and variance for the top 10 most costly states.

In this work, we model the demand sensitivity to insurance premium through a piece-wise linear demand function. We estimate the decline rate using historical data from several states. We include Figure B.1 and B.2 in the appendix showing the scatter plot of number of policy holders in a year against the mean policy premium of that state at that year. Different states have different degrees of sensitivity to price, but in general we observe a downward trend of decline in policy holder number as a function of increased price. For the illustrative purpose of this work, we do not specify different sensitivity in different states, but use the same demand damping function across all states.

We use the following piece-wise linear demand damping function to model demand damping

$$f(p) = \begin{cases} 1, & \text{if p} \leq P_0 \text{ ;} \\ 1 - m \cdot (p - P_0), & \text{if p} \geq P_0 \text{ and } f(p) \geq c_{min} \text{ ;} \\ \mathrm{c}_{min}, & \text{otherwise.} \end{cases} \tag{4.57}$$

where P_0 is the minimum premium at which demand damping starts to occur, and c_{min} is the minimum fraction of demand. In this work, we choose P_0 to be $\frac{1}{10} \cdot P_{hist}^{max}$, a fraction of maximum historical premium ever charged. We choose c_{min} to be 0.2 representing at least 20% of the total demand is preserved regardless of price. We experimented with different demand damping rate: $m = 1/P_{hist}^{max}$. We include Figure B.3 in the appendix to illustrate several choices of the demand damping curve.

4.4.3 Machine Learning Model

The goal of the machine learning model is to obtain the risk measure $q_{i,k}$, denoting the risk of major flooding event occurrence in the next k years at state i, in the machine learning risk uncertainty 4.10. Recall that we denote a major flooding event as the total state-level loss surpassing a certain threshold in the next 1-K years, and obtain the probabilistic prediction result obtained from the binary machine learning task.

Specifically, we train machine learning models to predict future annual losses exceeding specific thresholds, based on both current and past losses, across three different threshold levels and time frames. We train binary classification models to predict the target, with 1 indicating for a particular state at a particular year, the state will suffer an annual loss passing through the threshold Θ in the next 1 to K years. We have experimented with three threshold values, corresponding to 90th, 95th and 99th percentile annual claim amount values across all states over all training data, corresponding to USD 18,558,788, USD 50,688,672 and USD 321,903,271. In addition, we have experimented with three K values, corresponding to 3, 5, 10 years respectively. A detailed explanation of the data processing, feature construction, and model training methodology is provided in the appendix.

Table B.3 record out of sample prediction results using testing data, corresponding to data between 2012 to 2022. We treat data in the testing period on a rolling basis, and we drop the years where we do not have target data, i.e., in year 2019, we predict for K=3 but not for K = 5 or 10. We remark that accuracy is generally higher for longer forecasting horizons. This is likely due to the following reasons: first, longer forecasting horizon lead to higher probability of flood, which leads to more balanced data; second, we have less testing samples. Finally, we use the probabilistic prediction results for each state at each testing year $q_{i,k}$ as input to construct uncertainty sets for the robust optimization model as given by equation 4.17.

4.5 Results

We implement three types of robust optimization models, with linearly decreasing demand damping and machine learning risk forecasts. In addition, we implement the adaptive robust optimization model. We compare results against two baseline policy premiums.

- Historical premiums charged for each state, which is referred to as 'hist'.
- Cumulative moving average loss up to that year, which we refer to as 'CMA'. We

Scores	3 Years		5 Years		10 Years	
	logreg	xgb	logreg	xgb	logreg	xgb
90% Threshold						
auc	0.743	0.776	0.763	0.808	0.767	0.912
f1	0.630	0.665	0.593	0.735	0.621	0.817
accu	0.693	0.675	0.632	0.736	0.623	0.830
accu_bl	0.625	0.706	0.605	0.753	0.641	0.809
precision	0.407	0.457	0.520	0.625	0.658	0.777
recall	0.437	0.793	0.362	0.901	0.500	0.967
95% Threshold						
auc	0.818	0.871	0.818	0.897	0.794	0.921
f1	0.684	0.673	0.700	0.744	0.764	0.763
accu	0.856	0.781	0.808	0.792	0.830	0.774
accu_bl	0.665	0.742	0.699	0.829	0.736	0.820
precision	0.300	0.306	0.368	0.460	0.595	0.560
recall	0.391	0.688	0.516	0.891	0.500	0.938
99% Threshold						
auc	0.920	0.945	0.922	0.956	0.952	0.992
f1	0.704	0.687	0.737	0.771	0.835	0.906
accu	0.910	0.950	0.909	0.943	0.925	0.962
accu_bl	0.809	0.646	0.786	0.731	0.958	0.979
precision	0.253	0.215	0.313	0.380	0.556	0.714
recall	0.696	0.304	0.640	0.480	1.000	1.000

Table 4.1: Consolidated out-of-sample accuracy for percentile threshold predictions at 90%, 95%, and 99%, across different time horizons using logistic regression (logreg) and XGBoost (xgb) models.

compute the cumulative moving average loss as follows

$$p_{i,t}^{CMA} = \frac{1}{t} \sum_{t'=0}^{t} {}_{i,t'}. \tag{4.58}$$

An overall summary of models implemented can be found in Table 4.2 below.

We evaluate performance during the testing period, the last ten years of available data in the NFIP data set between 2013 to 2022. For the rest of the section, we choose γ_1 to be 50000, and δ to be 10000. We also undertake a sensitivity analysis to assess the impact of our model's parameter selections. Overall, we evaluate our model performance using the following two criteria:

Model Name	**Description**
Hist	Uses the historical level of insurance premium collected by the NFIP program, relying on past data to set future premiums without adjustment for future uncertainties.
CMA	Employs a cumulative moving average to compute insurance premiums using historical losses, aiming for stability through past data analysis.
RO1	Implements robust optimization with a CLT (Central Limit Theorem) uncertainty set, focusing on mitigating risk within a specific range of uncertainty based on statistical theory.
RO2	Extends robust optimization by incorporating both CLT and ML (Machine Learning) uncertainty sets, aiming for a comprehensive approach to risk management by leveraging advanced analytics.
ARO	Adaptive robust optimization that dynamically adjusts insurance premiums in response to changing conditions and uncertainties, optimizing strategies over time to minimize risk.

Table 4.2: Summary of models for insurance premium calculation

- Effectiveness: to evaluate the effectiveness of models to cover losses at a state level. We count the number of insolvent states, i.e., the cumulative premium collected over the testing period does not cover cumulative claim losses.

- Efficiency: to ensure models are charging reasonable levels of premiums to cover losses, we evaluate the overall surplus (or deficit) level as well as the absolute deviation from actual losses to evaluate models' capabilities in realistically assessing risks.

4.5.1 Sensitivity Analysis

We examine the model sensitivity to parameter choices in our model. Specifically, we examine the effect of the value of γ_2 and the demand damping rate m. Recall that γ_2 is the parameter controlling how conservative the CLT uncertainty set given by equation 4.16, and the demand damping rate controls how fast demand declines in response to increase in price given by equation 4.57. To examine the overall performance of the premium, we compute the cumulative surplus S across all states over the testing period as follows

$$S(\gamma_2) = \sum_{i=1}^{N} \sum_{t=1}^{T1} p_{i,t} - \sum_{i=1}^{N} \sum_{t=1}^{T1} l_{i,t}^{act}. \tag{4.59}$$

Figure 4.2 shows the level of surplus as a function of different γ_2, with different demand damping rate. The two dotted line shows the constant surplus computed by two baselines: using the actual premiums collected during this period and the CMA rule. We observe that both baselines incur a loss over the testing period, with historical premiums resulting in about 20 billion loss, and CMA rule resulting in 8 million loss.

We let γ_2 to take values between 0 and 1.5 at a stepsize of 0.1, and resolve the optimization model at each γ_2 value, and compute the sum of surplus across all states across testing years. $\gamma_2 = 0$ corresponding to convex optimization without uncertainty, and $\gamma_2 = 1.5$ with the maximum degree of uncertainty. As we increase the value of γ_2, the size of the uncertainty set increases, and the model becomes more conservative resulting in surplus as expected. We remark that the surplus breaks even when γ_2 takes value between 0.6 and 0.7. And we observe a smooth increase in surplus as γ_2 increases.

In addition, we experiment with three demand damping rates: no damping, $m_1 = 1/P_{hist}^{max}$, $m_2 = 1/(2 * P_{hist}^{max})$, with m_2 damps twice as fast as m_1. Similar as above, we experiment with varying γ_2 corresponding to the different demand damping rates. We observe that the choice of demand damping is less significant compared with the variation of γ_2.

4.5.2 Effectiveness

Evaluating the effectiveness of insurance schemes in covering losses is crucial, particularly through the lens of insolvency rates. Therefore, we consider the insolvency status at a state level, where the cumulative premium fails to cover the cumulative loss, as an indicator of a scheme's financial resilience and risk management efficiency. Our analysis spans the 2013 - 2022 testing period, corresponding the the last ten years of data, and focuses on the impact of γ_2, a parameter which significantly influences model outcomes. As detailed in Table 4.3,

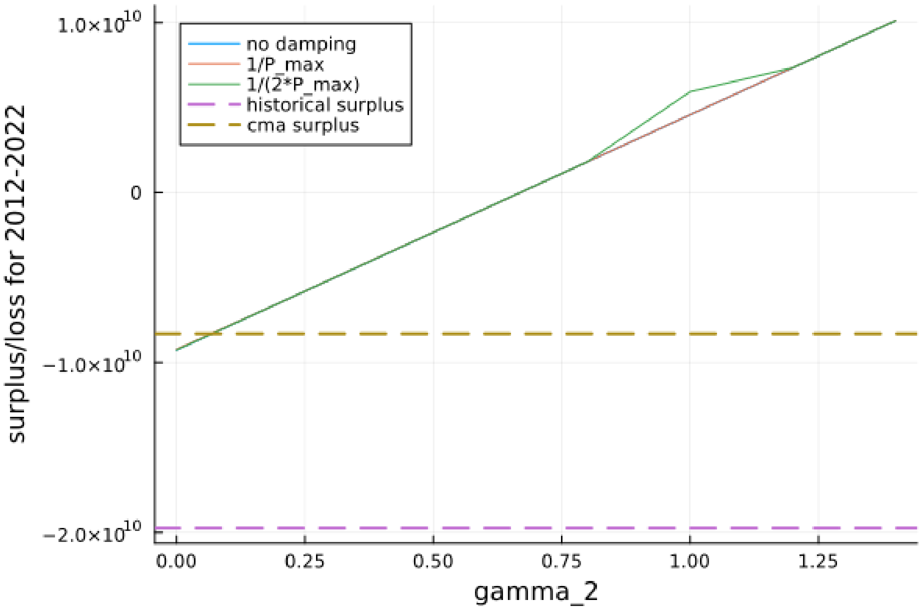


Figure 4.2: Surplus (or loss) computed during 2012 to 2022 across all states when vary different levels of γ_2. Two dotted lines demonstrate the level of surplus from two baseline models: historical surplus calculated from the actual premiums collected, cma surplus is computed using the cumulative moving average.

we explore the effects of varying γ_2 values, from 0 to 2 in increments of 0.2, on the number of insolvent states induced by each model.

Note that the performance of the Historical and CMA schemes is invariant to changes in γ_2. Specifically, the CMA approach results in 36 insolvent states, outperforming the Historical scheme, which results in insolvency across all states. This outcome underscores the limitations of relying solely on past data for setting future premiums. In contrast, both the RO and ARO schemes exhibit increased conservatism—and consequently, fewer insolvent states—with higher γ_2 values. This is because γ_2 directly relates to the size of the CLT uncertainty set, as specified by equation 4.16, where larger γ_2 values necessitate higher premium values to mitigate risk.

This mechanism within the RO framework affords decision-makers the flexibility to adjust the conservatism of their models based on risk tolerance and financial strategy. Among the evaluated schemes, the RO2 model, which integrates both CLT and ML uncertainty sets, consistently achieves the lowest number of insolvent states for a given γ_2 level. This performance is followed closely by the ARO scheme and the RO1 model. Such findings highlight the nuanced balance between risk management and financial sustainability, illustrating the advanced capabilities of RO2 and ARO in navigating the complexities of insurance premium optimization under uncertainty.

γ_2	**ARO**	**RO1**	**RO2**	**CMA**	**Hist**
0.0	36.0	36.0	32.0	36.0	52.0
0.2	30.0	32.0	28.0	36.0	52.0
0.4	26.0	26.0	23.0	36.0	52.0
0.6	25.0	26.0	23.0	36.0	52.0
0.8	22.0	23.0	20.0	36.0	52.0
1.0	21.0	22.0	20.0	36.0	52.0
1.2	19.0	21.0	19.0	36.0	52.0
1.4	16.0	18.0	16.0	36.0	52.0
1.6	15.0	15.0	14.0	36.0	52.0
1.8	13.0	13.0	12.0	36.0	52.0
2.0	12.0	12.0	11.0	36.0	52.0

Table 4.3: The number of insolvent states during the testing period based on different methods. In total we test over 52 states, which are the 50 US states with additionally Puerto Rico and the US virgin islands.

4.5.3 Efficiency

In addition, we consider the efficiency of the models by considering the surplus and deficit level, as well as the absolute deviation from the actual loss incurred. This is to ensure models are not over-charging states, and pricing premiums correctly align with risks.

Table 4.4 exhibits the overall surplus (or deficit) level and the absolute deviation level across all states over the testing period. We compute the overall absolute deviation across all states over the testing period from the actual loss, which is an indicator of how well models are able to trace the risks. We compute AD as a function of γ_2 as follows

$$AD(\gamma_2) = \sum_{i=1}^{N} \sum_{t=1}^{T1} |p_{i,t} - l_{i,t}^{act}|. \tag{4.60}$$

First, we observe that Historical premiums significantly undercharges over the testing period, resulting in 19 billion losses, and CMA rule results in 8 million losses. This suggests historical levels are insufficient to cover future losses. Similar as before, we observe as γ_2 increases, RO and ARO schemes increase the level of conservatism and reaches more surplus. In particular, comparing the two RO models, RO2 scheme is more conservative than RO1, albeit not significantly, as expected. Because RO2 contains the additioanl ML uncertainty set. With same level of γ_2, we observe that ARO achieves the surplus the slower than RO schemes. Especially when γ_2 reaches the level of 1.4, ARO scheme increases premiums much slower than RO schemes, which increases at a constant rate with increasing γ_2. Figure 4.3 illustrates the efficient frontier illustrating the trade-off between the number of insolvent states versus the surplus (or deficit) achieved during the testing period.

The superiority of the ARO scheme over RO scheme is reinforced when looking at the absolute deviation metric. We observe that the level of error using the ARO scheme is more stable than both RO schemes. The stability of ARO scheme makes it desirable for policy-makers, as the scheme offers great robustness against insolvency with the least premium charged when chosen a high level of γ_2. In conclusion, the results underscore the ARO and RO models' superior performance in managing uncertainty and adapting to changing risk profiles, highlighting their potential for enhancing catastrophe insurance premium pricing strategies. Figure 4.4 illustrates the trade-off between the number of insolvent states versus the absolute deviation during the testing period.

γ_2	ARO		RO1		RO2		CMA		Hist	
	S/D	A/D	S/D	A/D	S/D	A/D	S/D	A/D	S/D	A/D
0.0	$-9.23e9$	$2.26e10$	$-9.27e9$	$2.25e10$	$-9.16e9$	$2.26e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
0.2	$-5.88e9$	$2.47e10$	$-6.50e9$	$2.42e10$	$-6.41e9$	$2.42e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
0.4	$-3.56e9$	$2.65e10$	$-3.73e9$	$2.59e10$	$-3.67e9$	$2.59e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
0.6	$-1.80e9$	$2.78e10$	$-0.96e9$	$2.77e10$	$-0.92e9$	$2.77e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
0.8	$-0.46e9$	$2.88e10$	$1.80e9$	$2.95e10$	$1.84e9$	$2.95e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
1.0	$0.55e9$	$2.96e10$	$4.57e9$	$3.13e10$	$4.60e9$	$3.13e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
1.2	$1.41e9$	$3.03e10$	$7.34e9$	$3.32e10$	$7.37e9$	$3.32e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
1.4	$2.16e9$	$3.09e10$	$10.11e9$	$3.53e10$	$10.14e9$	$3.53e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
1.6	$2.78e9$	$3.13e10$	$12.88e9$	$3.74e10$	$12.90e9$	$3.74e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
1.8	$3.32e9$	$3.18e10$	$15.64e9$	$3.96e10$	$15.67e9$	$3.96e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$
2.0	$3.82e9$	$3.21e10$	$18.41e9$	$4.17e10$	$18.43e9$	$4.17e10$	$-8.31e9$	$2.30e10$	$-1.98e10$	$1.98e10$

Table 4.4: Condensed table showing surplus/deficit (S/D) and absolute deviation (A/D) across different γ_2 levels for ARO, RO1, RO2, CMA, and Hist.

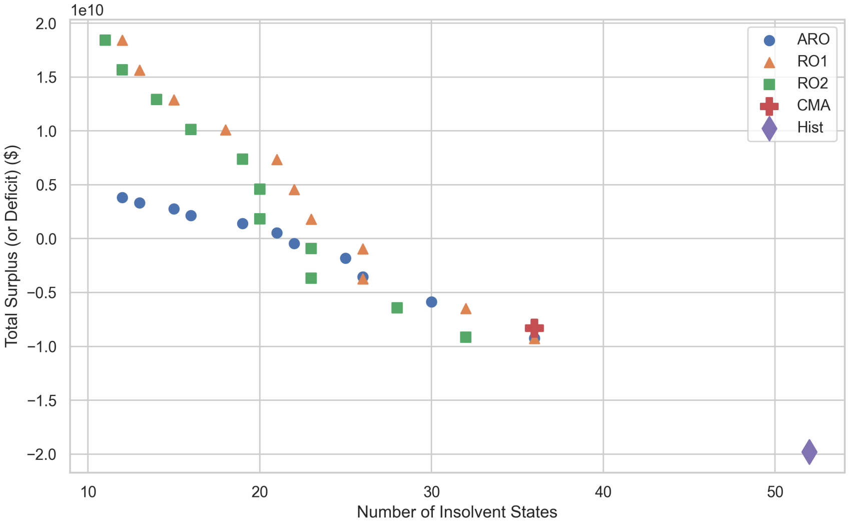


Figure 4.3: Scatter plot visualizing the efficient frontier, showing how different values of γ_2 affect the number of insolvent states (x-axis) and the total surplus (or deficit) (y-axis) computed as the total premium charged minus actual loss over the testing period. Note that CMA and Hist are plotted as static points because their values do not change with varying γ_2 values.

4.6 Conclusion

In conclusion, we present a Robust Optimization (RO) framework to catastrophe insurance premium pricing. We first present a nominal linear optimization formulation to introduce the problem of setting insurance prices for rare catastrophe events, and present a robust opti-

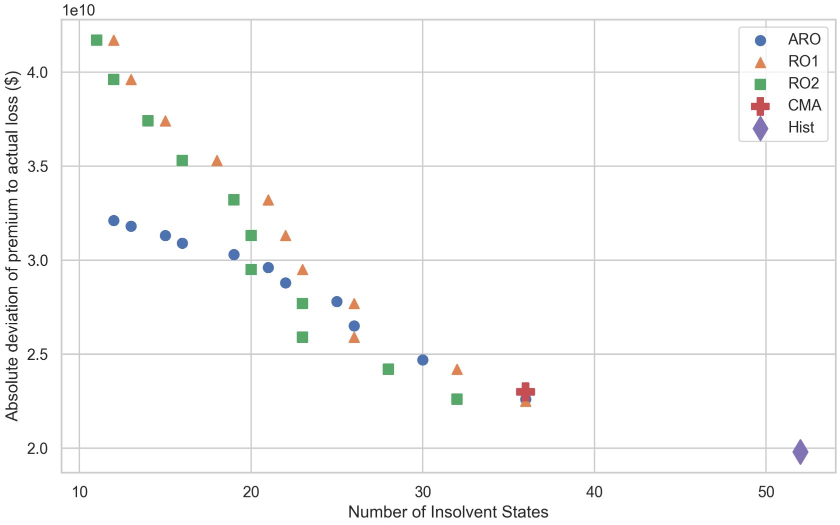


Figure 4.4: Scatter plot visualizing the efficient frontier, showing how different values of γ_2 affect the number of insolvent states (x-axis) and the total absolute deviation (y-axis) computed as the absolute value of the difference between premium charged and actual losses over the testing period. Note that CMA and Hist are plotted as static points because their values do not change with varying γ_2 values. The plot demonstrates that ARO achieves higher efficiency with lower absolute deviation at high γ_2 values, highlighting its better performance under these conditions.

mization formulation with two distinct uncertainty sets. The Central Limit Theorem (CLT) uncertainty set protects against deviations from historical losses, and the Machine Learning (ML) uncertainty set to incorporate predicted risks. We derive the robust counter part by solving the inner problem in closed form, and present a convex optimization re-formulation. In addition, we extned the RO framework to an Adaptive Robust Optimization (ARO) model, enabling premium adjustments based on actual loss experiences through linear decision rules.

We applied the framework to the US flood insurance and evaluate our performance against two baseline benchmarks: the US National Flood Insurnace Program (NFIP) premiums and CMA premiums. We employed historical data from 1975 to 2012 to construct the uncertainty sets and train machine learning models, and we used the last ten years of available NFIP data to evaluate RO insurance scheme against the two benchmarks: the actual historical premium policies and premiums derived from cumulative moving average (CMA) rules. We demonstrate the superiority of an ARO approach in two metrics: effectiveness and efficiency. First, optimization-based models are able to effectively cover losses, achieving a smooth transition from high number of insolvent states to a low number, thus granting policy providers the discretion to determine the desired insolvency level. Second, optimization-based models are capable of efficiently charge premiums, resulting in a smooth transition from deficit to surplus balance depending on model parameter value. In particular, we recommend policy makers to use an ARO model, with conservative parameter values, to achieve superior performance in both effectiveness and efficiency, resulting in achieving simultaneously low insolvent rate and relatively low premiums charged.

We emphasize the versatility and broad applicability of our framework. This underscores the possibility of employing the ARO approach not just in the context of flood insurance, but also in pricing various catastrophic events such as wildfires, droughts, and other extreme weather conditions. The adaptability of our model suggests it could be a valuable tool in diverse scenarios, offering insights into risk assessment and pricing strategies across different disaster types.

Chapter 5

Reducing Air Pollution through Machine Learning

This chapter presents a data-driven approach to mitigate the effects of air pollution from industrial plants on nearby cities by linking operational decisions with weather conditions. Our predictive and prescriptive framework links operational decisions to weather forecasts to effectively minimize the impact of air pollution in industrial settings. The predictive component of our framework employs various machine learning models, such as gradient-boosted tree-based models and ensemble methods, for time series forecasting. The prescriptive component utilizes interpretable optimal policy trees and explores the effect of different trade-offs ratios.

We have successfully implemented our framework at OCP Group's phosphate production site near the city of Safi. Our deployed algorithm significantly reduced forecasting errors, ranging between 38-52% for less than 12-hour lead time and 14-46% for 12 to 48-hour lead time compared to official weather forecasts. This work aims to provide a pathway to explore sustainable industrial development by linking operational decisions with data-driven weather forecasting capabilities, and a similar framework can be applied in other scenarios where weather plays a role in the management of operational decisions.

5.1 Introduction

Sustainable industrial development is an important issue shared by many countries. The trade-offs between economic activities, environmental pollution, and public health must be managed attentively. Studies show that urbanization and industrialization have released many environmental toxins into the atmosphere over the last 200 years [98]–[100]. In particular, emissions from chemical power plants can pose significant health risks to those living in the surrounding area [101], [102]. Therefore, there is a pressing need to develop technologies and infrastructures to simultaneously achieve economic objectives and environmental preservation.

Air pollution is one of the leading causes of health risks due to industrial activities, according to [103], 3.3 million people die pre-maturely every year globally. The WHO and other health organizations continuously emphasize the importance of monitoring and managing air quality to mitigate these risks. The interplay between airborne pollutants and meteorological conditions is a critical factor in determining the air quality of urban areas and its subsequent impact on human health. This intricate interaction between industrial emissions, wind patterns, and urban meteorological conditions underscores the complex challenge of managing air quality and protecting public health in densely populated regions.

As data availability and computing methods continue to advance, there has been growing interest in applying machine learning techniques to air pollution management. Previous research has primarily focused on predicting the health consequences of pollution exposure [104]. Additionally, various studies have attempted to forecast air pollution, air quality, and airborne particle concentrations using data such as satellite imagery, weather data, and air quality monitoring data [105]–[109]. Despite these efforts, there remains a lack of literature connecting air pollution prediction to decision-making and mitigation actions. Earlier works on technology-aided tools to reduce pollution include [110], which discusses a mathematical formulation and algorithm for controlling air pollution using weather forecasts and numerical models to minimize control-related costs, and [111], which proposes a decision support tool to find optimal Best Management Practice locations for minimizing diffuse surface water pollution.

This work tackles the critical issue of urban air pollution management by proposing a novel plant operation scheduling methodology that leverages machine learning and optimization. Our predictive and prescriptive framework links operational decisions to weather forecasts to effectively minimize the impact of air pollution in industrial settings. To the best of our knowledge, our work is the first attempt to reduce industrial air pollution through machine learning. In addition, the framework is implemented and currently operational on the Safi production site of the OCP group in Morocco. In summary, our contributions are three-fold:

- A data-driven pollution framework incorporating two components: (i) a machine learning-enhanced weather forecasting system that utilizes onsite sensors and official forecasts (ii) an optimization-based operational decision recommendation system optimizing the trade-off between potential pollution risk and operational loss.

- The predictive component of our framework has been deployed at our industrial partner OCP's internal management system and guides production planning in real-time production rate since July 2021. Since implementation, our machine learning-enhanced forecasts significantly improved accuracy: we reduced the next 12-hour wind forecasting errors by 38-52% and the next 12 to 48-hour errors by 14-46%. In addition, our optimization-based operational decision framework is shown to reduce potential polluting cases by 33-47% while achieving 40-63% operational savings.

- Our work offers a case study of achieving sustainable industrial activities through enhanced data-driven meteorological forecasts in interplay of weather conditions and industrial activities. We hope to inspire future work applying machine learning driven meteorology for industrial development which depends on meteorological conditions.

5.2 Methodology

In urban areas located nearby factories emitting airbourn pollutants, wind conditions play a critical role in the dynamic determining the concentration of pollutants like particulate matter, nitrogen oxides, and sulfur dioxide produced by industrial processes. When winds are directed towards urban areas from industrial zones, they can carry a higher load of pollutants, significantly worsening air quality. Conversely, favorable wind conditions can disperse these pollutants, mitigating their impact. However, in scenarios of low wind speeds or temperature inversions, pollutants tend to accumulate, particularly in urban areas with the urban heat island effect, leading to higher concentrations of harmful substances and increased exposure for the population.

This study proposes a data-driven framework to reduce the impact of air pollution from industrial plants on nearby cities by responsively adapting production levels based on wind speed and direction. Our pipeline encompasses two parts: i) machine learning algorithms producing more accurate and frequent wind forecasts by combining official weather data and onsite real-time sensory data to aid short-to-medium term factory and personnel planning; ii) an optimization-based framework to recommend real-time optimal operational decisions taking into account the various forecasts from the machine learning models. Figure 5.2 illustrates our overarching methodology.

5.2.1 Case Study on Safi

The Safi city in Morocco has more than 300,000 residents, and is located 10km northeast of a large phosphate manufacturing plant operated by the OCP group. The OCP Group is world's largest phosphate producer, accounting for more than 30% of global production of phosphate, an important ingredient for agricultural fertilizer. However, phosphate production releases harmful airborne substances such as sulfur dioxide (SO_2), sulfur trioxide (SO_3), hydrogen sulfide (H_2S), and hydrogen fluoride (HF), as well as fine and coarse dust, which

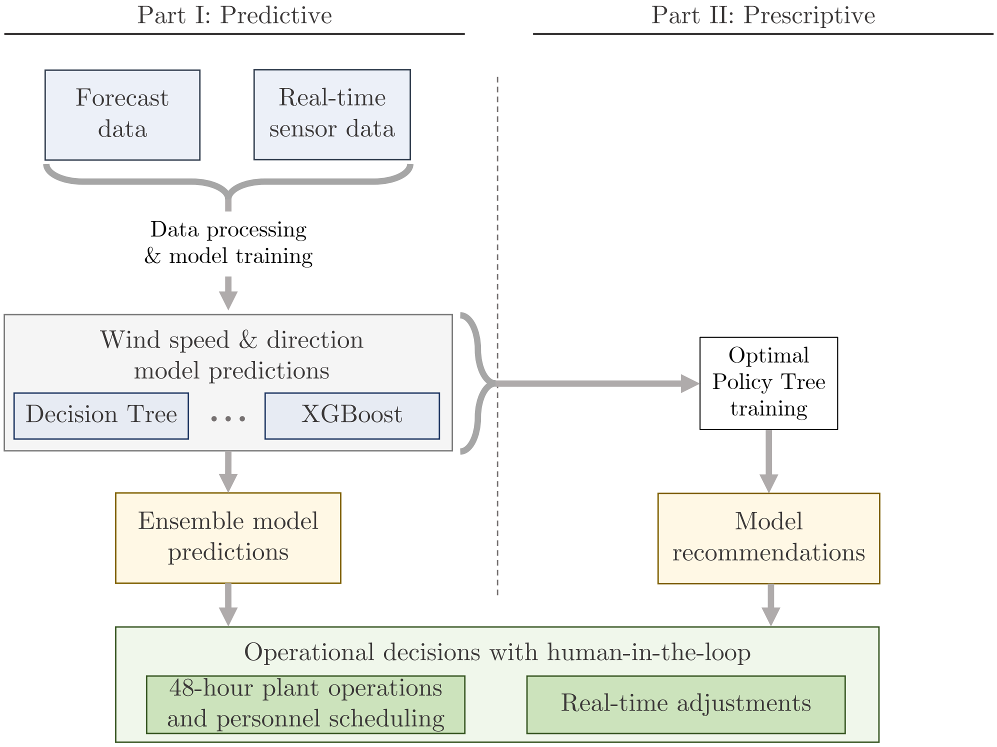

Part I: Predictive
Part II: Prescriptive
Forecast data
Real-time sensor data
Data processing & model training
Wind speed & direction model predictions
Decision Tree
. . .
XGBoost
Optimal Policy Tree training
Ensemble model predictions
Model recommendations
Operational decisions with human-in-the-loop
48-hour plant operations and personnel scheduling
Real-time adjustments

Figure 5.1: Predictive and prescriptive approach to plant operations scheduling.

can pose serious health risks such as respiratory diseases and cancer [112].

To reduce the air pollution impact from industrial activities, the Safi site set up a monitoring procedure with responsive production rates — and consequently airborne emissions — depending on the meteorological conditions. However, there are two main challenges of such an attempt. Firstly, there is a significant gap gap between official meteorological forecasts and real-time conditions, thus leading to unnecessary and costly production shutdowns or missed dangerous weather conditions. The national weather forecasts are frequently inaccurate because they are calculated at a regional level and come with a 5 to 7-hour lag-time due to the long computational costs of dynamical weather forecasts. Secondly, the process is highly manual and depends on the expertise and experience of operators to make advance scheduling according to weather forecasts. The lack of transparency and error tracking make the process hard to improve.

In this work, we apply our predictve-and-prescriptive framework using on-site data to improve the operations of the Safi site.

5.2.2 The Previous Operational Procedure at Safi

The OCP group is the world's largest phosphate producer, controlling 75% of the world's phosphate reserves and accounting for more than 30% of global production. The OCP Safi site was established in the 1970s to produce various phosphates for export. However, fertilizer production is a known contributor to air pollution, releasing harmful airborne substances such as sulfur dioxide (SO_2), sulfur trioxide (SO_3), hydrogen sulfide (H_2S), and hydrogen fluoride (HF), as well as fine and coarse dust, which can pose serious health risks such as respiratory diseases and cancer [112]. The site is located 10 km southwest of the Safi city center, with more than 300,000 residents. Due to the geographical location, weather conditions play a critical role in air pollution dispersion. Depending on the wind speed and direction, airborne pollution can be carried into Safi, thus posing a threat to public health and bringing high respiratory and ocular discomfort. In 2013, the site set up a monitoring procedure to reduce the amount of air pollution in the city with responsive production rates — and consequently airborne emissions — depending on the meteorological weather forecasts and real-time on-site wind monitoring system. This procedure schedules production rates and personnel based on next-day weather forecasts. It uses real-time wind monitoring systems to adjust in dangerous weather conditions, ensuring the safety of the surrounding community.

Before this work, the main bottleneck of the procedure was the gap between meteorological forecasts and real-time conditions, thus leading to unnecessary and costly production shutdowns or missed dangerous weather conditions, leading to negative health outcomes. The operators in the Safi production site received operational weather forecasts from the national meteorological agency every 12 hours for the next 48 hours. However, these forecasts are frequently inaccurate because they are calculated at a regional level and come with a 5 to 7-hour lag-time due to the long computational costs of dynamical weather forecasts. As a result, planning activities had no access to real-time forecast information and were sensitive to uncertain weather conditions.

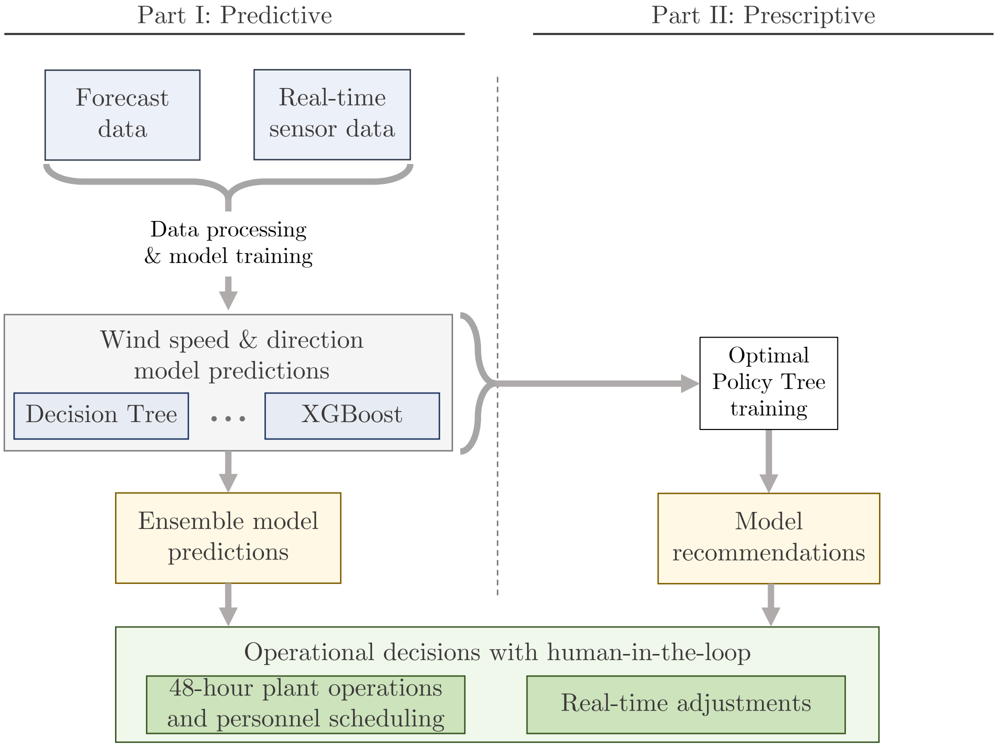

Part I: Predictive
Part II: Prescriptive
Forecast data
Real-time sensor data
Data processing & model training
Wind speed & direction model predictions
Decision Tree
...
XGBoost
Optimal Policy Tree training
Ensemble model predictions
Model recommendations
Operational decisions with human-in-the-loop
48-hour plant operations and personnel scheduling
Real-time adjustments

Figure 5.2: Predictive and prescriptive approach to plant operations scheduling.

This study aims to develop a data-driven framework to reduce the impact of air pollution from industrial plants on nearby cities by responsively adapting production levels based on wind speed and direction. Our pipeline encompasses two parts: i) machine learning algorithms producing more accurate and frequent wind forecasts by combining official weather data and onsite real-time sensory data to aid short-to-medium term factory and personnel planning; ii) an optimization-based framework to recommend real-time optimal operational decisions taking into account the various forecasts from the machine learning models. Figure 5.2 illustrates our overarching methodology.

5.2.3 Scenario Definition

To categorize different wind scenarios and subsequent polltion dispersion impacts for the Safi site, we developed a warning system to categorize weather conditions into several scenarios. Scenarios are differentiated as either *favorable* (S1, S2, S2b, S3) or *dangerous* (S3b, S4) based on wind speed and direction, as outlined in Table 5.1. This categorization accounts for five wind speed buckets and three wind direction buckets, providing detailed guidance on production rates for each scenario. A dangerous scenario is characterized by low wind speed combined with an unfavorable wind direction, which results in pollutants being directed toward and lingering in the city (see illustration in Figure 5.3). Based on the real-time and predicted scenarios, operational decisions are made to reduce air pollution according to the action rules outlined in Table 5.2.

Wind	**Favorable Wind Direction**	**Very Unfavorable Wind Direction**	**Unfavorable Wind Direction**
Speed	NW, N-NW, N, N-NE, NE, E-NE, E	S-SW, S, S-SE	E-SE, SE, SW, W-SW, W, W-NW
(m.s^{-1})	0° – 101.25° & 303.75° – 0°	146.25°– 213.75°	101.25° – 146.25° & 213.75° – 303.75°
$V < 0.5$	S3 a	S4	S4
$0.5 \leq V < 1$	S2	S3 b	S2 b
$1 < V \leq 2$	S1	S3 b	S2 b
$2 < V \leq 4$	S1	S3 b	S2
$4 < V$	S1	S1	S1

Table 5.1: Scenario definitions based on wind speed and direction, accounting for five wind speed buckets and three wind direction buckets. Scenarios are differentiated as either favorable (S1, S2, S2b, S3a) or dangerous (S3b, S4).

Scenario Type	**Underlying Scenarios**	**Scenario Characteristics**	**Public Health Consequences**
Favorable	S1, S2, S2b, S3	High wind speed and/or favorable wind direction	Limited
Dangerous	S3b, S4	Low wind speed and unfavorable wind direction	Pollutants directed toward and lingering in city

Table 5.2: Categorization of scenarios as favorable and dangerous based on wind speed and direction.

5.2.4 Predictive Methodology

We employ machine learning models to produce accurate hourly wind forecasts by integrating official weather data and onsite real-time sensory data to aid short-to-medium factory and personnel planning. We used data from the Safi site as a case study, and the success in this framework led to an implementation and integration of our algorithm into the the OCP Safi internal system. Our forecasts have been guiding their operational planning since July 2021. This successful implementation serves as a model for other factories seeking to improve their sustainability efforts and reduce their environmental impact.

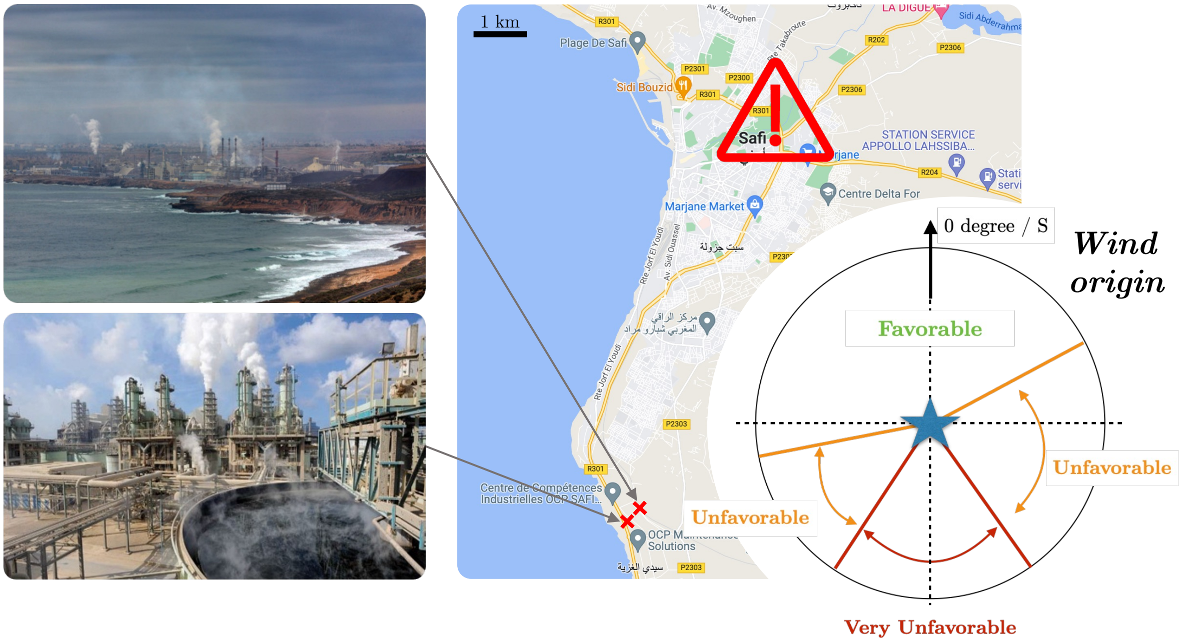


Figure 5.3: Wind direction and wind speed determine the dissemination of pollutants. Winds coming from the South with low speeds are the most dangerous conditions.

Data Processing We combined two datasets to make predictions: the official regional weather forecast data and real-time weather measurement data collected with on-site sensors. Data used in this study range from July 2015 to March 2022.

Official forecasts are received twice daily, around 6:00 am (GMT) and 6:00 pm (GMT) from the Moroccan National Meteorological Department. These forecasts are produced by traditional dynamical models with initial conditions and often take 5-7 hours of computational time. They provide hourly values for the next 48 hours for wind speed, wind direction, humidity, solar irradiance, and temperature at the Safi site. We call this model the *baseline model* in the rest of the paper. The on-site sensors measure the same five weather features (wind speed, wind direction, humidity, solar irradiance, and temperature) at one-minute intervals.

We first imputed the missing values caused by electronic or server malfunctions with linear interpolation. We then averaged the measurement data over one-hour intervals. We used the arithmetic average for the humidity, solar irradiance, and temperature, and the vector average technique [113] for wind speed and direction (e.g., the vector average of a southerly and a northerly wind of 5 m.s^{-1} gives a mean wind speed of 0 m.s^{-1} because there is no resultant wind speed). We encoded the wind direction using the cosine and sine transformations to avoid singularities at endpoints due to the cyclical nature of the feature.

Training data creation We transformed the time-series data into a standard tabular form to train traditional machine-learning models. To make wind predictions at time t for the hour $t + h$, we concatenated the present and past 48 hours of weather measurement features at each time step into a vector. Then, we appended the following features: the latest operational forecast available at time t for wind speed, wind direction, pluviometry, and solar irradiance; the cosine and sine of the day and the hour corresponding to time t. Table 5.3 summarizes the 304 features and associated processing techniques.

Feature Description	**Processing Technique**	**Initial Feature Range**	**Number of features**
Wind speed	Vector average	0.00 - 14.20 m.s^{-1}	49
Wind direction	Vector average, cos/sin encoding	[0, 360]°	49 × 2
Solar irradiance	Arithmetic average	0.0 - 978.4 W.m^{-2}	49
Temperature	Arithmetic average	4.8 - 46.7°C	49
Pluviometry	Arithmetic average	0.0 - 17.2 mm	49
Day of the year	Cos/sin encoding	1 - 365	2
Hour of the day	Cos/sin encoding	0 - 23	2
Official forecast for wind speed		0.0 - 16.5 m.s^{-1}	1
Official forecast for wind direction	Cos/sin encoding	[0, 360]°	2
Official forecast for pluviometry		0.0 - 20.8 mm	1
Official forecast for solar irradiance		0 - 1074 W.m^{-2}	1
Official forecast for temperature		3.2 - 43.1°C	1

Table 5.3: Table recording all the features and processing techniques. The number of features obtained accounts for concatenating the past 48-hour values.

Model Training For the prediction task, we trained five different types of machine learning models to predict wind speed and direction, including Elastic Net, Decision Trees, Random Forest, LightGBM, and XGBoost. To handle the cyclical property for wind direction, we predicted the cosine and sine of the angle instead of the raw angle degree. Predictions are then converted back into scenario predictions using Table 5.1. We trained one model for each lead time between 1 and 48 hours ahead, i.e., 48 × 3 regression models for wind speed, cosine, and sine of wind direction. We performed hyperparameter tuning for each model using the validation set as explained later in Section 5.2.6.

In addition, we trained ensemble models to predict wind speed and direction for every lead time using predictions from these previous individual machine-learning models. Ensemble modeling is a well-established technique to leverage the strengths and limitations of multiple models and benefit from their diversity. The principle is to combine the predictions of the forecasting models available to obtain a more accurate, stable, and robust predictor. In our case, we used the stacking [114] concept and tried several ensemblers, including decision trees, regularized linear regression, and gradient-boosted trees. Elastic Net regression performed the best, and we considered it our final ensemble model technique.

5.2.5 Prescriptive Methodology

As the second component and to acknowledge the errors in machine learning driven forecasts, we developed an optimization-based decision recommendation system to determine the most optimal decision in real-time given the forecasts made by the different machine learning models. We employed Optimal Policy Trees (OPT) [115] to determine the most optimal decision in real-time given the forecasts made by the different machine learning models.

The prescriptive approach employs observational data of the form $\{(\mathbf{x}_i, y_i, z_i)\}$. Each observation i consists of features $\mathbf{x}_i \in \mathbb{R}^{18}$ (the ensemble members' predictions), an applied prescription $z_i \in \{0, 1\}$ (reduce plant production or not), and an observed outcome $y_i \in \mathbb{R}$ (real-world costs associated with the decision). Our prescriptive task is determining the optimal policy that, given the features $\mathbf{x}$, prescribes the treatment z that results in the best outcome y. The prescription involves choosing between one of two available decisions, either to reduce production or not.

Table 5.4 outlines the reward matrix used to train the Optimal Policy Tree and quantifies the costs associated with false positives and false negatives. First, no cost is incurred if the forecasted scenario and actual conditions are favorable. When the plant operates at reduced levels as a conservative measure after forecasting a dangerous scenario, the factory

Forecasted Scenario	**Actual Scenario**	**Cost** (USD)	**Decision Outcome**	**Public Health Impact**
Favorable	Favorable	0	Full level production	Low
Dangerous	Favorable	2000	Reduced level production + anti-odor injection	Low
Dangerous	Dangerous	2000	Reduced level production + anti-odor injection	Low
Favorable	Dangerous	4000 - 20000	Full production before urgent shutdown + anti-odor injection	High

Table 5.4: Reward matrix for training the Optimal Policy Trees based upon forecasted and actual weather conditions.

incurs a loss of earnings of $2,000 per hour due to decreased production and the expenses of injecting odor control chemicals to minimize unpleasant odors in the surrounding area. On the other hand, the failure to forecast a dangerous scenario leads to the plant operating at a normal level and polluting the nearby city when the weather conditions turn dangerous. Afterward, the plant operators must shut down production urgently and inject odor control chemicals. We propose evaluating various public health costs ranging from $2,000 to $18,000. This parameter yields differing trade-offs between pollution and costs and can be determined based on the decision-makers' conservatism and risk aversion level.

5.2.6 Training Protocol

The data covers August 2015 to March 2022, totaling 43,952 hourly samples. The data set was divided into training (60%), validation (20%), and testing (20%) sets. The validation set was used to tune the hyperparameters of the machine-learning models. The ensemble models and optimal policy tree parameters were 5-fold cross-validated on the predictions made on the validation set. All models were evaluated on the unseen test set corresponding to the real-world deployment phase.

Software Tools We used Python 3.8 [47] and the scikit-learn package [116] to implement all machine learning models. We used the Python package InterpretableAI [117] to train Optimal Policy Trees.

Model	**Hyperparameters**	**Values**
Elastic Net	regularization α coefficient	0.2, 0.4, 0.6, 0.8, 1
	ℓ_1 ratio	0.5, 0.7
Decision Trees	maximum tree depth	5, 6, 7, 8, 9, 10
	minimum samples per split	3, 5, 7
	minimum samples per leaf	4, 6
Random Forest	bootstrap	True, False
	number of estimators	100, 150
	maximum tree depth	5, 6
	min samples split	4, 6
LightGBM	number of leaves	31, 60
	maximum tree depth	4, 6
	learning rate	0.1, 0.3
	lambda ℓ_1	0, 1
XGBoost	number of estimators	100, 150
	maximum tree depth	4, 6
	learning rate	0.1, 0.3
Elastic Net Ensemble	regularization α coefficient	0.2, 0.4, 0.6, 0.8, 1
	ℓ_1 ratio	0, 0.25, 0.5, 0.75, 1.0

Table 5.5: Hyperparameters searched for our models.

5.3 Results

This section reports the results of the two components of the framework: predictive and prescriptive. We have successfully implemented the predictive component on machine learning-based wind forecasts since December 2020, and we are currently implementing the prescriptive component on operational decision-making recommendations. As such, we report real-world deployment results for the predictive component and back-tested results for the prescriptive component.

5.3.1 Predictive Methodology Results

Tables 5.6 and 5.7 report the results of the wind speed and wind direction forecasting tasks for all the regression models we deployed at the Safi site: the baseline model, Elastic Net, Decision Tree, Random Forest, Light GBM, XGBoost, and the Elastic Net ensemble model. For each wind prediction task, we report the mean absolute error (MAE) and the expected shortfall at 85%, corresponding to the average error on the worst 15% samples. The baseline model refers to the weather forecast guidance from the Moroccan Meteorological Department.

All machine learning models generally improve upon the baseline model, with XGBoost and Light GBM achieving the lowest errors. In addition, the ensemble model further improves the MAE and expected shortfall, especially in near-term horizons. Looking at speed prediction, for less than 12-hour lead time, the best-performing machine learning approaches can outperform the baseline model by 40-50% in both metrics. For longer-term predictions with more than a 12-hour horizon, the best-performing machine learning approaches can outperform the baseline model by 20-30%. We observe a similar trend for angle prediction: machine learning approaches can achieve 30-50% improvement upon the baseline model for less than 12-hour lead time predictions and 10-20% improvement for longer lead time predictions.

In addition, we observe that the ensemble model outperforms the best single machine learning model consistently across tasks and error measures. The advantage of an ensemble model is especially strong for less than 12-hour lead time predictions. The ensemble model can achieve 0-8% MAE reduction depending on the specific lead time (except for a slightly worse performance on the longer-term expected shortfall for speed).

Lead Time	**Metric**	**Baseline**	**Elastic Net**	**Decision Tree**	**Random Forest**	**LightGBM**	**XGBoost**	**Ensemble**
1	MAE ($m.s^{-1}$)	0.96	0.54	0.56	0.53	0.49	0.49	**0.48**
2		1.05	0.71	0.76	0.71	0.64	0.65	**0.63**
3		1.18	0.80	0.85	0.8	0.72	0.72	**0.70**
6		1.61	0.91	0.97	0.91	0.85	0.85	**0.84**
12		1.94	1.0	1.05	0.99	0.96	0.96	**0.94**
24		1.37	1.07	1.11	1.08	1.06	1.06	**1.04**
36		2.13	1.17	1.20	1.17	1.16	1.16	**1.15**
48		1.57	**1.17**	1.22	1.18	**1.17**	**1.17**	**1.17**
1	Expected Shortfall 85%	2.37	1.40	1.48	1.36	1.27	1.28	**1.26**
2		2.60	1.80	1.95	1.77	1.63	1.65	**1.61**
3		2.94	2.01	2.16	1.99	1.81	1.82	**1.78**
6		3.82	2.27	2.45	2.25	2.15	**2.14**	**2.14**
12		4.55	2.48	2.64	2.44	**2.38**	2.40	2.39
24		3.51	2.61	2.77	2.60	**2.58**	**2.58**	2.59
36		4.93	2.79	2.89	2.76	**2.75**	**2.75**	2.77
48		4.03	2.80	2.95	2.79	**2.78**	**2.78**	2.81

Table 5.6: Beta test results on the test set for wind speed prediction for all models. We record the MAE and expected shortfall at 85% level for different lead times ranging from 1 hour to 48 hours.

Lead Time	Metric	Baseline	Elastic Net	Decision Tree	Random Forest	LightGBM	XGBoost	Ensemble
1	MAE (m.s^{-1})	25	26	14	13	**12**	13	**12**
2		26	31	20	18	17	17	**16**
3		29	35	23	21	19	19	**18**
6		41	40	29	28	**24**	**24**	**24**
12		58	43	35	33	31	31	**30**
24		42	45	40	38	37	38	**36**
36		70	49	45	42	42	42	**41**
48		52	48	47	44	44	44	**42**
1	Expected Shortfall 85%	73	82	59	53	**51**	**51**	**51**
2		79	103	77	72	**68**	**68**	**68**
3		89	117	90	83	76	75	**74**
6		115	132	108	107	**94**	95	95
12		136	138	127	125	**117**	**117**	**117**
24		**127**	143	138	135	134	135	132
36		155	148	145	**140**	**140**	**140**	**140**
48		145	148	146	**143**	**143**	**143**	**143**

Table 5.7: Beta test results on the test set for wind direction prediction for all models. We record the MAE and expected shortfall at 85% level for different lead times ranging from 1 hour to 48 hours.

5.3.2 Prescriptive Methodology Results

Table 5.8 compares the performance of several models for recommending binary hourly actions (anticipating dangerous conditions or maintaining production levels). It includes the baseline model, the previous Elastic Net ensemble model, and a series of Optimal Policy Trees (OPT) with different health costs associated with false negatives (-4000, -6000, -10000, -15000, and -20000). Recall that the health cost used to train OPTs is a parameter to tune conservatism towards pollution of our models: a higher cost leads to more cautious care towards recommending operation at normal levels. Table 5.8 reports the number of false positives and false negatives for each model. A false positive refers to when the plant operates with reduced production levels and undertakes actions to mitigate the odor impact, while in reality, these actions are unnecessary. A false positive is therefore associated with a loss of $2000 corresponding to the anti-odor injection costs and consequences of reduced production. On the other hand, a false negative refers to when the plant is operating at normal levels, while in reality, weather conditions are unfavorable, and pollution is carried to the city, leading to an air pollution incident.

As a remark, since the implementation of this component is underway, and we do not track the actual decisions undertaken by operators, we showcase back-testing results using the baseline model as a benchmark. We simulate decisions using forecasts from the baseline and ensemble models and translate them into decisions using Table 5.1. The actual decisions can deviate from the simulated decisions because operators make decisions based on forecasts and expertise.

In general, our framework can lead to reductions in both false positives and false negatives. Specifically, looking at Optimal Policy Tree models, different choices of health cost lead to different levels of conservatism, which gives the modeler the space to explore the trade-off between cost savings and pollution mitigation goals. As the health cost increases, false negatives decrease, and false positives increase. Comparing the OPT models with the baseline model shows that the OPT models have overall better performance, as they have lower health costs and fewer false positives.

Model	**Health Cost**	**False Positives**	**False Negatives**	**Cost savings**	**Pollution Reduction**
Baseline		288	110	0%	0%
Ensemble		51	133	82%	-21%
Optimal Policy Tree	-4000	20	113	93%	-3%
	-6000	32	102	89%	7%
	-10000	106	74	63%	33%
	-15000	174	58	40%	47%
	-20000	282	38	2%	65%

Table 5.8: Performance of three families of models for recommending actions. We include the baseline model, the Elastic Net ensemble, and a series of optimal policy trees trained with different health costs associated with false negatives.

In addition, the OPTs provide interpretable insights on how the different ensemble members are used to prescribe, as illustrated by Figure 5.4 below. In particular, we notice that a simple tree like the one corresponding to choosing a health cost of $15,000 (tree on the right below) can reduce pollution emissions during dangerous scenarios by 47% and save 40% of the unnecessary costs. Conveniently, it also relies on only three ensemble members: XGBoost and Elastic Net predicting speed, and Random Forest predicting the cosine component of the wind direction. It also suggests that different ensemble members capture different aspects of the data and together make better recommendations.

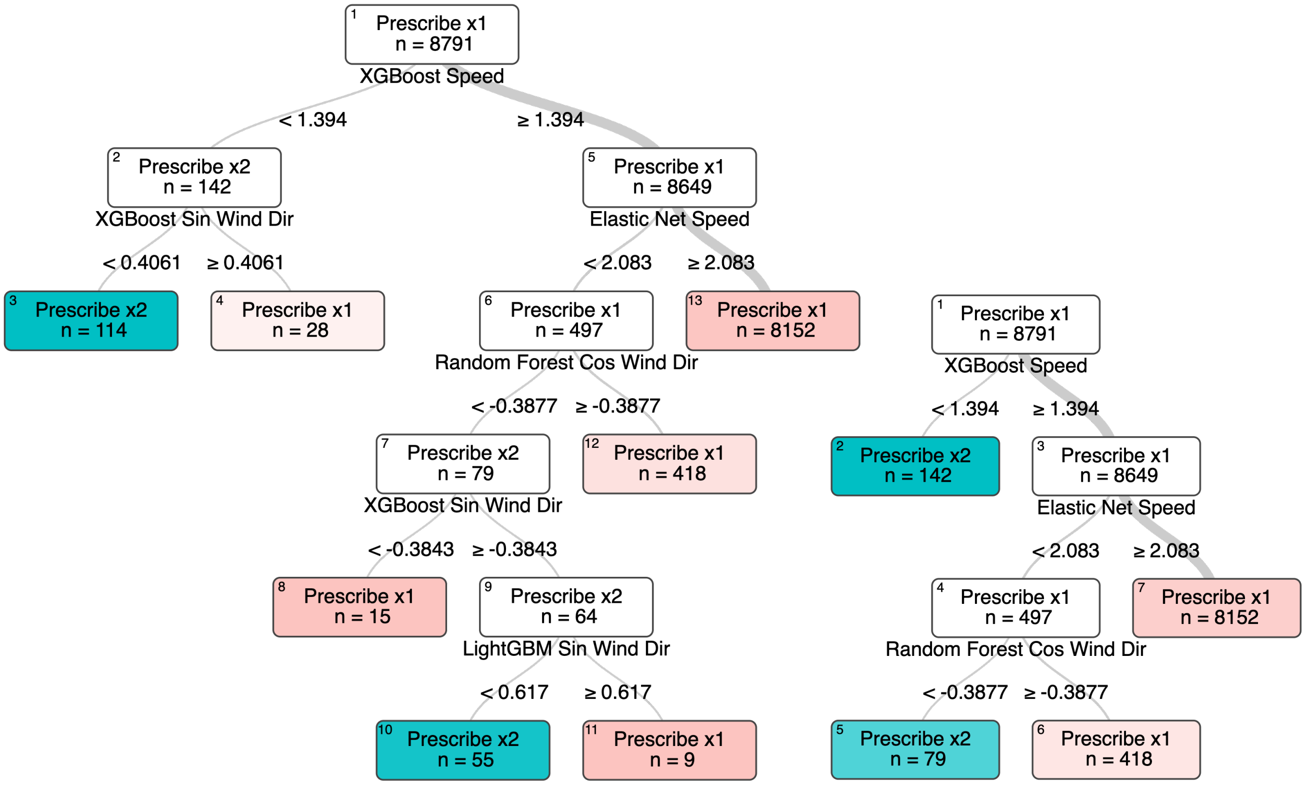
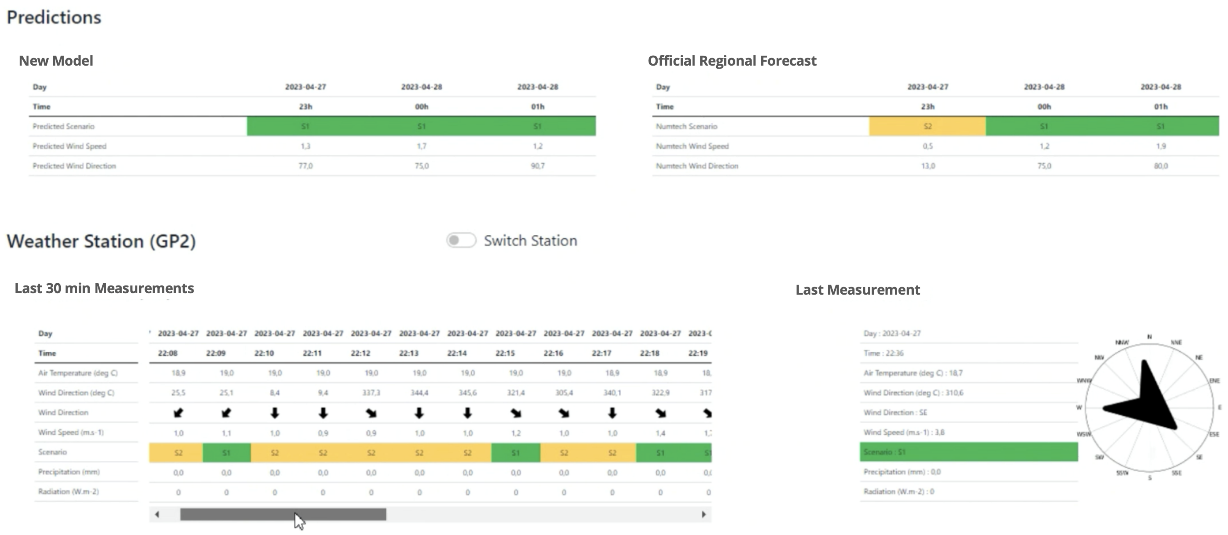


Figure 5.4: Optimal policy trees trained using a health cost of $10,000 (left) and $15,000 (right). The trees illustrate an interpretable decision-making process to arrive at certain recommended decisions (prescription options). `Prescribe x1` corresponds to maintaining production rate while `Prescribe x2` corresponds to reducing plant operations and injecting odor control chemicals.

Qualitative Feedback from Real-World Implementation

Our collaboration with OCP's software development team has seamlessly integrated our weather forecasts into the company's internal system (see Figure 5.5). As of July 2021, the site manager and plant operators have been utilizing the forecasts produced by our framework through a simple user interface. They check the hourly forecasts before scheduling production shutdowns, leading to a significant reduction in production downtime.

Qualitative feedback from production managers has indicated that our forecasts are substantially more accurate than official weather forecasts and provide valuable real-time updates that are particularly advantageous during winter when wind conditions are more unpredictable. This has improved factory planning and resource allocation, allowing for more



Predictions

New Model

Day	2023-04-27	2023-04-28	2023-04-28
Time	23h	00h	01h
Predicted Scenario	S1	S1	S1
Predicted Wind Speed	1,3	1,7	1,2
Predicted Wind Direction	77,0	75,0	90,7

Official Regional Forecast

Day	2023-04-27	2023-04-28	2023-04-28
Time	23h	00h	01h
Numtech Scenario	S2	S1	S1
Numtech Wind Speed	0,5	1,2	1,9
Numtech Wind Direction	13,0	75,0	80,0

Weather Station (GP2)

Last 30 min Measurements

Day	2023-04-27	2023-04-27	2023-04-27	2023-04-27	2023-04-27	2023-04-27	2023-04-27	2023-04-27	2023-04-27	2023-04-27	2023-04-27	2023-(
Time	22:08	22:09	22:10	22:11	22:12	22:13	22:14	22:15	22:16	22:17	22:18	22:19
Air Temperature (deg C)	18,9	19,0	19,0	19,0	19,0	19,0	19,0	19,0	19,0	18,9	18,9	18,
Wind Direction (deg C)	25,5	25,1	8,4	9,4	337,3	344,4	345,6	321,4	305,4	340,1	322,9	317
Wind Direction	↙	↙	↓	↓	↘	↓	↓	↘	↘	↓	↘	↘
Wind Speed (m.s-1)	1,0	1,1	1,0	0,9	0,9	1,0	1,0	1,2	1,0	1,0	1,4	1,:
Scenario	S2	S1	S2	S2	S2	S2	S2	S1	S2	S2	S1	S1
Precipitation (mm)	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,(
Radiation (W.m-2)	0	0	0	0	0	0	0	0	0	0	0	0

Last Measurement

Day : 2023-04-27
Time : 22:36
Air Temperature (deg C) : 18,7
Wind Direction (deg C) : 310,6
Wind Direction : SE
Wind Speed (m.s-1) : 3,8
Scenario : S1
Precipitation (mm) : 0,0
Radiation (W.m-2) : 0

Figure 5.5: Screenshot of the software platform used by the plant operators in Safi. Our model predictions for the next 3 hours are displayed on the upper left, while the baseline model is displayed on the upper right. Below, the operator can check the real-time 1-min measurements of the previous 30 minutes to adapt decisions.

efficient production, better personnel scheduling, and cost savings for the company.

The successful implementation of our framework at OCP Safi is a testament to our approach's effectiveness in optimizing factory operations. We believe that utilizing our framework has the potential to advance how factories approach planning and resource allocation, ultimately leading to improved sustainability efforts and environmental impact reduction.

5.4 Conclusion

In conclusion, our study introduces a novel and data-driven solution to mitigate the harmful effects of air pollution caused by industrial plants in urban areas. We provide a comprehensive solution for managing industrial operations and weather-related risks by combining advanced weather forecasting and decision-making models. Our framework, which incorporates both predictive and prescriptive machine learning models, was successfully implemented at the OCP Safi production site, resulting in improved forecasting accuracy and decision-making efficiency. Given the crucial role of weather in industrial environmental impact, we believe that our approach can be adapted and effectively applied in similar settings.

Our framework has demonstrated its value in managing air pollution in chemical production sites, and the results achieved at the OCP Safi site hold the potential to inspire a more sustainable and responsible chemical production industry globally. The flexibility and adaptability of our approach enable its core components of data enhancement, real-time monitoring, and prescriptive models to be universally applied to different chemical factories. Although each production site presents unique challenges, our data-driven approach can be customized to meet the needs and conditions of each location. Utilizing the latest advancements in weather forecasting and data analysis, we aim to assist factories in effectively managing air pollution and promoting the safety and well-being of the surrounding communities.

Acknowledgement

We thank the OCP group for their invaluable contribution to this work. We warmly acknowledge Karim Bendioub, Yassine Azhari, Aimad Aneddame, and Walid Daou, whose support was critical in developing and implementing this work. Karim Bendioub made a major contribution by building the software to integrate the models into the pipelines at the OCP Safi factories and replicating the code behind the firewall.

Chapter 6

Conclusion

In this dissertation, we embark on a journey to explore ways of leveraging machine learning and optimization methods to navigate the challenges related to climate change adaptation and sustainable development. We harness the power of machine learning to enhance our predictive capabilities for extreme weather conditions, and thereby combine with decision-making tools to develop robust preparatory and adaptive strategies.

Structured into two principal segments, this thesis comprises the methodological component of machine learning frameworks for extreme weather forecasting and the subsequent applications towards climate adaptation and sustainable development goals. The initial chapters are dedicated to the development of a generalizable multimodal machine learning framework to synthesize diverse data formats, ranging from satellite imagery, tabular time series datasets to textual descriptions, in order to enhance weather forecasting capabilities. Specifically, Chapter 2 focuses on applying the framework to predict hurricanes with a 12-hour lead time by integrating satellite and tabular time series data, while Chapter 3 further expands to include text data for long-term flood risk assessments.

These methodological investigations not only extend the frontiers of our predictive capabilities but also shed light on enhancing decision-making processes across a multitude of sectors. Therefore, the subsequent chapters pivot towards the pragmatic deployment of these advanced weather models in devising strategies for climate change adaptation. In chapter 4, we integrate machine learning-driven risks using an adaptive robust optimization framework for catastrophe insurance pricing, and demonstrate the efficiency using US National Flood Insurance data. In chapter 5, we develop a data-driven framework, employing machine learning models for near-term wind prediction, to reduce air pollution impact from industrial operations to surround urban areas.

This body of work represents a critical junction in the application of machine learning with the field of meteorology, facilitating a data-driven paradigm shift towards enhanced resilience against the challenges posed by climate change. These innovations hold profound implications across numerous sectors, encompassing infrastructure planning, urban development, insurance, and sustainable energy management, among others. Yet, this dissertation merely scratches the surface of potential advancements. Looking forward, there is a vast landscape of opportunities to expand upon.

I believe that Artificial Intelligence (AI) will fundamentall change our relationship with the weather. As we look to the future, it becomes increasingly clear that adapting our societal frameworks and operational methodologies to the ever-evolving climate is not just a necessity but an imperative for sustainable growth. I am committed to advancing technological solutions that not only facilitate climate change adaptation but also propel us towards enduring sustainable development.

Appendix A

Appendix for Chapter 2

A.1 Encoder-Decoder Architectures

A.1.1 Overall Architecture and Mechanisms

The CNN-encoder At each time step, we feed the nine reanalysis maps into the CNN-encoder, which produces one-dimensional embeddings. The CNN-encoder consists of three convolutional layers, with ReLU activation and MaxPool layers in between, followed by two fully connected layers.

Next, we concatenate the reanalysis maps embeddings with processed statistical data corresponding to the same time step. At this point, data is still sequentially structured as 8 time steps to be passed on to the decoder.

The GRU-Decoder Our GRU-decoder consists of two unidirectional layers. The data sequence embedded by the encoder is fed sequentially in chronological order into the GRU-decoder. For each time step, the GRU-decoder outputs a hidden state representing a "memory" of the previous time steps. Finally, a track or intensity prediction is made based upon these hidden states concatenated all together and given as input to fully-connected layers (see Figure 2.3).

The Transformer-Decoder Conversely to the GRU-decoder, we feed the sequence as a whole into the Transformer-decoder. The time-sequential aspect is lost since attention mechanisms allow each hidden representation to attend holistically to the other hidden representations. Therefore, we add a *positional encoding* token at each timestep-input, following standard practices [38]. This token represents the relative position of a time-step within the sequence and re-introduces some information about the inherent sequential aspect of the data and experimentally improves performance.

Then, we use two Transformer layers that transform the 8 time steps (of size 142) into an 8-timestep sequence with similar dimensions. To obtain a unique representation of the sequence, we average the output sequence feature-wise into a one-dimensional vector, following standard practices. Finally, a track or intensity prediction is made based upon this averaged vector input into one fully-connected layer (see Figure 2.4).

Loss function The network is trained using an objective function $\mathcal{L}$ based on a mean-squared-error loss on the variable of interest (maximum sustained wind speed or TC displacement) added to an $L2$ regularization term on the weights of the network:

$$\mathcal{L} := \frac{1}{N}\sum_{i=1}^{N}\left(y_i^{\text{true}} - y_i^{\text{pred}}\right)^2 + \lambda \sum_{l}\sum_{k,j} W_{k,j}^{[l]2},$$

where N is the number of predictions, y_i^{pred} the predicted forecast intensity or latitude-longitude displacements with a lead time of 24 h, y_i^{true} the ground-truth values, λ a regularization parameter chosen by validation, $W^{[l]}$ the weights of the l-th layer of the network. We minimize this loss function using the Adam optimizer [118].

A.1.2 Technical Details on the CNN-Encoder GRU-Decoder Network

We provide more formal and precise explanations of our encoder-decoder architectures.

CNN-encoder GRU-decoder architecture details Let t the instant when we want to make a 24-hour lead time prediction. Let $\mathbf{x}_t^{\text{viz}} \in \mathbb{R}^{8\times 9\times 25\times 25}$ be the corresponding spatial-temporal input of the CNN, where 8 is the number of past time steps in the sequence, 9 is the number of pressure levels times the number of features maps, 25° × 25° is the pixel size of each reanalysis map. Let $\mathbf{x}_t^{\text{stat}} \in \mathbb{R}^{8\times 31}$ be the corresponding statistical data, where 8 is the number of time steps in the sequence, and 31 the number of features available at each time step.

First, $\mathbf{x}_t^{\text{viz}}$ is embedded by the CNN into $\mathbf{x}_t^{\text{emb}} \in \mathbb{R}^{8\times 128}$ where 8 is the number of time steps in the sequence, 128 is the dimension of the embedding space. Figure A1 provides an illustration of this embedding process by the CNN-encoder.

Let $i \in \{0, \dots, 7\}$ be the corresponding index of the time step in the sequence t. At each time step t_i of the sequence, the CNN embedding $\mathbf{x}_{t_i}^{\text{emb}}$ is concatenated with the statistical data $\mathbf{x}_{t_i}^{\text{stat}}$ and processed as

$$\mathbf{h}_{t_i} := \text{GRU}(\mathbf{h}_{t_{i-1}}, [\mathbf{x}_{t_i}^{\text{emb}}, \mathbf{x}_{t_i}^{\text{stat}}]),$$

with $\mathbf{h}_{t_0} = \mathbf{0}$, $\mathbf{h}_{t_i} \in \mathbb{R}^{128}, \forall i$. $[\cdot, \cdot]$ means concatenation of the two vectors along the column axis, to keep a one-dimensional vector.

Finally, we concatenate $\mathbf{h}_{t_0}, \mathbf{h}_{t_1}, \dots, \mathbf{h}_{t_7}$ to obtain a one-dimensional vector $\mathbf{x}_t^{\text{hidden}}$ of size $8 \cdot 128 = 1024$ and pass this vector into a series of 3 fully connected linear layers, of input-output size: (1024, 512); (512, 128); (128,c), where $c = 2$ for track forecast task and and $c = 1$ for intensity task. The final layer makes the prediction.

To extract the spatial-temporal embedded features, we use the output of the second fully connected layer, of dimension 128. Therefore, this technique allows to reduce $8 \cdot 9 \cdot 25 \cdot 25 = 45,000$ features into 128 predictive features that can be input into our XGBoost models.

For each convolutional layer of the CNN, we use the following parameters: kernel size = 3, stride = 1, padding = 0. For each MaxPool layer, we use the following parameters: kernel

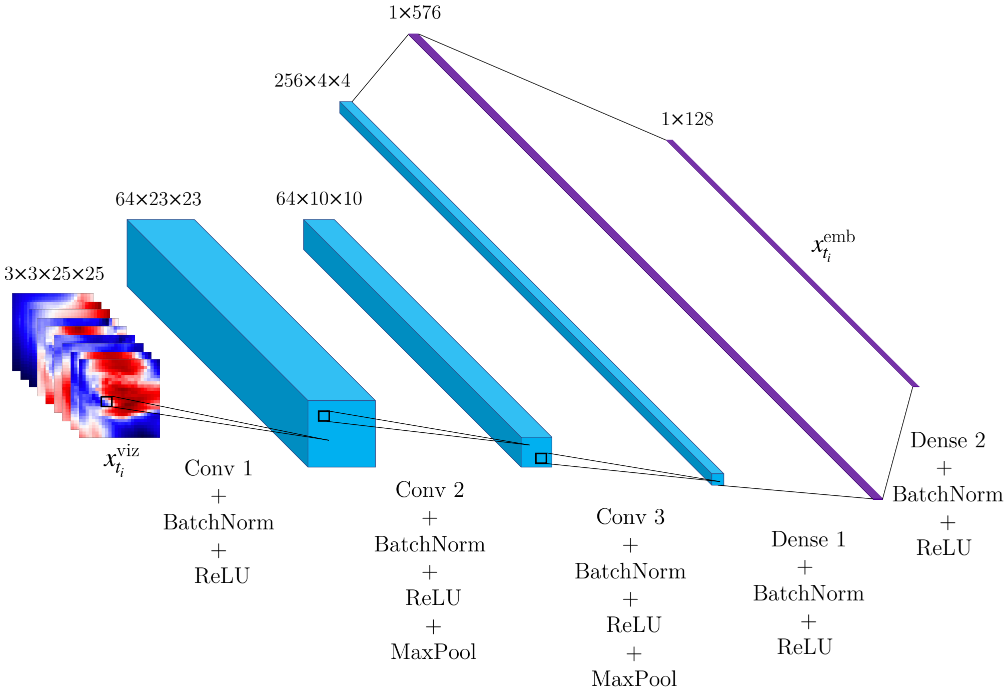

1×576
256×4×4
1×128
64×23×23
64×10×10
$x_{t_i}^{\text{emb}}$
3×3×25×25
$x_{t_i}^{\text{viz}}$
Conv 1 + BatchNorm + ReLU
Conv 2 + BatchNorm + ReLU + MaxPool
Conv 3 + BatchNorm + ReLU + MaxPool
Dense 1 + BatchNorm + ReLU
Dense 2 + BatchNorm + ReLU

Figure A.1: Representation of our CNN-encoder. We use 3 convolutional layers, with batch normalization, ReLU and MaxPool in between. We use fully connected (dense) layers to obtain in the end a one-dimensional vector $x_{t_i}^{\text{emb}}$.

size = 2, stride = 2, padding = 0.

The CNN-encoder architecture is inspired from [19]. The combination with the GRU-decoder or Transformer-decoder and the feature extraction is a contribution of our work.

A.1.3 Technical Details on the Transformer-Decoder Architecture

As with the CNN-encoder GRU-decoder network, the spatial-temporal inputs are processed and concatenated with the statistical data to obtain a sequence of input $[\mathbf{x}_{t_i}^{emb}, \mathbf{x}_{t_i}^{stat}], \forall i \in \{0, ..., 7\}$. As suggested by [38], we add to each $[\mathbf{x}_{t_i}^{emb}, \mathbf{x}_{t_i}^{stat}]$ input a positional encoding $\mathbf{P}_i$ token in order to provide some information about the relative position i within the sequence. We eventually obtain $\mathbf{x} = [\mathbf{x}_{t_i}^{emb}, \mathbf{x}_{t_i}^{stat}] + \mathbf{P}_i$ which is being processed by the Transformer's layers. In this work, we use $P_{i,2j} = \sin(i/10000^{2j/d})$and $P_{i,2j+1} = \cos(i/10000^{2j/d})$, where

i is the position in the sequence, j the dimension and d the dimension of the model, in our case 142. A layer is composed of a multi-head attention transformation followed by a fully-connected layer, similar to the Transformer's encoder presented in [38].

We used self-attention layers (i.e., $Q = K = V$), specifically 2 layers with 2 heads, the model's dimension d_k being fixed to 142 and the feedforward dimension set to 128.

We then averaged the outputs of our Transformer $\mathbf{h}_{t_0}, \ldots, \mathbf{h}_{t_7}$ feature-wise to obtain the final representation of the sequence.

A.2 Tucker Decomposition for Tensors

The multilinear singular value decomposition (SVD) expresses a tensor $\mathcal{A}$ as a small core tensor $\mathcal{S}$ multiplied by a set of unitary matrices. The size of the core tensor, denoted by $[k_1, \ldots k_N]$, defines the rank of the tensor.

Formally, the multilinear decomposition can be expressed as:

$$\begin{aligned} \mathcal{A} &= \mathcal{S} \times_1 U^{(1)} \times_2 \cdots \times_N U^{(N)}, \\ \text{where } \mathcal{A} &\in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}, \\ \mathcal{S} &\in \mathbb{R}^{k_1 \times k_2 \times \cdots \times k_N}, \\ U^{(i)} &\in \mathbb{R}^{I_i \times k_i}, \end{aligned}$$

where each $U^{(i)}$ is a unitary matrix, i.e., its conjugate transpose is its inverse $U^{(i)*}U^{(i)} = U^{(i)}U^{(i)*} = I$, and the mode-n product, denoted by $\mathcal{A} \times_n U$, denotes the multiplication operation of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by a matrix $U \in \mathbb{R}^{I_n \times J_n}$. Figure A2 exhibits a geometric representation of the Tucker decomposition applied to a three-dimensional tensor $\mathcal{A}$, which is decomposed as a smaller core tensor $\mathcal{S}$ and projection maps $U^i_{i=1,2,3}$. It can be viewed as a slice-wise matrix product along n-th dimension. Formally:

$$(A \times_n U)_{i_1 \ldots j_n \ldots i_N} = \sum_{i_n} a_{i_1 \ldots i_n \ldots i_N} u_{j_n i_n}$$

.

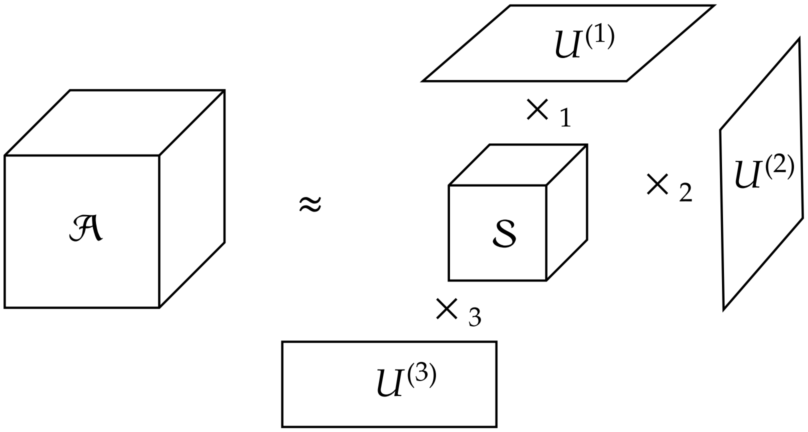


Figure A.2: Illustration of the tensor decomposition of a 3 dimensional tensor. Tensor $\mathcal{A}$ is the original tensor, which is approximated through Tucker decomposition using a core tensor tensor $\mathcal{S}$ and three linear projection maps along each axis $U^{(1)}, U^{(2)}, U^{(3)}$.

Analogous to truncated SVD, we can reduce the dimensionality of tensor $\mathcal{A}$ by artificially truncating the core tensor $\mathcal{S}$ and corresponding $U^{(i)}$. For instance, given a 4-dimensional tensor of TC maps, we can decide to reduce the tensor to any desired rank by keeping only the desired size of core tensor $\mathcal{S}$. For instance, to reduce TC tensor data into rank $3 \times 5 \times 3 \times 3$, we first perform multilinear SVD, such that S reflects descending order of the singular values, and then truncate $\mathcal{S}$ by keeping only the first $3 \times 5 \times 3 \times 3$ entries, denoted by $\mathcal{S}'$, and the first 3 columns of each of $U^{(i)}$, denoted by $U'^{(i)}$. Finally, a compressed tensor $\mathcal{A}'$ can be expressed as:

$$\mathcal{A}' = \mathcal{S}' \times_1 U'^{(1)} \times_2 \cdots \times_N U'^{(N)}.$$

Finally, we flatten the truncated core tensor $\mathcal{S}'$ into a vector, which is treated as the extracted vision features in order to train the XGBoost model.

A.3 Experiment Details

A.3.1 Testing Methodology

We employed the validation set to perform hyperparameter tuning. Then, we retrained the models on the training and validation set combined using the best combination of hyperparameters. We then evaluate our models' performance on the test set.

We report the performance obtained on the NA and EP test set with each method for 24-hour lead time for both intensity and track forecasts. As a remark, in reality, there is often a time lag when operational models become available. Such lag is shorter for statistical models but longer for dynamical models (up to several hours) because of expensive computational time. Due to the lag time variability, we do not consider such lag in our comparisons with operational models. In other words, we neglect the time lag for all models and compare model results assuming all forecasts compute instantaneously. We hope to provide an overall sense of the predictive power of our methodology, although we acknowledge that using reanalysis maps data is not possible in real-time. We discussed this bottleneck in section 2.6.

A.3.2 The Specific Protocol for HUML-ensemble

For the HUML-ensemble model, we used the HUML models 1-4 trained on the training set only (i.e., data until 2011). We then used their forecasts on the unseen validation set (2012 to 2015) and their forecasts on the unseen test set (2016 to 2019) as the training and testing data for the ensemble. The goal is to understand how each model behaves with respect to the others on unseen data. We cross-validated the ElasticNet parameters on the 2012-2015 HUML forecasts and we finally tested on the same cases as before using the best hyperparameter combination found.

A.3.3 Hyperparameter Tuning

We distinguish five categories of hyperparameters to tune: (1) the data-related features, (2) the neural-network related features, (3) the tensor decomposition-related features, (4) the tree-based method related features, (5) the consensus models-related features.

Data-related features

The data-related features include the area covered by the reanalysis maps (grid size) and the number of historical time steps of data to use for each forecast. We tune these features by comparing the 24-hour lead time forecast performance of the encoder-decoders for each different hyperparameter configuration.

We found that using eight past time steps (i.e., up to 21 hours in the past) and a grid size of 25×25 degrees for the reanalysis maps was the best combination. We also found that standardizing the vision and statistical data — i.e., rescaling each feature to mean 0 and standard deviation 1 — yielded better results than normalizing — i.e., rescaling each feature to the $[0, 1]$ range.

Neural network-related features

The neural network-related features include the optimizer, the architecture itself, the batch size during training, and the loss function's regularizer.

The best results were obtained using a batch size of 64, a λ regularization term of 0.01, and the encoder-decoder architectures described previously. Regarding the optimizer, we use Adam [118] with a learning rate of 10^{-3} for the intensity forecast and $4 \cdot 10^{-4}$ for the track forecast.

Tensor decomposition features

The tensor decomposition algorithm includes the choice of the core tensor size, i.e., the compressed size of the original tensor. Recall that the original tensor size is $8 \times 9 \times 25 \times 25$.

Based on empirical testing, we found using a small tensor size of $3 \times 5 \times 3 \times 3$ yielded the best performance when compressed reanalysis maps are included as features in XGBoost models.

Tree-based method features

Based on empirical testing, we found XGBoost models consistently outperforming Decision Trees and Random Forests or other ML methods such as Support Vector Machines, Regularized Linear Regression and Multi-Layer Perceptrons. XGBoost trains also fast which is a considerable advantage for heavy hyperparameter search. Therefore, we selected XGBoost as the core model for prediction.

Then, there is variability in the best combinations of hyperparameters depending on each task (track or intensity), basin (NA or EP) or data sources to use (statistical, various reanalysis maps embeddings). However, these particular features were typically important and were the best in the following ranges: maximum depth of the trees (between 6 and 9), number of estimators (between 100 and 300), learning rate (between 0.03 and 0.15), subsample (between 0.6 and 0.9), column sampling by tree (between 0.7 and 1), minimum child by tree (between 1 and 5).

Consensus-models-related features

We tested different kinds of consensus models on the HUML forecasts, including ElasticNet [119], tree-based models, and multi-layer perceptrons (MLPs) as meta-learners. MLPs had similar performance with ElasticNet, but since they are less interpretable and stable, ElasticNet is the strongest ensembler candidate and our final choice for HUML-ensemble. We tune the L1/L2 ratio between 0 and 1 and the regularization penalty between 10^{-4} and 10.

A.3.4 Metrics

Haversine Formula

Formally, the Haversine distance between one pair of predicted point and actual point, denoted by d, is calculated by:

$$\begin{aligned} d &= 2R \arcsin\left(\sqrt{\alpha}\right), \text{ where }, \\ \alpha &= \sin^2\left(\frac{\hat{\phi}-\phi}{2}\right) + \cos\left(\hat{\phi}\right)\cos\left(\phi\right)\sin^2\left(\frac{\hat{\lambda}-\lambda}{2}\right), \end{aligned}$$

where (ϕ, λ) are the actual latitude and longitude of one data point, $(\hat{\phi}, \hat{\lambda})$ are the predicted latitude and longitude, and R is Earth's radius, approximated to be the mean radius at 6,371 km.

Skill

Skill represents a normalization of the forecast error against a standard or baseline. We computed the skill s_f of a forecast f following [6]:

$$s_f(\%) = 100 \cdot \frac{e_b - e_f}{e_b},$$

where e_b is the error of the baseline model and e_f is the error of the forecast being evaluated. Skill is positive when the forecast error is smaller than the error from the baseline.

Appendix B

Appendix for Chapter 4

B.1 Optimization Model Parameter Estimation

In this section, we expand on how to estimate parameters for the optimization model.

B.1.1 Computing L_i^{CLT}

Recall from equation 4.16, to compute L_i^{CLT} we need to compute the historical mean and variance for each state. Table B.1 in the appendix exhibiting historical mean and standard deviation for the top 10 most costly states, computed on the annual basis.

State	Max	Mean	Std	Median
LA	3,763,390	45,084	58,467	20,583
TX	8,973,270	44,046	65,420	19,801
NJ	4,022,518	32,727	58,000	13,042
NY	9,467,720	35,725	74,510	13,306
FL	9,100,033	25,351	63,809	8,052
MS	10,000,000	46,603	94,924	14,823
NC	1,294,678	21,725	41,345	7,992
PA	1,889,793	19,040	41,943	6,939
AL	4,900,000	28,059	95,474	8,466
SC	1,764,000	25,574	43,752	10,088

Table B.1: Statistics of top 10 most costly states in the US, including maximum annual claim loss, mean, standard deviation and median.

B.1.2 Demand Damping

In this work, we model the demand sensitivity to insurance premium through a piece-wise linear demand function. We estimate the decline rate using historical data from several states. Figures B.1 and B.2 below shows the scatter plot of number of policy holders in a year against the mean policy premium of that state at that year. Different states have different degrees of sensitivity to price, but in general we observe a downward trend of decline in policy holder number as a function of increased price. For the illustrative purpose of this work, we do not specify different sensitivity in different states, but use the same demand damping function across all states.

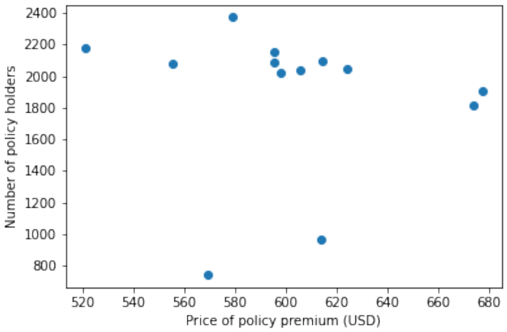


Figure B.1: (a) Demand damping estimation for Louisiana state (LA).

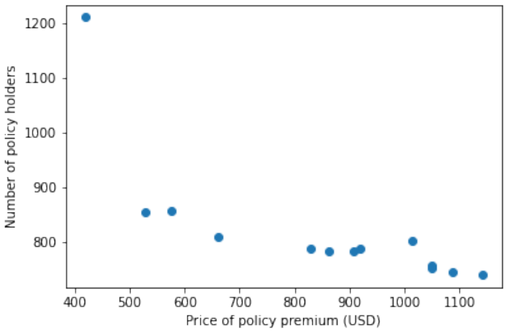


Figure B.2: (b) Demand damping estimation for New York state (NY).

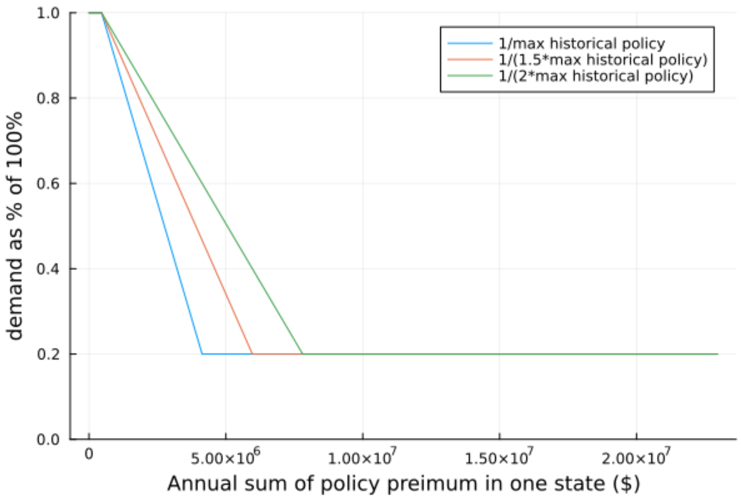


Figure B.3: Different piece-wise linear demand damping curves corresponding to different rate of decline.

B.2 Training Protocol of Machine Learning Models

B.2.1 Data Processing

Our dataset spans from 1975 to 2022. We exclude any claims lacking a specified claim amount to ensure the integrity of our analysis. Additionally, data from 'MP' (Northern Mariana Islands), 'AS' (American Samoa), 'GU' (Guam), and 'DC' (District of Columbia) have been omitted due to their limited data records. As a result, our cleaned dataset encompasses information from 52 jurisdictions over 48 years. This includes all 50 U.S. states, alongside two territories recognized as island states: the U.S. Virgin Islands and Puerto Rico, enhancing the geographical breadth of our study.

To construct the machine learning model, first, we aggregate data to state and annual level. The index of the data is two levels: state and year. For missing data, we performed linear interpolation within each state using previous and later years. Then for each state at a particular year, we construct the following features: state (categorical), current year annual loss, past 1-5 years annual loss. Current and past year losses are numerical features, where as state (categorical) feature is treated with one-hot encoding.

We train binary classification models to predict the target, with 1 indicating for a particular state at a particular year, the state will suffer an annual loss passing through the threshold Θ in the next 1 to K years. We have experimented with three threshold values, corresponding to 90th, 95th and 99th percentile annual claim amount values across all states over all training data, corresponding to USD 18,558,788, USD 50,688,672 and USD 321,903,271. In addition, we have experimented with three K values, corresponding to 3, 5, 10 years respectively.

B.2.2 Training and Testing Protocols

We split the data set chronologically into training period from 1975 to 2011, and testing period from 2012 to 2022. We experiment with two standard machine learning models, logistic regression and XGBoost. We employed 3-fold cross validation to search for the best parameters for each type of models. The search space for hyperparameter tuning can be found in Table B.2 below.

Model	**Hyperparameters**	**Values**
XGBoost	number of estimators	100, 150
	maximum tree depth	4, 6
	learning rate	0.1, 0.3
Logistic Regression	C	0, 0.2, 0.4, 0.6, 0.8, 1
	penalty	L1, L2

Table B.2: Hyperparameters searched for our models.

B.2.3 Detailed Prediction Results

Figure B.4 illustrates machine learning predicted risks for all states surpassing 90th percentile flooding risk within the next 5-year time frame on 2016. For each state, we produce one prediction for each year over the testing period, for each threshold, and for 3-year, 5-year, 10-year time frames. Table B.3 record out of sample prediction results using testing data, corresponding to data between 2012 to 2022. We treat data in the testing period on a rolling basis, and we drop the years where we do not have target data, i.e., in year 2019, we predict

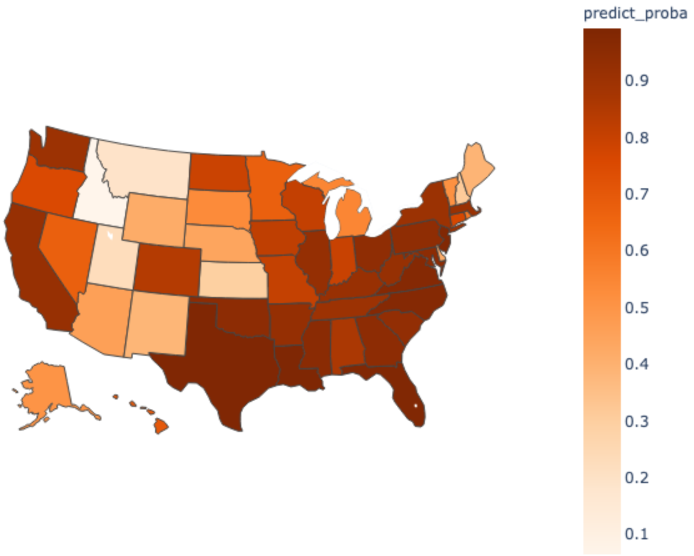


Figure B.4: Map illustrating machine learning (ML) predicted risks for all states surpassing 90th-percentile flooding risk within a 5-year time frame from 2016. Regions are color-coded, with darker shades indicating a higher probability of predicted risks.

for K=3 but not for K = 5 or 10. We remark that accuracy is generally higher for longer forecasting horizons. This is likely due to the following reasons: first, longer forecasting horizon lead to higher probability of flood, which leads to more balanced data; second, we have less testing samples. We use the probabilistic prediction results for each state at each testing year $q_{i,k}$ as input to construct uncertainty sets for the robust optimization model as given by equation 4.17.

B.2.4 Computational resources

The source codes for this study, implemented in Julia 1.7 and Python 3.9, are publicly accessible at [repository link]. The convex optimization problems were solved using Ipopt and Gurobi solvers, with machine learning models trained on a local machine with 4 Intel CPUs on a Macbook Pro personal computer. Comprehensive documentation detailing the methodology and specific parameters can be found in the repository's code comments.

Scores	3 Years		5 Years		10 Years	
	logreg	xgb	logreg	xgb	logreg	xgb
90% Threshold						
auc	0.743	0.776	0.763	0.808	0.767	0.912
f1	0.630	0.665	0.593	0.735	0.621	0.817
accu	0.693	0.675	0.632	0.736	0.623	0.830
accu_bl	0.625	0.706	0.605	0.753	0.641	0.809
precision	0.407	0.457	0.520	0.625	0.658	0.777
recall	0.437	0.793	0.362	0.901	0.500	0.967
95% Threshold						
auc	0.818	0.871	0.818	0.897	0.794	0.921
f1	0.684	0.673	0.700	0.744	0.764	0.763
accu	0.856	0.781	0.808	0.792	0.830	0.774
accu_bl	0.665	0.742	0.699	0.829	0.736	0.820
precision	0.300	0.306	0.368	0.460	0.595	0.560
recall	0.391	0.688	0.516	0.891	0.500	0.938
99% Threshold						
auc	0.920	0.945	0.922	0.956	0.952	0.992
f1	0.704	0.687	0.737	0.771	0.835	0.906
accu	0.910	0.950	0.909	0.943	0.925	0.962
accu_bl	0.809	0.646	0.786	0.731	0.958	0.979
precision	0.253	0.215	0.313	0.380	0.556	0.714
recall	0.696	0.304	0.640	0.480	1.000	1.000

Table B.3: Consolidated out-of-sample accuracy for percentile threshold predictions at 90%, 95%, and 99%, across different time horizons using logistic regression (logreg) and XGBoost (xgb) models.

References

[1] L. Boussioux, C. Zeng, T. Guénais, and D. Bertsimas, "Hurricane Forecasting: A Novel Multimodal Machine Learning Framework," *Weather and Forecasting*, vol. 37, no. 6, pp. 817–831, 2022.

[2] C. Zeng and D. Bertsimas, "Global Flood Prediction: A Multimodal Machine Learning Approach," *ICML Tackling Climate Change with Machine Learning Workshop*, 2023.

[3] D. Bertsimas and C. Zeng, "Catastrophe Insurance: A Robust Optimization Approach," *To be submitted to Management Science*,

[4] D. Bertsimas, L. Boussioux, and C. Zeng, "Reducing Air Pollution Through Machine Learning," *To be submitted to INFORMS Manufacturing & Service Operations Management*,

[5] A. Grinsted, P. Ditlevsen, and J. H. Christensen, "Normalized us hurricane damage estimates using area of total destruction, 1900-2018," *Proceedings of the National Academy of Sciences*, vol. 116, no. 48, pp. 23 942–23 946, 2019, ISSN: 0027-8424.

[6] J. P. Cangialosi, "National hurricane center forecast verification report," *National Hurricane Center*, 2020. URL: https://www.nhc.noaa.gov/verification/pdfs/Verification_2020.pdf.

[7] ECWMF, "Part iii: Dynamics and numerical procedures," in *IFS Documentation CY46R1* (IFS Documentation 3), IFS Documentation 3. ECMWF, 2019. URL: https://www.ecmwf.int/node/19307.

[8] S. D. Aberson, "Five-day tropical cyclone track forecasts in the north atlantic basin," *Weather and Forecasting*, vol. 13, no. 4, pp. 1005–1015, 1998.

[9] J. Knaff, M. DeMaria, C. Sampson, and J. Gross, “Statistical 5-day tropical cyclone intensity forecasts derived from climatology and persistence,” *Weather and Forecasting*, vol. 18, pp. 80–92, 2003.

[10] M. DeMaria, M. Mainelli, L. K. Shay, J. A. Knaff, and J. Kaplan, “Further improvements to the statistical hurricane intensity prediction scheme (ships),” *Weather and Forecasting*, vol. 20, no. 4, pp. 531–543, 2005.

[11] C. R. Sampson, J. L. Franklin, J. A. Knaff, and M. DeMaria, “Experiments with a simple tropical cyclone intensity consensus,” *Weather and Forecasting*, vol. 23, no. 2, pp. 304–312, 2008.

[12] A. Simon, A. B. Penny, M. DeMaria, J. L. Franklin, R. J. Pasch, E. N. Rappaport, and D. A. Zelinsky, “A description of the real-time hfip corrected consensus approach (hcca) for tropical cyclone track and intensity guidance,” *Weather and Forecasting*, vol. 33, no. 1, pp. 37–57, 2018.

[13] J. P. Cangialosi, E. Blake, M. DeMaria, A. Penny, A. Latto, E. Rappaport, and V. Tallapragada, “Recent Progress in Tropical Cyclone Intensity Forecasting at the National Hurricane Center,” *Weather and Forecasting*, pp. 1–30, Jul. 2020, ISSN: 0882-8156.

[14] M. Moradi Kordmahalleh, M. Gorji Sefidmazgi, and A. Homaifar, “A sparse recurrent neural network for trajectory prediction of atlantic hurricanes,” in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, 2016, pp. 957–964.

[15] S. Gao, P. Zhao, B. Pan, Y. Li, M. Zhou, J. Xu, S. Zhong, and Z. Shi, “A nowcasting model for the prediction of typhoon tracks based on a long short term memory neural network,” *Acta Oceanologica Sinica*, vol. 37, pp. 8–12, 2018.

[16] S. Alemany, J. Beltran, A. Perez, and S. Ganzfried, “Predicting hurricane trajectories using a recurrent neural network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 468–475.

[17] M. Mudigonda, S. Kim, A. Mahesh, S. Kahou, K. Kashinath, D. Williams, V. Michalski, T. O'Brien, and M. Prabhat, "Segmenting and tracking extreme climate events using neural networks," 2017.

[18] J. Lian, P. Dong, Y. Zhang, J. Pan, and K. Liu, "A novel data-driven tropical cyclone track prediction model based on cnn and gru with multi-dimensional feature selection," *IEEE Access*, 2020.

[19] S. Giffard-Roisin, M. Yang, G. Charpiat, C. Kumler Bonfanti, B. Kégl, and C. Monteleoni, "Tropical cyclone track forecasting using fused deep learning from aligned reanalysis data," *Frontiers in Big Data*, vol. 3, p. 1, 2020, ISSN: 2624-909X.

[20] R. Chen, X. Wang, W. Zhang, X. Zhu, A. Li, and C. Yang, "A hybrid cnn-lstm model for typhoon formation forecasting," *GeoInformatica*, 2019.

[21] H. Su, L. Wu, J. H. Jiang, R. Pai, A. Liu, A. J. Zhai, P. Tavallali, and M. DeMaria, "Applying satellite observations of tropical cyclone internal structures to rapid intensification forecast with machine learning," *Geophysical Research Letters*, vol. 47, no. 17, 2020.

[22] K. R. Knapp, M. C. Kruk, D. H. Levinson, H. J. Diamond, and C. J. Neumann, *The international best track archive for climate stewardship (ibtracs): Unifying tropical cyclone best track data*, 2010.

[23] M. DeMaria and J. Kaplan, "A statistical hurricane intensity prediction scheme (ships) for the atlantic basin," *Weather and Forecasting*, vol. 9, no. 2, pp. 209–220, 1994.

[24] B. Harper, J. Kepert, and J. Ginger, "Guidelines for converting between various wind averaging periods in tropical cyclone conditions," *Geneva, Switzerland: WMO, 2010*, 2010. URL: https://library.wmo.int/doc_num.php?explnum_id=290.

[25] H. Hersbach, B. Bell, P. Berrisford, *et al.*, "The era5 global reanalysis," *Quarterly Journal of the Royal Meteorological Society*, vol. 146, no. 730, pp. 1999–2049, 2020.

[26] U. Shimada, H. Owada, M. Yamaguchi, T. Iriguchi, M. Sawada, K. Aonashi, M. DeMaria, and K. D. Musgrave, "Further Improvements to the Statistical Hurricane Intensity Prediction Scheme Using Tropical Cyclone Rainfall and Structural Features," *Weather and Forecasting*, vol. 33, no. 6, pp. 1587–1603, Nov. 2018, ISSN: 0882-8156.

[27] B. A. Schenkel and R. E. Hart, "An examination of tropical cyclone position, intensity, and intensity life cycle within atmospheric reanalysis datasets," *Journal of Climate*, vol. 25, no. 10, pp. 3453–3475, 2012.

[28] K. Hodges, A. Cobb, and P. L. Vidale, "How well are tropical cyclones represented in reanalysis datasets?" *Journal of Climate*, vol. 30, no. 14, pp. 5243–5264, 2017.

[29] G.-F. Bian, G.-Z. Nie, and X. Qiu, "How well is outer tropical cyclone size represented in the era5 reanalysis dataset?" *Atmospheric Research*, vol. 249, p. 105 339, 2021, ISSN: 0169-8095.

[30] C. Sampson and A. J. Schrader, "The automated tropical cyclone forecasting system (version 3.2)," *Bulletin of the American Meteorological Society*, vol. 81, pp. 1231–1240, 2000.

[31] National Hurricane Center, *Automated tropical cyclone forecasting system (atcf)*, Accessed: 2021-04-06, 2021. URL: https://ftp.nhc.noaa.gov/atcf/.

[32] T. Burg and S. P. Lillo, "Tropycal: A python package for analyzing tropical cyclones and more," in *34th Conference on Hurricanes and Tropical Meteorology*, AMS, 2020.

[33] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: ACM, 2016, pp. 785–794, ISBN: 978-1-4503-4232-2.

[34] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. arXiv: 1412.3555.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5998–6008.

[39] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2015.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016, ISBN: 0262035618.

[42] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *CoRR*, vol. abs/1411.1792, 2014.

[43] D. Kiela and L. Bottou, "Learning image embeddings using convolutional neural networks for improved multi-modal semantics," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Oct. 2014, pp. 36–45.

[44] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[45] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *CoRR*, vol. abs/1808.01974, 2018. eprint: 1808.01974.

[46] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[47] G. Van Rossum and F. L. Drake Jr, *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[48] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035.

[49] T. Hamill, G. Bates, J. Whitaker, D. Murray, M. Fiorino, T. Galarneau, Y. Zhu, and W. Lapenta, "Noaa's second-generation global medium-range ensemble reforecast dataset," *Bulletin of the American Meteorological Society*, vol. 94, pp. 1553–1565, Oct. 2013.

[50] United Nations, "Pakistan floods: 9 million more risk being pushed into poverty, warns UNDP," *UN News*, Mar. 12, 2017. URL: https://news.un.org/en/story/2023/01/1132207 (visited on 01/05/2023).

[51] O. E. J. Wing, W. Lehman, P. D. Bates, C. C. Sampson, N. Quinn, A. M. Smith, J. C. Neal, J. R. Porter, and C. Kousky, "Inequitable patterns of US flood risk in the Anthropocene," en, *Nature Climate Change*, vol. 12, no. 2, pp. 156–162, Feb. 2022, ISSN: 1758-6798. DOI: 10.1038/s41558-021-01265-6. URL: https://www.nature.com/articles/s41558-021-01265-6 (visited on 10/27/2022).

[52] Y. Hirabayashi, R. Mahendran, S. Koirala, L. Konoshima, D. Yamazaki, S. Watanabe, H. Kim, and S. Kanae, "Global flood risk under climate change," en, *Nature Climate Change*, vol. 3, no. 9, pp. 816–821, Sep. 2013, Number: 9 Publisher: Nature Publishing Group, ISSN: 1758-6798. DOI: 10.1038/nclimate1911. URL: https://www.nature.com/articles/nclimate1911 (visited on 10/28/2022).

[53] A. Kauffeldt, F. Wetterhall, F. Pappenberger, P. Salamon, and J. Thielen, “Technical review of large-scale hydrological models for implementation in operational flood forecasting schemes on continental level,” en, *Environmental Modelling & Software*, vol. 75, pp. 68–76, Jan. 2016, ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2015.09.009. URL: https://www.sciencedirect.com/science/article/pii/S1364815215300529 (visited on 12/15/2022).

[54] C. C. Sampson, A. M. Smith, P. D. Bates, J. C. Neal, L. Alfieri, and J. E. Freer, “A high-resolution global flood hazard model,” *Water resources research*, vol. 51, no. 9, pp. 7358–7381, 2015.

[55] J. Thielen, J. Bartholmes, M.-H. Ramos, and A. De Roo, “The european flood alert system–part 1: Concept and development,” *Hydrology and Earth System Sciences*, vol. 13, no. 2, pp. 125–140, 2009.

[56] A. Mosavi, P. Ozturk, and K.-w. Chau, “Flood Prediction Using Machine Learning Models: Literature Review,” en, *Water*, vol. 10, no. 11, p. 1536, Nov. 2018, ISSN: 2073-4441. DOI: 10.3390/w10111536. URL: https://www.mdpi.com/2073-4441/10/11/1536 (visited on 09/20/2022).

[57] F. Sajedi-Hosseini, A. Malekian, B. Choubin, O. Rahmati, S. Cipullo, F. Coulon, and B. Pradhan, “A novel machine learning-based approach for the risk assessment of nitrate groundwater contamination,” en, *Science of The Total Environment*, vol. 644, pp. 954–962, Dec. 2018, ISSN: 0048-9697. DOI: 10.1016/j.scitotenv.2018.07.054. URL: https://www.sciencedirect.com/science/article/pii/S0048969718325373 (visited on 10/28/2022).

[58] B. Choubin, G. Zehtabian, A. Azareh, E. Rafiei-Sardooi, F. Sajedi-Hosseini, and Ö. Kişi, “Precipitation forecasting using classification and regression trees (CART) model: A comparative study of different approaches,” en, *Environmental Earth Sciences*, vol. 77, no. 8, p. 314, Apr. 2018, ISSN: 1866-6299. DOI: 10.1007/s12665-018-7498-z. URL: https://doi.org/10.1007/s12665-018-7498-z (visited on 10/28/2022).

[59] S. H. Elsafi, “Artificial Neural Networks (ANNs) for flood forecasting at Dongola Station in the River Nile, Sudan,” en, *Alexandria Engineering Journal*, vol. 53, no. 3,

pp. 655–662, Sep. 2014, ISSN: 1110-0168. DOI: 10.1016/j.aej.2014.06.010. URL: https://www.sciencedirect.com/science/article/pii/S1110016814000660 (visited on 10/28/2022).

[60] G. Kim and A. P. Barros, "Quantitative flood forecasting using multisensor data and neural networks," en, *Journal of Hydrology*, vol. 246, no. 1, pp. 45–62, Jun. 2001, ISSN: 0022-1694. DOI: 10.1016/S0022-1694(01)00353-5. URL: https://www.sciencedirect.com/science/article/pii/S0022169401003535 (visited on 10/28/2022).

[61] J. A. de Bruijn, H. de Moel, A. H. Weerts, M. C. de Ruiter, E. Basar, D. Eilander, and J. C. Aerts, "Improving the classification of flood tweets with contextual hydrological information in a multimodal neural network," *Computers & Geosciences*, vol. 140, p. 104 485, 2020.

[62] E. Rosvold and H. Buhaug, *Geocoded Disasters (GDIS) Dataset*, 2021. DOI: 10.7927/ZZ3B-8Y61. URL: https://sedac.ciesin.columbia.edu/data/set/pend-gdis-1960-2018 (visited on 10/28/2022).

[63] *EM-DAT The International Disaster Dataset*, 2021. URL: https://www.emdat.be/.

[64] *Wikipedia-API*. URL: https://pypi.org/project/Wikipedia-API/.

[65] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[66] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li, "On the sentence embeddings from pre-trained language models," *arXiv preprint arXiv:2011.05864*, 2020.

[67] A. Reuther, J. Kepner, C. Byun, *et al.*, "Interactive supercomputing on 40,000 cores for machine learning and data analysis," in *2018 IEEE High Performance extreme Computing Conference (HPEC)*, 2018, pp. 1–6. DOI: 10.1109/HPEC.2018.8547629.

[68] M. K. Van Aalst, "The impacts of climate change on the risk of natural disasters," *Disasters*, vol. 30, no. 1, pp. 5–18, 2006.

[69] J. Mercer, "Disaster risk reduction or climate change adaptation: Are we reinventing the wheel?" *Journal of International Development: The Journal of the Development Studies Association*, vol. 22, no. 2, pp. 247–264, 2010.

[70] J. Linnerooth-Bayer and S. Hochrainer-Stigler, “Financial instruments for disaster risk management and climate change adaptation,” *Climatic Change*, vol. 133, pp. 85–100, 2015.

[71] E. O. Michel-Kerjan, “Catastrophe economics: The national flood insurance program,” *Journal of economic perspectives*, vol. 24, no. 4, pp. 165–186, 2010.

[72] D. P. Horn and J. T. Brown, *Introduction to the national flood insurance program (nfip)*. Congressional Research Service Washington DC, USA, 2017.

[73] E. Michel-Kerjan and H. Kunreuther, “Redesigning Flood Insurance,” *Science*, vol. 333, no. 6041, pp. 408–409, Jul. 2011, Publisher: American Association for the Advancement of Science. DOI: 10.1126/science.1202616. (visited on 02/02/2023).

[74] C. Flavelle, J. Cowan, and I. Penn, “Climate Shocks Are Making Parts of America Uninsurable. It Just Got Worse.,” *The New York Times*, May 2023, ISSN: 0362-4331. URL: https://www.nytimes.com/2023/05/31/climate/climate-change-insurance-wildfires-california.html (visited on 11/04/2023).

[75] P. Grossi, *Catastrophe modeling: a new approach to managing risk*. Springer Science & Business Media, 2005, vol. 25.

[76] P. J. Schneider and B. A. Schauer, “Hazus—its development and its future,” *Natural Hazards Review*, vol. 7, no. 2, pp. 40–44, 2006.

[77] H. Kunreuther, G. Heal, M. Allen, O. Edenhofer, C. B. Field, and G. Yohe, “Risk management and climate change,” *Nature Climate Change*, vol. 3, no. 5, pp. 447–450, May 2013, ISSN: 1758-6798. DOI: 10.1038/nclimate1740.

[78] D. Wang, K. Yang, and L. Yang, “Risk-averse two-stage distributionally robust optimisation for logistics planning in disaster relief management,” *International Journal of Production Research*, vol. 61, no. 2, pp. 668–691, 2023.

[79] D. Alem, A. Clark, and A. Moreno, “Stochastic network models for logistics planning in disaster relief,” *European Journal of Operational Research*, vol. 255, no. 1, pp. 187–206, 2016.

[80] J. Salmerón and A. Apte, "Stochastic optimization for natural disaster asset prepositioning," *Production and operations management*, vol. 19, no. 5, pp. 561–574, 2010.

[81] W. Yang, K. Xu, C. Ma, J. Lian, X. Jiang, Y. Zhou, and L. Bin, "A novel multi-objective optimization framework to allocate support funds for flash flood reduction based on multiple vulnerability assessment," *Journal of Hydrology*, vol. 603, p. 127 144, 2021.

[82] T. Liu, J. Shao, and X. Wang, "Funding allocations for disaster preparation considering catastrophe insurance," *Socio-Economic Planning Sciences*, vol. 84, p. 101 413, 2022.

[83] A. Ben-Tal, B. Do Chung, S. R. Mandala, and T. Yao, "Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains," *Transportation research part B: methodological*, vol. 45, no. 8, pp. 1177–1189, 2011.

[84] S. Zokaee, A. Bozorgi-Amiri, and S. J. Sadjadi, "A robust optimization model for humanitarian relief chain design under uncertainty," *Applied Mathematical Modelling*, vol. 40, no. 17-18, pp. 7996–8016, 2016.

[85] T. Ermolieva, T. Filatova, Y. Ermoliev, M. Obersteiner, K. M. de Bruijn, and A. Jeuken, "Flood Catastrophe Model for Designing Optimal Flood Insurance Program: Estimating Location-Specific Premiums in the Netherlands," en, *Risk Analysis*, vol. 37, no. 1, pp. 82–98, 2017, ISSN: 1539-6924. DOI: 10.1111/risa.12589. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/risa.12589 (visited on 05/17/2023).

[86] T. Ermolieva and Y. Ermoliev, "Modeling catastrophe risk for designing insurance systems," *Integrated Catastrophe Risk Modeling: Supporting Policy Processes*, pp. 29–52, 2013.

[87] H. Hong, G. A. Karolyi, and J. A. Scheinkman, "Climate Finance," *The Review of Financial Studies*, vol. 33, no. 3, pp. 1011–1023, Mar. 2020, ISSN: 0893-9454. DOI: 10.1093/rfs/hhz146. URL: https://doi.org/10.1093/rfs/hhz146 (visited on 11/02/2023).

[88] L. M. Bouwer and J. C. Aerts, “Financing climate change adaptation,” en, *Disasters*, vol. 30, no. 1, pp. 49–63, 2006, ISSN: 1467-7717. DOI: 10.1111/j.1467-9523.2006.00306.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9523.2006.00306.x (visited on 11/02/2023).

[89] S. Hochrainer-Stigler, R. Mechler, G. Pflug, and K. Williges, “Funding public adaptation to climate-related disasters. Estimates for a global fund,” *Global Environmental Change*, vol. 25, pp. 87–96, Mar. 2014, ISSN: 0959-3780. DOI: 10.1016/j.gloenvcha.2014.01.011. URL: https://www.sciencedirect.com/science/article/pii/S0959378014000259 (visited on 11/02/2023).

[90] E. Mills, “Synergisms between climate change mitigation and adaptation: An insurance perspective,” en, *Mitigation and Adaptation Strategies for Global Change*, vol. 12, no. 5, pp. 809–842, Jun. 2007, ISSN: 1573-1596. DOI: 10.1007/s11027-007-9101-x. URL: https://doi.org/10.1007/s11027-007-9101-x (visited on 11/02/2023).

[91] J. K. Auh, J. Choi, T. Deryugina, and T. Park, *Natural Disasters and Municipal Bonds*, Working Paper, Jul. 2022. DOI: 10.3386/w30280. URL: https://www.nber.org/papers/w30280 (visited on 11/02/2023).

[92] J. Fowles, G. Liu, and C. B. Mamaril, “Accounting for Natural Disasters: The Impact of Earthquake Risk on California Municipal Bond Pricing,” en, *Public Budgeting & Finance*, vol. 29, no. 1, pp. 68–83, 2009, ISSN: 1540-5850. DOI: 10.1111/j.1540-5850.2009.00924.x. (visited on 11/02/2023).

[93] R. Keucheyan, “Insuring climate change: New risks and the financialization of nature,” *Development and Change*, vol. 49, no. 2, pp. 484–501, 2018.

[94] D. Bertsimas, D. B. Brown, and C. Caramanis, *Robust Optimization.* Princeton University Press, 2011.

[95] D. Bertsimas and D. den Hertog, *Robust and Adaptive Optimization.* Dynamic Ideas, 2022.

[96] F. E. M. A. (FEMA), *OpenFEMA Dataset: FEMA NFIP Redacted Claims - v2*, data retrieved on March 1 2023, https://www.fema.gov/openfema-data-page/fima-nfip-redacted-claims-v2, 2023.

[97] F. E. M. A. (FEMA), *OpenFEMA Dataset: FIMA NFIP Redacted Policies - v1*, data retrieved on March 1 2023, https://www.fema.gov/openfema-data-page/fima-nfip-redacted-policies-v1, 2023.

[98] I. Manisalidis, E. Stavropoulou, A. Stavropoulos, and E. Bezirtzoglou, "Environmental and health impacts of air pollution: A review," *Frontiers in public health*, p. 14, 2020.

[99] B. Brunekreef and S. T. Holgate, "Air pollution and health," *The lancet*, vol. 360, no. 9341, pp. 1233–1242, 2002.

[100] A. L. Power, R. K. Tennant, R. T. Jones, Y. Tang, J. Du, A. T. Worsley, and J. Love, "Monitoring impacts of urbanisation and industrialisation on air quality in the anthropocene using urban pond sediments," *Frontiers in Earth Science*, vol. 6, p. 131, 2018.

[101] Z. Tong and K. M. Zhang, "The near-source impacts of diesel backup generators in urban environments," *Atmospheric Environment*, vol. 109, pp. 262–271, 2015, ISSN: 1352-2310. DOI: https://doi.org/10.1016/j.atmosenv.2015.03.020.

[102] Z. Tong, B. Yang, P. K. Hopke, and K. M. Zhang, "Microenvironmental air quality impact of a commercial-scale biomass heating system," en, *Environmental Pollution*, vol. 220, pp. 1112–1120, Jan. 2017, ISSN: 0269-7491. DOI: 10.1016/j.envpol.2016.11.025.

[103] J. Lelieveld, J. S. Evans, M. Fnais, D. Giannadaki, and A. Pozzer, "The contribution of outdoor air pollution sources to premature mortality on a global scale," *Nature*, vol. 525, no. 7569, pp. 367–371, 2015.

[104] C. Bellinger, M. S. Mohomed Jabbar, O. Zaïane, and A. Osornio-Vargas, "A systematic review of data mining and machine learning for air pollution epidemiology,"

en, *BMC Public Health*, vol. 17, no. 1, p. 907, Nov. 2017, ISSN: 1471-2458. DOI: 10.1186/s12889-017-4914-3. (visited on 01/25/2023).

[105] T. Madan, S. Sagar, and D. Virmani, "Air Quality Prediction using Machine Learning Algorithms –A Review," in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Dec. 2020, pp. 140–145. DOI: 10.1109/ICACCCN51052.2020.9362912.

[106] Doreswamy, H. K s, Y. Km, and I. Gad, "Forecasting Air Pollution Particulate Matter (PM2.5) Using Machine Learning Regression Models," en, *Procedia Computer Science*, Third International Conference on Computing and Network Communications (CoCoNet'19), vol. 171, pp. 2057–2066, Jan. 2020, ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.04.221.

[107] M. Castelli, F. M. Clemente, A. Popovič, S. Silva, and L. Vanneschi, "A Machine Learning Approach to Predict Air Quality in California," en, *Complexity*, vol. 2020, e8049504, Aug. 2020, Publisher: Hindawi, ISSN: 1076-2787. DOI: 10.1155/2020/8049504.

[108] D. Sanjeev, "Implementation of Machine Learning Algorithms for Analysis and Prediction of Air Quality," en, *International Journal of Engineering Research*, vol. 10, no. 03,

[109] K. Kumar and B. P. Pande, "Air pollution prediction with machine learning: A case study of Indian cities," en, *International Journal of Environmental Science and Technology*, May 2022, ISSN: 1735-2630. DOI: 10.1007/s13762-022-04241-5. URL: https://doi.org/10.1007/s13762-022-04241-5 (visited on 01/27/2023).

[110] J. Zhu and Q. Zeng, "A mathematical formulation for optimal control of air pollution," *Science in China Series D: Earth Sciences*, vol. 46, no. 10, pp. 994–1002, Oct. 2003, ISSN: 1862-2801. DOI: 10.1007/BF02959394.

[111] Y. Panagopoulos, C. Makropoulos, and M. Mimikou, "Decision support for diffuse pollution management," *Environmental Modelling & Software*, vol. 30, pp. 57–70, Apr. 2012. DOI: 10.1016/j.envsoft.2011.11.006.

[112] SwissAid, *Négociants suisses et engrais dangereux : Violations de droits humains au maroc*, 2018. URL: https://voir-et-agir.ch/content/uploads/2018/12/Rapport%5C_Maroc.pdf.

[113] S. Grange, *Technical note: Averaging wind speeds and directions*, Jun. 2014. DOI: 10.13140/RG.2.1.3349.2006.

[114] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992. DOI: https://doi.org/10.1016/S0893-6080(05)80023-1.

[115] M. Amram, J. Dunn, and Y. D. Zhuo, "Optimal policy trees," en, *Machine Learning*, vol. 111, no. 7, pp. 2741–2768, Jul. 2022, ISSN: 1573-0565. DOI: 10.1007/s10994-022-06128-5. URL: https://doi.org/10.1007/s10994-022-06128-5 (visited on 03/05/2023).

[116] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[117] L. Interpretable AI, *Interpretable ai documentation*, 2023. URL: https://www.interpretable.ai.

[118] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, Dec. 2014.

[119] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.