Machine learning and optimization algorithms and their applications in agriculture

by

Javad Ansarifar

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:
Lizhi Wang, Major Professor
Guiping Hu
William Beavis
Sotirios Archontoulis
Sigurdar Olafsson

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

Iowa State University

Ames, Iowa

2021

DEDICATION

I would like to dedicate my dissertation work to my wife, Faezeh, and parents for their patience and support to complete my research. I also dedicate this dissertation to many friends for their help and support on my research path.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

With immense appreciation, I would like to express my gratitude to the people who helped me to bring this study into success.

First, I would like to express my sincere gratitude to my advisor Dr. Lizhi Wang, my major professor, for his continuous support, patience, motivation, valuable guidance, and immense knowledge throughout my graduate study and research. His enormous guidance and assistance in the path of the dissertation inspired me to pursue my academic goals.

Additionally, I would like to acknowledge my committee members Dr. Guiping Hu, Prof. William Beavis, Dr. Sotirios Archontoulis, and Dr. Sigurdar Olafsson for their time and insights that they provided at all levels of my research.

Finally, topmost gratitude to my wife and my parents. You are always there for me. Doubtlessly, I would not be here without your love and spiritual supports throughout my life.

ABSTRACT

This dissertation is devoted to using machine learning and optimization algorithms to develop explainable machine learning and decision-making models and their applications in agriculture. This dissertation consists of four journal papers. The first three papers focus on formulating explainable machine learning, and the last one is about the decision-making model for the planting scheduling.

The first paper proposes three new algorithms for multi-effect and multi-way epistases detection. Epistases refer to the phenomenon of genetic interactions that plays a significant role in many scientific discoveries such as the breeding process, case-and-control studies, and genome-wide association studies. Deciphering the exact genetic interactions is challenging because of the combinatorial nature of the problem. These three models are developed to detect the interaction between a binary representation of the genetic information so that one guaranteeing global optimality and the other two being local optimization-oriented heuristics. The computational performance of the proposed models were compared with several state-of-the-art methods using a yeast data set.

In the second paper, a new explainable machine learning model named the interaction regression model is developed for crop yield prediction. Crop yield prediction is a challenging issue because of multitudinous variables, including genotype, environment, management, and complex interactions that affect crop yield performance explicitly or implicitly. We integrate the power of optimization, machine learning, and agronomic insight to develop this explainable model with three salient properties. First, by outperforming state-of-the-art predictive models, the proposed model achieves an error prediction of 8% or less in three Midwest states (Illinois, Indiana, and Iowa) in the US Corn Belt for both corn and soybean yield prediction. Second, it detects the environment by management interactions for corn and soybean that are insightful agronomically. Third, this model can quantify and break down the yield into contributions from weather, soil, management, and their interactions that allow agronomists to analyze the favorable and unfavorable yield factors.

The third paper develops a new predictive framework that integrates random forest and an optimization-based model for G $\times$ E interaction detection to predict crosses' yield performance. In the plant breeding process, the yield performance plays a significant role in selecting more productive and adaptable parents to the changing environments. The proposed framework integrates a random forest with a combinatorial optimization-based interaction-detection model and attempts to combine their strengths. This model consists of three main components; a random forest model that captures complex non-linear relationships between input and output variables, an interaction detection model that captures interactions among hybrid, location, and weather variables, and another random forest model that utilizes the interactions to augment the prediction performance of the first random forest model. This model won the first place in the 2020 Syngenta crop challenge in analytics.

The fourth paper concerns the planting time scheduling problem of different population seeds in the year-round breeding process so that there is a consistent harvest quantity. Although developing the breeding process and producing higher-quality crops ensures global food availability and security, they raise new logistical and productivity challenges for seed industries in the year-round breeding process due to the storage limitation. 2021 Syngenta crop challenge in analytics was launched to challenge participants to design an optimization model for the planting time scheduling of several population seeds so that weekly harvest quantity would be consistent at the lowest possible capacity. We address this problem with uncertainty weather information by developing a new hybrid framework that combines the weather time series model and optimization model to schedule the planting time. Comparison with actual planting time scheduling reveals that the developed models scheduled the seed population's planting time at the fewest number of weeks with a more consistent weekly harvest quantity.

CHAPTER 1. GENERAL INTRODUCTION

Feeding the rapidly growing population is one of the most critical challenges that agricultural systems face, especially because of the continuously changing climate [7]. A wide range of agricultural food production topics has been investigated to improve food production and security, including optimization of planting regime, sustainable farming practices, traits introgression, and modeling of plant physiology and ecology. Among these topics, developing a decision-making framework for a farming system based on crop yield prediction has received special attention among industry players, researchers, and academic actors. Hence, this dissertation focuses on integrating machine learning and optimization algorithms to develop explainable machine learning and decision-making models for agricultural applications. This dissertation is developed in the form of four journal papers. In the first paper, we design three new algorithms to detect interactions between variables with respect to minimize the prediction error. These interactions shed light on prediction problems by providing biological insights. In the second paper, the interaction regression model is developed to predict crop yield by combining the power of machine learning, optimization, and insights. In the next paper, we address the hybrids' yield performance prediction by combining a random forest with a combinatorial optimization-based interaction-detection model. The last paper optimizes the scheduling planting time of different population seeds in the year-round breeding process. These studies are introduced in more detail in the remainder of this chapter.

In the first paper, we design three new algorithms to detect multi-effect and multi-way epistases detection. Epistasis detection, the deciphering of genetic interactions, is key in many scientific discoveries such as understanding the relationship between genotype and phenotype, curing genetic diseases, and accelerating genetic adaptation of crops to changing environments. Epistasis detection's concept can be used for other purposes such as recommender systems, yield prediction that shows the importance of research challenges. Detecting exact combinations of genes that trigger the

interactions is known as one of the most mysterious yet essential research challenges in genetics due to the problem's combinatorial nature. Many methods only focus on two-way interactions when two genes or features trigger the interactions. Some of the most effective methods used machine learning, specifically factorization machines [16] and random forest [13, 15, 8] to detect high-order interactions. On the other hand, many studies have been developed for special case-and-control studies (classification problem) [9, 10, 5] and there are a few papers that applied interactions concept in regression problems [14].

We designed three new algorithms for multi-effect and multi-way epistases detection to guarantee global optimality and the other two being local optimization-oriented heuristics. They have three salient features. First, these algorithms minimize the root mean square error (RMSE) such that they guarantee to find either global or local optimal solutions. Second, the algorithms are less prone to overfitting problem because of their sparsity of modeling structure. Third, the maximally tolerable computation time option allows users to terminate the algorithm at a predefined deadline. These features specify the tradeoff between speed and quality of the solution. We cast the multi-effect and multi-way epistases detection problem as mixed-integer quadratic programming for the first approach. Because the first model has a combinatorial and nonlinear nature, we propose a local search algorithm as a more computationally tractable algorithm for solving the first formulation. For the third algorithm, we develop a heuristic algorithm that minimizes the validation RMSE rather than the training RMSE, making it less prone to overfitting. Also, this algorithm explores multiple local optima and avoid brute force enumeration within local neighborhoods. The computational performance of the proposed models was compared with several state-of-the-art methods using a yeast data set. Moreover, the results show that searching for global optimal interactions is extremely time-consuming. The heuristic algorithm is more effective and efficient in detecting close-to-optimal interactions.

In the second paper, the interaction regression model as a new explainable machine learning model is proposed to predict crop yield prediction. Crop yield prediction is a fundamental research question in plant biology in addressing food security, particularly by global climate change.

Agricultural sectors can optimize their economic and management decisions by understanding how genotype (G), environment (E), management (M), and their interactions (G×E×M) effect crop yield [2, 6, 4, 3]. This paper proposes a new explainable model by combining the power of optimization, machine learning, and agronomic insight. The core of this model is an optimization algorithm that detects the most revealing E and M features in yield prediction and their most pronounced interactions which are insightful agronomically. The iterative mechanism in the proposed model tries to select a subset of E and M features for crop yield prediction that are spatially and temporally robust and find the interactions between them. The robustness means that they should be consistently predictive of crop yield across all counties in all years. Moreover, the model can quantify and break down the yield into contributions from weather, soil, management, and their interactions because a multiple linear regression of these features and interactions shaped the prediction of the crop yield performances. These break down of features and their interactions allows agronomists to analyze the favorable and unfavorable yield factors.

To test the performance of the model, we consider a comprehensive case study of corn and soybean yield prediction in 293 counties of Illinois, Indiana, and Iowa from 2015 to 2018. We also compare the proposed model with eight other machine learning models to predict corn and soybean yield. It achieves an error prediction of 8% or less in three states in the US Corn Belt for both corn and soybean yield prediction by outperforming other machine learning models. The interaction regression model produces explainable insights, in particular interactions and total yield into contributions from weather, soil, management, and their interactions. Additionally, we evaluate our model's performance in terms of both temporal and spatial extrapolation by training the model using historical data from two states up to 2017 and applying it to predict corn yield in a third state for 2018. The result shows that it achieves an average error prediction of less than 10%.

In the third paper, we develop a new predictive model for predicting new hybrids' yield performance based on historical data of other combinations. One of the most challenging plant breeding process issues is selecting breeding parents for crosses [1]. Hence, breeders make the various hybrids with high-yield parents and plant them in multiple locations and weather to measure the hybrids'

yield performance. Then, they select the best-breeding parents for crosses. But empirical breeding processes, including selecting, mating, planting, and evaluating biparental combinations, are expensive, labor-intensive, and time-consuming. Therefore, plant breeders have developed decision making frameworks over artificial intelligence methods used for performance prediction and selection of promising breeding parents for hybridization. In 2020, Syngenta designed a new agronomic question to challenge participants by developing a predictive approach to predict the yield performance of inbred-tester combinations based on historical data of other hybrids. Syngenta released a dataset that included the historical yield performance of 294,128 corn hybrids through the crossing of 593 unique inbreds and 496 unique testers across multiple locations between 2016 and 2018. To address this challenge, we integrate the power of optimization and machine learning to develop this explainable model that can decipher G $\times$ E interaction. A new predictive framework with three components integrates random forest and a combinatorial optimization-based interaction-detection model to predict crosses' yield performance.

The first component of the model is a random forest model that tries to predict crosses' yield performance by constructing a multitude of trees. The random forest model has the capability to captures complex nonlinear relationships between input and output variables. Although the random forest can capture the interaction between hybrid, location, and weather variables, this model is ineffective in deciphering specific features interactions that have the most significant interactions because of its sampling method. Hence, we use an interaction detection model as the second component to augment the random forest's performance by strategically searching for G $\times$ E interactions. An optimization-based interaction detection model has the capability to capture interactions among hybrid, location, and weather variables. The detected G $\times$ E for this model has a linear relationship to crosses' yield performance. Therefore, the interaction detection model cannot find more complex nonlinear functions of interactions. To fit the nonlinear function on interactions, we use another random forest model as the third component to use the detected interactions from the second component to predict the first random forest's residual error. The third model augments the prediction performance of the first random forest model. Our computational results reveal that the proposed

model achieves a relative root-mean-square-error (RMSE) of 0.0869 for the validation data by outperforming other state-of-the-art models such as factorization machine, random forest, and extreme gradient boosting tree. The model can find G $\times$ E interactions that are potentially biologically insightful. This model won the first place in the 2020 Syngenta crop challenge in analytics.

In the fourth paper, we develop the optimization model for scheduling the planting time of different population seeds in the year-round breeding process. Recently, seed industries have been applied a wide range of analyses on farming systems, including optimization of farming systems, breeding processes, and operational processes to improve agriculture's productivity and sustainability. Although these data-driven strategies address global food availability and sustainability, they intensify logistical and productivity issues because of limited storage capacity and erratic weekly harvest quantities [12, 11]. On the other hand, optimizing management practices, including scheduling, irrigation, fertilizing, tilling, and harvesting, plays a significant role in addressing these new challenges. Hence, this decision-making framework must consider logistic and storage limitations, seed production processes, and environmental uncertainty during the year-round breeding process to lead to a reasonable and robust solution for the farming system.

The 2021 Syngenta crop challenge in analytics was designed to challenge participants to develop a decision-making framework for the planting time scheduling of several population seeds to ensure when crops are harvested, facilities are not over capacity and there is a consistent weekly harvest quantity. They released a dataset that contained the historical weather information (growing degree units) from 2009 to 2019 and 2569 population seeds information, including their planting site, planting windows, the required number of growing degree units in Celsius needed for the harvest, and harvest quantity. For this challenge, there are two cases. In the first cases, we know the capacities. This challenge's objective function for this case is to optimize the population seed's planting time by harvesting at fewer numbers of the week, having consistent weekly harvest quantity, and satisfying capacity limitation constraint. There is no capacity limitation in another case, and the objective is to optimize the population seed's planting time with consistent weekly harvest quantity at the lowest possible capacity. To address this challenge, we designed a new hybrid model consists of a

weather time series model and a mathematical programming model to schedule the seed populations' planting time in the year-round breeding process under weather uncertainty. To predict the weather in the future, we construct a deep recurrent neural network. The uncertainty of forecasted weather is model using a Gaussian process model on top of the time-series model. We create several weather scenarios by sampling from the Gaussian distribution via Monte Carlo rollouts. Then, the proposed optimization model schedules the seed population's planting time at the fewest number of weeks by maximizing weekly harvest quantity consistency under all weather scenarios. The daily benchmark of weather prediction performance of the proposed deep recurrent neural network and other deep learning models for test years (2015-2019) illustrates that the proposed time-series model outperformed other machine learning models for all test years. Moreover, the modeling uncertainty of forecasted weather creates the weather scenarios into the future with meaningful trajectories. The optimization model at case one reduces the required capacity by 69% at site 0 and 48% at site 1 compared to the original planting time at the fewer harvesting weeks. For case two, we determine the minimum required capacities by decreasing the capacity by 69% at site 0 and by 51% at site 1. Then, the mathematical programming model schedules the planting times of seed populations at the lower harvesting weeks.

1.1 References

[1] Bertan, I., Carvalho, F., and Oliveira, A. d. (2007). Parental selection strategies in plant breeding programs. *Journal of Crop Science and Biotechnology*, 10(4):211–222.

[2] Cooper, M., Tang, T., Gho, C., Hart, T., Hammer, G., and Messina, C. (2020). Integrating genetic gain and gap analysis to predict improvements in crop productivity. *Crop Science*.

[3] Dai, Z. and Li, Y. (2013). A multistage irrigation water allocation model for agricultural land-use planning under uncertainty. *Agricultural Water Management*, 129:69–79.

[4] Filippi, C., Mansini, R., and Stevanato, E. (2017). Mixed integer linear programming models for optimal crop selection. *Computers & Operations Research*, 81:26–39.

[5] Hardison, N. E. and Motsinger-Reif, A. A. (2011). The power of quantitative grammatical evolution neural networks to detect gene-gene interactions. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pages 299–306. ACM.

[6] Hipólito, J., Boscolo, D., and Viana, B. F. (2018). Landscape and crop management strategies to conserve pollination services and increase yields in tropical coffee farms. *Agriculture, Ecosystems & Environment*, 256:218–225.

[7] Huai, J. (2017). Dynamics of resilience of wheat to drought in australia from 1991–2010. *Scientific Reports*, 7(1):9532.

[8] Lin, H.-Y., Ann Chen, Y., Tsai, Y.-Y., Qu, X., Tseng, T.-S., and Park, J. Y. (2012). Trm: A powerful two-stage machine learning approach for identifying snp-snp interactions. *Annals of Human Genetics*, 76(1):53–62.

[9] Motsinger, A. A., Dudek, S. M., Hahn, L. W., and Ritchie, M. D. (2006). Comparison of neural network optimization approaches for studies of human genetics. In *Workshops on Applications of Evolutionary Computation*, pages 103–114. Springer.

[10] Ritchie, M. D., Motsinger, A. A., Bush, W. S., Coffey, C. S., and Moore, J. H. (2007). Genetic programming neural networks: A powerful bioinformatics tool for human genetics. *Applied Soft Computing*, 7(1):471–479.

[11] Tilman, D., Balzer, C., Hill, J., and Befort, B. L. (2011). Global food demand and the sustainable intensification of agriculture. *Proceedings of the National Scademy of Sciences*, 108(50):20260–20264.

[12] Twine, T. E., Kucharik, C. J., and Foley, J. A. (2004). Effects of land cover change on the energy and water balance of the mississippi river basin. *Journal of Hydrometeorology*, 5(4):640–655.

[13] Uppu, S. and Krishna, A. (2018). A deep hybrid model to detect multi-locus interacting snps in the presence of noise. *International Journal of Medical Informatics*, 119:134–151.

[14] Wang, L. and Mehr, M. N. (2018). An optimization approach to epistasis detection. *European Journal of Operational Research*.

[15] Yoshida, M. and Koike, A. (2011). Snpinterforest: a new method for detecting epistatic interactions. *BMC Bioinformatics*, 12(1):469.

[16] Yurochkin, M., Nguyen, X., et al. (2017). Multi-way interacting regression via factorization machines. In *Advances in Neural Information Processing Systems*, pages 2598–2606.

CHAPTER 2. NEW ALGORITHMS FOR DETECTING MULTI-EFFECT AND MULTI-WAY EPISTATIC INTERACTIONS

A paper accepted by *Bioinformatics*

Javad Ansarifar and Lizhi Wang

2.1 Abstract

Epistasis, which is the phenomenon of genetic interactions, plays a central role in many scientific discoveries. However, due to the combinatorial nature of the problem, it is extremely challenging to decipher the exact combinations of genes that trigger the epistatic effects. Many existing methods only focus on two-way interactions. Some of the most effective methods used machine learning techniques, but many were designed for special case-and-control studies or suffer from overfitting. We propose three new algorithms for multi-effect and multi-way epistases detection, with one guaranteeing global optimality and the other two being local optimization oriented heuristics. The computational performance of the proposed heuristic algorithm was compared with several state-of-the-art methods using a yeast data set. Results suggested that searching for the global optimal solution could be extremely time consuming, but the proposed heuristic algorithm was much more effective and efficient than others at finding a close-to-optimal solution. Moreover, it was able to provide biological insight on the exact configurations of epistases, besides achieving a higher prediction accuracy than the state-of-the-art methods.

2.2 Introduction

Detecting epistatic interactions remains one of the most mysterious yet important research challenges in genetics, which holds the key to many scientific discoveries such as understanding the relationship between genotype and phenotype, curing genetic diseases, and accelerating genetic

adaptation of crops to changing environments. For example, Combarros et al. [3] and Gusareva et al. [9] used genome-wide association studies to analyze epistases for Alzheimer's disease; ; Ritchie et al. [21] identified three significant genes in sporadic breast cancer; Taylor and Ehrenreich [28] detected and identified significant genes that contribute to a complex trait in yeast.

Numerous approaches have been proposed to decipher epistatic interactions, most of which used genetic algorithms or machine learning methods. For example, Guan et al. [8] developed an ant colony optimization algorithm for epistasis detection; ; Rekaya and Robbins [20] and Sapin et al. [23] integrated ant colony optimization algorithm with logistic regression and decision tree and contingency table models. Machine learning based methods include support vector machine [2, 17, 25, 5, 40], neural networks [16, 22, 11], Bayesian networks [42, 27, 10], nonparametric Bayesian [43], factorization machines [39], random forest [29, 37, 12, 24, 15], among others [32, 33, 14, 35]. Reviews of machine learning approaches can be found in Upstill-Goddard et al. [30] and Koo et al. [13].

Despite significant previous work, the scientific community is still in need of more advanced approaches to decipher the epistatic effects hidden in many forms of datasets [18]. Most previous approaches could only detect two-way rather than multi-way interactions [4, 19, 6, 38, 7, 31, 41, 26, 36]. Many machine learning based approaches were classification algorithms designed for case-and-control studies with disease datasets, and not applicable to continuous phenotypes [16, 22, 11].

We propose new algorithms for detecting multi-effect and multi-way epistatic interactions, which have three features. First, these algorithms use the root mean square error (RMSE) as the objective function for minimization and guarantee to find either global or local optimal solutions. Second, due to the sparsity of modeling structure, the algorithms are less prone to overfitting than some machine learning approaches. Third, maximally tolerable computation time was explicitly considered in the algorithm design, which allows the user to specify the tradeoff between speed and quality of the solution.

The rest of this paper is organized as follows. In Section 2.3, we cast the multi-effect multi-way epistatic interactions detection problem as an optimization model. Section 2.4 presents three

algorithms for solving the aforementioned problem, with the first one being global optimality oriented and the last two being heuristic. In Section 2.5, we compared the proposed algorithms with state-of-the-art approaches such as a neural network and the Multi-way Interacting Regression via Factorization Machines (MiFM) algorithm [39] in a case study using a yeast dataset. Concluding remarks are made in Section 2.6.

2.3 Problem Definition

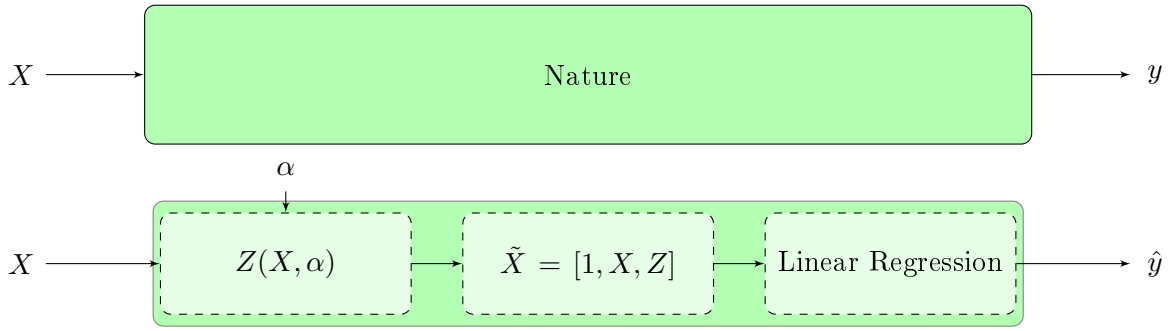

Figure 2.1 Epistases as a descriptive model of nature.

We propose an optimization model that attempts to describe how genotype ($X \in \mathbb{B}^{n\times p}$) manifests its additive effects and epistatic interactions towards the phenotype ($y \in \mathbb{R}^{n\times 1}$), where n is the number of individuals and p the number of genes. As illustrated in Figure 3.1, matrix α is part of the ground truth that defines the epistases, which the model is trying to discovery. The dimension of matrix α is $K \times p$, where K is the number of epistatic effects. Each row in matrix α defines one epistatic effect and each column corresponds to one gene. Elements in α can take three possible values: if $\alpha_{k,j} = 0$, then epistasis k requires that gene j be 0 ($X_{i,j} = 0$) for any individual i to receive this effect; if $\alpha_{k,j} = 1$, then epistasis k requires that gene j be 1 ($X_{i,j} = 1$) for any individual i to receive this effect; if $\alpha_{k,j} = 0.5$, then gene j is not involved in epistasis k. In Figure 3.1, matrix Z is also part of the ground truth, indicating whether or not the individuals receive the epistatic effects, which the model is trying to discovery. The dimension of the binary matrix Z is $n \times K$, with each row corresponding to one individual and each column corresponding to one epistatic effect. If

$Z_{i,k} = 1$, then individual i receives the epistatic effect k, and vice versa. The relationship among X, α, and Z can be described as

$$Z_{i,k} = \prod_{j=1}^{p} I(X_{i,j} + \alpha_{k,j} \neq 1),$$

where, $I(\cdot)$ is the indicator function that is equal to 1 if the statement inside the parentheses is true and 0 otherwise.

The phenotype y is then determined with a multiple linear regression model $y = \tilde{\beta}\tilde{X} + \epsilon$, where $\tilde{X} = [1, X, Z]$ and $\tilde{\beta} = [\beta_0, \beta, b]$ includes coefficients for the intercept (β_0), additive effects (β), and epistatic effects (b). More specially, the epistases model can be formulated as

$$y_i = \beta_0 + \sum_{j=1}^{p} X_{i,j}\beta_j + \sum_{k=1}^{K} b_k Z_{i,k} + \epsilon_i, \quad \forall i \in \{1, ..., n\}. \tag{2.1}$$

For a given training dataset ($X^{\mathrm{T}} \in \mathbb{B}^{n^{\mathrm{T}} \times p}, y^{\mathrm{T}} \in \mathbb{R}^{n^{\mathrm{T}} \times 1}$) and a validation dataset ($X^{\mathrm{V}} \in \mathbb{B}^{n^{\mathrm{V}} \times p}, y^{\mathrm{V}} \in \mathbb{R}^{n^{\mathrm{V}} \times 1}$), the objective of the epistases detection problem is to decipher α, and subsequently Z, β_0, β, and b so that the validation RMSE is minimized. We cast the epistases detection problem as the following optimization model.

$$\min \qquad \zeta = \sqrt{\tfrac{1}{n^{\mathrm{V}}} \sum_{i=1}^{n^{\mathrm{V}}} \left(y_i^{\mathrm{V}} - \hat{y}_i^{\mathrm{V}}\right)^2} \tag{2.2}$$

$$\text{s.t.} \qquad \hat{y}_i^{\mathrm{V}} = \beta_0 + \sum_{j=1}^{p} X_{i,j}^{\mathrm{V}}\beta_j + \sum_{k=1}^{K} b_k Z_{i,k}^{\mathrm{V}} \qquad \forall i \in \{1, ..., n^{\mathrm{V}}\} \tag{2.3}$$

$$\begin{bmatrix} \beta_0 \\ \beta \\ b \end{bmatrix} = \left[\left(\tilde{X}^{\mathrm{T}}\right)^{\top} \tilde{X}^{\mathrm{T}}\right]^{-1} \left(\tilde{X}^{\mathrm{T}}\right)^{\top} y^{\mathrm{T}} \tag{2.4}$$

$$\tilde{X}^{\mathrm{T}} = \left[\mathbf{1}, X^{\mathrm{T}}, Z^{\mathrm{T}}\right] \tag{2.5}$$

$$Z_{i,k}^{\mathrm{V}} = \prod_{j=1}^{p} I(X_{i,j}^{\mathrm{V}} + \alpha_{k,j} \neq 1) \qquad \forall i \in \{1, ..., n^{\mathrm{V}}\}, \forall k \in \{1, ..., K\} \tag{2.6}$$

$$Z_{i,k}^{\mathrm{T}} = \prod_{j=1}^{p} I(X_{i,j}^{\mathrm{T}} + \alpha_{k,j} \neq 1) \qquad \forall i \in \{1, ..., n^{\mathrm{T}}\}, \forall k \in \{1, ..., K\}. \tag{2.7}$$

In the rest of the paper, we will use $\zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, \alpha, X^{\mathrm{V}}, y^{\mathrm{V}})$ to denote the RMSE of a given α, which may or may not be an optimal solution to (3.3)-(2.7).

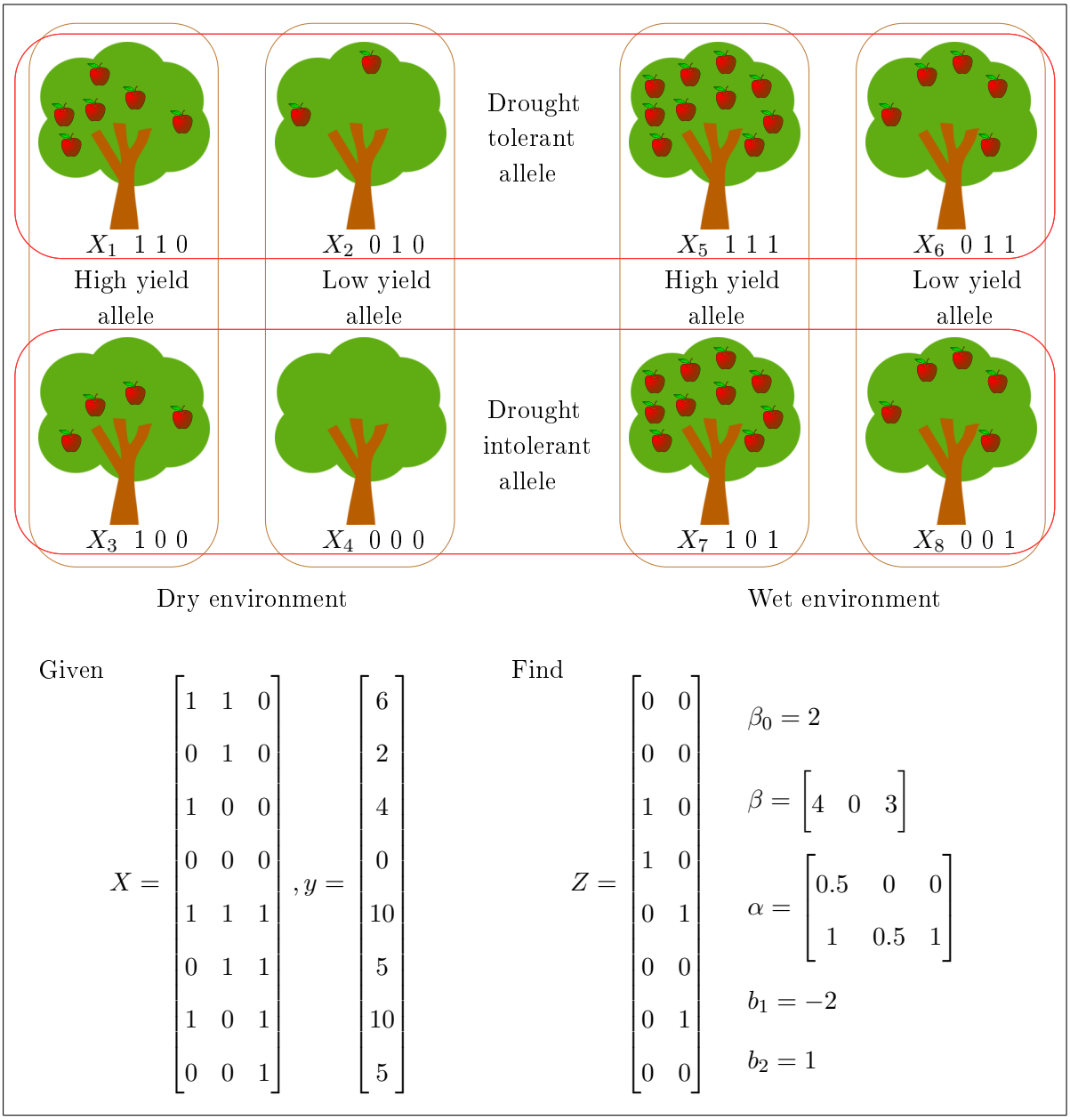

Figure 2.2 An illustrative example of epistatic interactions.

To illustrate the problem definition, consider the example in Figure 2.2, in which eight individuals ($n = 8$) of apple trees bear different numbers of fruits as a result of three genes ($p = 3$): high/low yield, drought tolerance/intolerance, and wet/dry environment (treated as a special gene). The ground truth, which needs to be deciphered from the observable data X and y, includes the following.

Each tree has two apples as a base case ($\beta_0 = 2$). A high yield allele gives an extra 4 apples ($\beta_1 = 4$), a drought tolerant allele does not have a direct effect on the yield ($\beta_2 = 0$), and a wet environment gives an extra 3 apples ($\beta_3 = 3$). On top of these additive effects, there are two epistatic effects, each affecting two individuals: if a tree has drought intolerant allele ($\alpha_{1,2} = 0$) and grows in the dry environment ($\alpha_{1,3} = 0$), then it loses two apples ($b_1 = -2$); if a tree has high yield allele ($\alpha_{2,1} = 1$) and grows in the wet environment ($\alpha_{2,3} = 1$), then it gains one extra apple ($b_2 = 1$). Apple tree i receives effect k ($Z_{i,k} = 1$) if and only if $X_{i,j} + \alpha_{k,j} \neq 1$, or equivalently $X_{i,j} = \alpha_{k,j}$, for each gene j.

2.4 Methods

In this section, we propose three approaches to detecting multi-effect and multi-way epistases, including a mixed integer quadratic programming (MIQP) based model that can be solved by existing algorithms and solvers to global optimality and two local optimality oriented heuristics.

2.4.1 MIQP Model

When the objective is to minimize the RMSE for the training set, we formulate the epistases detection problem as the following MIQP, which is a special case of model (3.3)-(2.7) when the validation set is the same as the training set. This model is an extension of the mixed integer linear programming model presented in Wang and Mehr [34] by directly minimizing the mean square error and detecting multiple epistatic effects.

$$\min \quad \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \tag{2.8}$$

$$\text{s.t.} \quad \hat{y}_i=\beta_0+\sum_{j=1}^{p}X_{i,j}\beta_j+\sum_{k=1}^{K}w_{i,k} \qquad \forall i\in\{1,...,n\} \tag{2.9}$$

$$\underline{b}_k Z_{i,k}\le w_{i,k}\le \bar{b}_k Z_{i,k} \qquad \forall i\in\{1,...,n\}, k\in\{1,...,K\} \tag{2.10}$$

$$w_{i,k}\le b_k-\underline{b}_k(1-Z_{i,k}) \qquad \forall i\in\{1,...,n\}, k\in\{1,...,K\} \tag{2.11}$$

$$w_{i,k}\ge b_k-\bar{b}_k(1-Z_{i,k}) \qquad \forall i\in\{1,...,n\}, k\in\{1,...,K\} \tag{2.12}$$

$$\lambda_{k,j}+\mu_{k,j}\le 1 \qquad \forall j\in\{1,...,p\}, k\in\{1,...,K\} \tag{2.13}$$

$$\sum_{j=1}^{p}X_{i,j}(\lambda_{k,j}-\mu_{k,j})\ge -p(1-Z_{i,k})+\sum_{j=1}^{p}\lambda_{k,j} \qquad \forall i\in\{1,...,n\}, k\in\{1,...,K\} \tag{2.14}$$

$$\sum_{j=1}^{p}X_{i,j}(\lambda_{k,j}-\mu_{k,j})\le \sum_{j=1}^{p}\lambda_{k,j}-1+pZ_{i,k} \qquad \forall i\in\{1,...,n\}, k\in\{1,...,K\} \tag{2.15}$$

$$\sum_{j=1}^{p}(\lambda_{k,j}+\mu_{k,j})\le C_k \qquad \forall k\in\{1,...,K\} \tag{2.16}$$

$$\lambda_{k,j},\mu_{k,j},Z_{i,k}\in\{0,1\};\beta_0,\beta_j,b_k \text{ free} \qquad \forall i\in\{1,...,n\}, j\in\{1,...,p\}, k\in\{1,...,K\}. \tag{2.17}$$

The objective function (2.8) minimizes the mean square error, which is the square of RMSE, which is equivalent to minimizing RMSE and is solvable by many quadratic program solvers. In order to linearize the non-convex quadratic term $b\cdot Z$ in (3.4), we replaced it with an auxiliary variable w. Constraints (2.10)-(2.12) are a commonly used modeling technique to linearize the product of a continous variable and a binary variable, where $\underline{b}$ and $\bar{b}$ are assumed to be the lower and upper bounds of b. Two new variables λ and μ are introduced to represent α, where $\lambda_{k,j}=I(\alpha_{k,j}=1)$ and $\mu_{k,j}=I(\alpha_{k,j}=0)$ for all $j\in\{1,...,p\}, k\in\{1,...,K\}$. Matrices λ and μ are an alternative definition of the epistases. The dimensions of λ and μ are both $K\times p$, with each row corresponding to one epistatic effect and each column corresponding to one gene. For each j and k, if $\alpha_{k,j}=0.5$, then $\lambda_{k,j}=0$ and $\mu_{k,j}=0$; if $\alpha_{k,j}=1$, then $\lambda_{k,j}=1$ and $\mu_{k,j}=0$; if $\alpha_{k,j}=0$, then $\lambda_{k,j}=0$ and $\mu_{k,j}=1$. Constraint (2.13) imposes the obvious logical requirement for λ and μ. Constraint (2.14) enforces that $Z_{i,k}=1$ when individual i receives epistasis k and constraint (2.15) does similar for

the other case. Constraint (2.16) sets an upper bound to the complexity of each epistasis, which can be determined based on the statistical power of the sample size n. Supports and types of all decision variables are given in constraint (2.17).

2.4.2 Local search algorithm

Due to the combinatorial and nonlinear nature of model (2.8)-(2.17), solving it to global optimality using existing algorithms and solvers is expected to be time consuming. In this section, we present a more computationally tractable algorithm for solving the MIQP model (2.8)-(2.17) at the price of having a local optimal (and not necessarily global optimal) solution. This algorithm is an extension of the local search algorithm in Wang and Mehr [34], which was designed for detecting only one epistatic effect. The algorithm is defined as follows and illustrated in Figure 2.3.

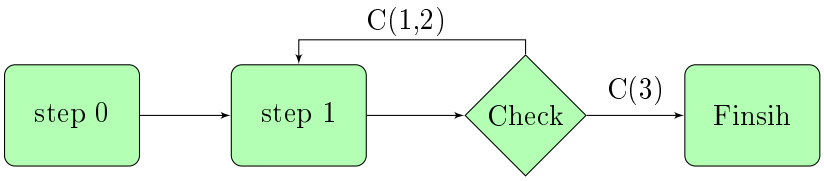

Figure 2.3 The local search algorithm diagram

We provide additional remarks of the algorithm as follows.

- The hyperparameter C should be appropriately determined according to the dimension of the training dataset. On average, to observe an individual with an epistasis of complexity C, it requires 2^C samples of training data. Similarly, hyperparameter D should reflect the computation time requirement. Any increase in D will greatly expand the space of the neighborhood and thus the computation time. When $D = p$, then the algorithm reduces to global search through enumeration.

- In Step 1 and the check point, the search space $\mathcal{A}(k, d)$ is defined for the kth epistasis within the d-hop neighborhood, which is adaptively expanded or shrunk depending on the search outcome. The rule is to always search the smallest neighborhood for a potentially better

solution to improve any of the K epistases, and to expand to a broader neighborhood only if a better solution was confirmed to not exist in the previous neighborhood.

- We can combine the MIQP approach and the local search algorithm by obtaining an integer feasible (and not necessarily optimal) solution to model (2.8)-(2.17) and then use it as the initial incumbent solution in Step 1. The rationale for this combination is that existing branch and bound algorithms may be able to find a close-to-optimal solution to (2.8)-(2.17) relatively quickly but take a long time to identify and confirm the global optimal solution. If we use the local search algorithm to refine a feasible solution, then its local optimum may turn out to be the global optimal one.

Algorithm 1 Local search algorithm

Input: Training data $(X \in \mathbb{B}^{n\times p}, y \in \mathbb{R}^{n\times 1})$.

Output: A local optimal solution $\alpha^* \in \{0, 0.5, 1\}^{K\times p}$ to model (2.8)-(2.17).

Hyperparameters: Complexity parameter C and maximal depth D.

Step 0: Initialize the incumbent solution $\alpha^* = 0.5^{K\times p}$. Set $k = 1, d = 1$, and matrix $O = 0^{K\times D}$.

Step 1: Identify the d-hop neighborhood of $\alpha^*_{k,:}$ as follows, where $\alpha^*_{k,:}$ denotes the kth row of matrix α^* and $\|\cdot\|_0$ is the L_0 norm that is equal to the total number of nonzero elements.

$$\mathcal{A}(k,d) \quad = \{ \quad \alpha \in \{0, 0.5, 1\}^{K\times p} : \alpha_{i,:} = \alpha^*_{i,:}, \forall i \neq k; \|\alpha_{k,:} - \alpha^*_{k,:}\|_0 \leq d; \sum_{j=1}^{p} 2|\alpha_{k,j} - 0.5| \leq C\}.$$

Evaluate $\zeta(X, y, \alpha, X, y)$ for all $\alpha \in \mathcal{A}(k, d)$ and let $\tilde{\alpha}$ denote the best solution $\tilde{\alpha} = \underset{\alpha\in\mathcal{A}(k,d)}{\arg\min}\, \zeta(X, y, \alpha, X, y)$.

Check point

if $\zeta(X, y, \tilde{\alpha}, X, y) < \zeta(X, y, \alpha^*, X, y)$ **then**

  C(1): Update $\alpha^* \leftarrow \tilde{\alpha}$, reset $d = 1$ and $O = 0^{K\times D}$, and go to Step 1.

else if $\sum_d \sum_k O_{k,d} \leq K \cdot D - 2$ **then**

  C(2): Set $O(k, d) = 1$, reset (k, d) as $\arg\min_{k,d}\{d : O(k, d) = 0\}$, and go to Step 1.

else

  C(3): Finish.

end if

2.4.3 Heuristic Algorithm

We present another algorithm for solving the epistases detection problem (3.3)-(2.7), which complements the two approaches in Sections 2.4.1 and 2.4.2 with several features. First, it minimizes the validation RMSE rather than the training RMSE, making it less prone to overfitting. Second, it attempts to explore multiple local optima and avoid brute force enumeration within local neighborhoods. Third, it adjusts the search strategy in response to the maximally tolerated computation

times, allowing the user to make a tradeoff between speed and solution quality. The algorithm is defined in Algorithm 2 and diagrammed in Figure 2.4.

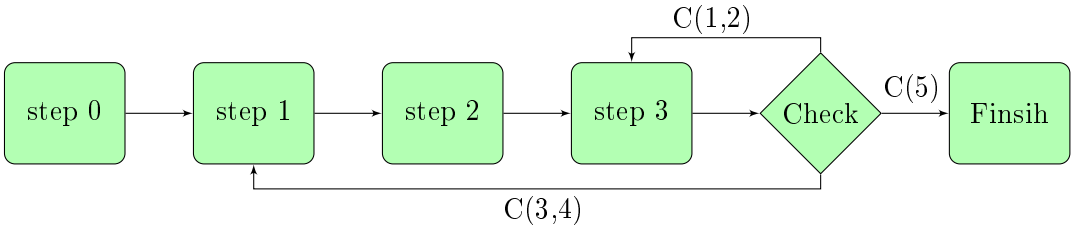

Figure 2.4 The heuristic algorithm diagram

We provide additional remarks of the algorithm as follows.

- One of the hyperparameters is the selection intensity θ_t for each iteration t, which is the number of "high quality" genes that are considered to be more likely to be involved in epistases. When the computation time limit is low, smaller values of θ should be used to improve the chance of finding a local optimal solution at the price of potentially eliminating important genes from the search space.

- In Step 0, set $\mathcal{A}_2$ is created as a set of random epistases with a complexity of 2. Then matrix A represents a random subset of these epistases, which will be used in Step 1 to select high quality genes.

- In Step 1, the subset of high quality genes, $\mathcal{P}$, was selected based on the frequency of genes that are involved in epistases in A that result in the lowest training RMSE values.

- An alternative approach to the third bullet in Step 1 is to replace the RMSE calculation with a much faster calculation of correlations between Z^{T} and regression residual $|\hat{y}^{\mathrm{T}} - y^{\mathrm{T}}|$. Here, Z^{T} can be calculated using matrix A and Constraint (2.7), and $\hat{y}^{\mathrm{T}}$ using Constraint (3.4) with training data rather than validation data. Then set $\mathcal{P}$ can be determined as the top θ_t genes whose corresponding Z^{T} columns have the highest correlations with the regression residual.

Algorithm 2 Heuristic algorithm

Input: Training data $(X^{\mathrm{T}} \in \mathbb{B}^{n^{\mathrm{T}} \times p}, y^{\mathrm{T}} \in \mathbb{R}^{n^{\mathrm{T}} \times 1})$ and validation data $(X^{\mathrm{V}} \in \mathbb{B}^{n^{\mathrm{V}} \times p}, y^{\mathrm{V}} \in \mathbb{R}^{n^{\mathrm{V}} \times 1})$.

Output: A local optimal solution $\alpha^* \in \{0, 0.5, 1\}^{K \times p}$ to model (3.3)-(2.7).

Hyperparameters: Complexity C, maximal depth D, maximally tolerated computation time $T^{\max}$, sample size S, and selection intensity $\theta_t, \forall t \in \{1, 2, ...\}$.

Step 0: Initialize the incumbent solution $\alpha^* = 0.5^{1 \times p}$, set $t = 0$ and $K = 1$, and define $\mathcal{A}_2$ as $\mathcal{A}_2 = \left\{\alpha \in \{0, 0.5, 1\}^{1 \times p} : \sum_{j=1}^{p} |\alpha_j - 0.5| = 1\right\}$. Create an arbitrary matrix $A \in \{0, 0.5, 1\}^{S \times p}$ with each row being from the set $\mathcal{A}_2$: $A_{i,:} \in \mathcal{A}_2, \forall i \in \{1, ..., S\}$.

Step 1: Update $t \leftarrow t+1$, identify set $\mathcal{P}$ that satisfies all of the requirements $\mathcal{P} \subseteq \{1, ..., p\}, |\mathcal{P}| = \theta_t, \zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, A_{i,:}, X^{\mathrm{T}}, y^{\mathrm{T}}) \leq \zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, A_{j,:}, X^{\mathrm{T}}, y^{\mathrm{T}})$ for any i and j such that $\max_{l \in \mathcal{P}} |A_{i,l} - 0.5| = 0.5$ and $\max_{l \in \mathcal{P}} |A_{j,l} - 0.5| = 0$. Then go to Step 2.

Step 2: Initialize the current solution as $\hat{\alpha}_{i,j} = \begin{cases} \alpha^*_{i,j}, & \text{if } i \leq K-1 \\ 0.5, & \text{otherwise.} \end{cases}, \forall i \in \{1, ..., K\}, j \in \{1, ..., p\}$. Set $d = 1$, and go to step 3.

Step 3: Identify the d-hop neighborhood of $\hat{\alpha}$ as follows.

$$\begin{aligned} \mathcal{A}^1(K, d) \quad = \{ \quad & \alpha \in \{0, 0.5, 1\}^{K \times p} : \alpha_{i,:} = \hat{\alpha}_{i,:}, \forall i \in \{1, ..., K-1\}; \|\alpha_{K,:} - \hat{\alpha}_{K,:}\|_0 \leq d \\ & \sum_{j \in \mathcal{P}} 2|\alpha_{K,j} - 0.5| \leq C; \alpha_{K,j} = 0.5, \forall j \in \{1, ..., p\} \backslash \mathcal{P}\} \end{aligned}$$

Evaluate $\zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, \alpha, X^{\mathrm{T}}, y^{\mathrm{T}})$ for all $\alpha \in \mathcal{A}^1(K, d)$ and let α^d denote the best solution: $\alpha^d = \underset{\alpha \in \mathcal{A}^1(K,d)}{\arg\min}\ \zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, \alpha, X^{\mathrm{T}}, y^{\mathrm{T}})$.

Check point

if $\zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, \alpha^d, X^{\mathrm{T}}, y^{\mathrm{T}}) < \zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, \hat{\alpha}, X^{\mathrm{T}}, y^{\mathrm{T}})$ **then**

  C(1): Update $\hat{\alpha} \leftarrow \alpha^d$, reset $d = 1$, and go to Step 3.

else if $d \neq D$ **then**

  C(2): Set $d \leftarrow d + 1$ and go to Step 3.

else

  if $\zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, \hat{\alpha}, X^{\mathrm{V}}, y^{\mathrm{V}}) < \zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, \alpha^*, X^{\mathrm{V}}, y^{\mathrm{V}})$ **then**

    C(3): Update $\alpha^* \leftarrow \hat{\alpha}$, reset $d = 1$, $K \leftarrow K + 1$, and go to Step 1.

  else if $T^{\max}$ has not been exceeded **then**

    C(4): Update $K \leftarrow K - 1$, and find the least effective epistatic effect $\tilde{k} = \underset{k \in \{1, ..., K\}}{\arg\max} \{\zeta(X^{\mathrm{T}}, y^{\mathrm{T}}, \alpha^*_{k,:}, X^{\mathrm{T}}, y^{\mathrm{T}})\}$. Update α^* as $\alpha^* \leftarrow \alpha^* \backslash \alpha^*_{\tilde{k},:}$, reset $d = 1$, and go to Step 1.

  else

    C(5): Finish.

  end if

end if

- Step 2, Step 3, and check point with C(1,2) in the heuristic algorithm are similar with Step 0, Step 1, and check point with C(1,2) in the local search algorithm, but only one epistatic effect involving high quality genes within set $\mathcal{P}$ are under consideration.
- Condition C(3) will trigger the algorithm to search for another epistatic effect. Under condition C(4), when the recently added epistasis failed to improve the RMSE for the validation data, the algorithm will identify the least useful epistasis and then try to replace it a new one.
- Although time limit is only checked in C(4) of the heuristic algorithm, it can also be checked in other places of the algorithm to enforce termination by the deadline.

2.5 Case Study

We conducted a case study to compare the performances of the proposed algorithms and those from the literature, which are summarized as follows.

- The MIQP model, from Section 2.4.1, solved using CPLEX 12.
- The local search algorithm, from Section 2.4.2, a straightforward extension of the algorithm from Wang and Mehr [34].
- The heuristic algorithm, from Section 2.4.3.
- The linear regression algorithm, which represents a special case of Equation (4.1) with $b_k = 0, \forall k$ and serves as a comparison benchmark.
- A neural network model, which is able to approximate almost any nonlinear input-output relationship and commonly used to approximate complex relationships between response and explanatory variables.
- The MiFM from Yurochkin et al. [39], which is a Bayesian regression method using a factorization mechanism for representing the regression coefficients of interactions.

2.5.1 Data

We used the yeast data set from Bloom et al. [1], which contained n =4,390 individuals and p =28,220 genes, represented by single nucleotide polymorphism (SNPs). In order to have access to the ground truth of epistases, we simulated phenotype data using the following parameters.

- β_0: Uniform distribution $\mathcal{U}(0, 30)$
- β: Uniform distribution $\mathcal{U}(0, 30)$
- b: Uniform distribution $\mathcal{U}(15, 30)$
- ϵ: Normal distribution $\mathcal{N}(0, \sigma^2)$

2.5.2 Design of experiments

We designed computational experiments to test the sensitivity of the six algorithms listed at the beginning of Section 2.5 with respect to multiple parameters, which are summarized in Table 2.1. A full factorial of 5,280 combinations of these parameter values were tested. For each combination, p genes were randomly selected from the 28,220 in the original dataset, additive effects of these genes were randomly created using the parameter settings in Section 2.5.1 with the specified σ, K epistatic effects were randomly created, each with a random complexity up to C, and $T^{\max}$ was used as the computation deadline.

Table 2.1 Parameters for the sensitivity analysis

Parameter	**Description**	**Values**
p	Number of genes	$(50, 100, 150, 200)$
K	Number of epistatic effects	$(1, 2, 3, 4, 5)$
C	Maximal epistisis complexity	$(2, 3, 4, 5)$
$T^{\max}$	Computation deadline	$(10, 110, \ldots, 1010)$
σ	Standard deviation of random error	$(0, 1, 2, 3, 4, 5)$

The Matlab Neural Network Toolbox was used to create neural networks with 2 or 3 hidden layers and each containing between 10 and 20 neurons. Different structures were tested for each combination of parameters and the best performance was recorded for comparison. The Python implementation of the MiFM algorithm by Yurochkin et al. [39] was used with some modification in our case study. The entire data set was divided into training (56%), validation (24%), and test (20%) sets. Improving incumbent solutions from all algorithms were recorded iteratively. The experiment was executed on several computers with identical configurations, each having a 3.70 GHz CPU and 16 GB memory. The total computation time for these experiments was approximately 156 CPU days. Results of the experiment are shown in Figures 2.5-2.9.

2.5.3 Results

Figures 2.5-2.9 compare the computational performance of the six algorithms with respect to RMSEs on training and test data sets. We make the following observations.

- Figure 2.5 reveals the performance of the algorithms under different computational deadlines. The heuristic algorithm took two to three minutes to achieve its best performance, which was fairly close to the ground truth. The neural network achieved its best performance even faster, but it suffered from overfitting, lower prediction accuracy, and high variability in prediction accuracy. The linear regression and MIQP algorithms demonstrated the same performance in this figure (and others) within a computational deadline of 1,010 seconds, but their long-term performances could be dramatically different. Linear regression was more computationally tractable than all other algorithms under comparison, and it showed no sensitivity with respect to the computational deadline, but its prediction accuracy was the lowest. On the other hand, MIQP would eventually find the global optimal solution, but it could take extremely long time. For the first 1,010 seconds, the performance of MIQP was the same as linear regression, since MIQP used the linear relaxation solution (same as linear regression) as the initial incumbent solution, which could take longer than 1,010 seconds to be updated. Local search and MiFM algorithms demonstrated similar performances, which slowly improved over time and might

keep improving after 1,010 seconds. Their prediction accuracies were comparable with neural network but with less variability.

- Figure 2.6 indicates the performance of the algorithms with respect to the number of genes, p. The heuristic algorithm had low prediction errors and was insensitive to p. This was due to Step 1 of the algorithm, in which high quality genes were identified and used to detect epistases. In contrast, the local search and MiFM algorithms considered all genes, which became increasingly time consuming for larger p. As such, their prediction errors were relatively small for lower p but deteriorated quickly as p increased. Neural network worked almost as well as the heuristic algorithm for lower p but was also prone to overfitting and sensitive to p. It also had higher variability in prediction accuracy than other algorithms.

- Figure 2.7 compares the sensitive of different algorithms to the number of epistatic effects, K. The heuristic algorithm had low prediction errors and was insensitive to K, which was because it was designed to take advantage of the special structure being assumed in model (3.3)-(2.7). All other algorithms demonstrated large sensitivity to K, since larger K makes the input-output relationship more complex and harder to predict. In particular, neural network had a much more variable prediction error than other algorithms, and its overfitting issue was also more visible than others.

- Figure 2.8 demonstrates how all algorithms were insensitive to the maximal complexity, C, of the epistases. The heuristic algorithm had much lower prediction errors than others. The performances of linear regression / MIQP, local search, and MiFM algorithms improved for more complex epistases. A similar phenomenon was also observed in [34], and the interpretation was that more complex epistases would be rarer, affecting fewer individuals in the population, allowing less sophisticated models to better represent the entire population. On the other hand, neural network had a complex modeling structure and did not demonstrate improved prediction accuracy for higher C.

- Figure 2.9 shows the greatest contrast between the heuristic algorithm and others with respect to the standard deviation of the random error, σ. Since the performance of the heuristic algorithm is close to the ground truth, most of the non-random variability in the phenotype was explained, and prediction errors were almost completely caused by σ. In contrast, all other algorithms demonstrated much larger variability in prediction errors, with neural network being even more so than others.

As another way to demonstrate the effectiveness of the heuristic algorithm, Table 2.2 shows the numbers of true epistatic effects (K) versus numbers of correctly deciphered ones. Overall, the heuristic algorithm was able to correctly (correct number of effects with exact combinations of genes for each epistasis) decipher all epistatic effects in 4,003 out of the 5,280 instances simulated ground truth, which was a 75.81% success rate. If success was more loosely defined as achieving an RMSE within 0.01 of the ground truth, then the heuristic algorithm had a 88.20% success rate.

Table 2.2 The number of true epistatic effects (K) versus the number of correctly deciphered ones (K') .

	$K'=0$	$K'=1$	$K'=2$	$K'=3$	$K'=4$	$K'=5$
$K=1$	70	986	0	0	0	0
$K=2$	17	134	905	0	0	0
$K=3$	8	50	202	796	0	0
$K=4$	8	32	83	240	693	0
$K=5$	12	24	45	122	230	623

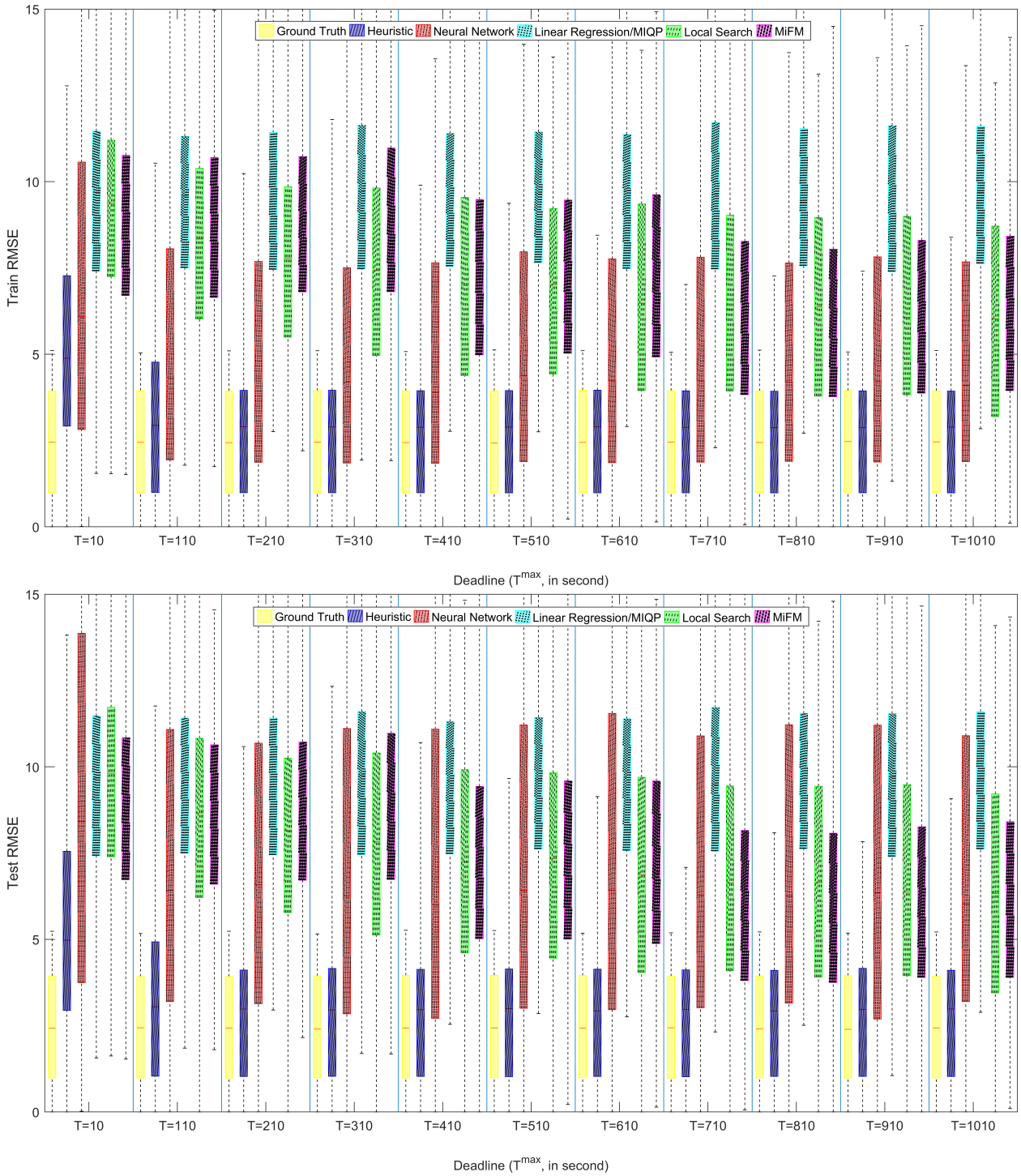

Figure 2.5 Comparing algorithm in terms of computation deadline ($T^{\max}$)

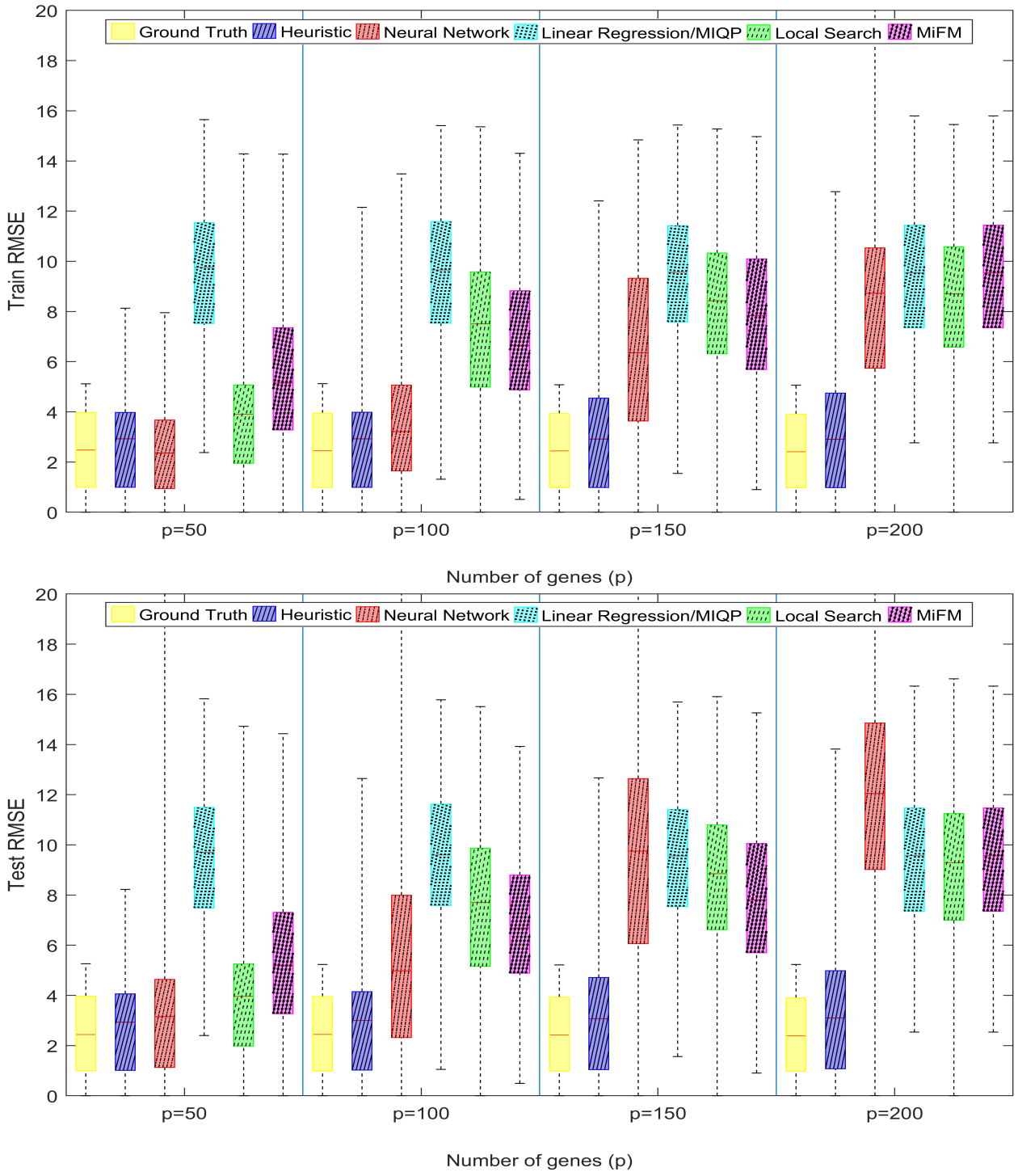

Figure 2.6 Comparing algorithm in terms of the number of genes (p)

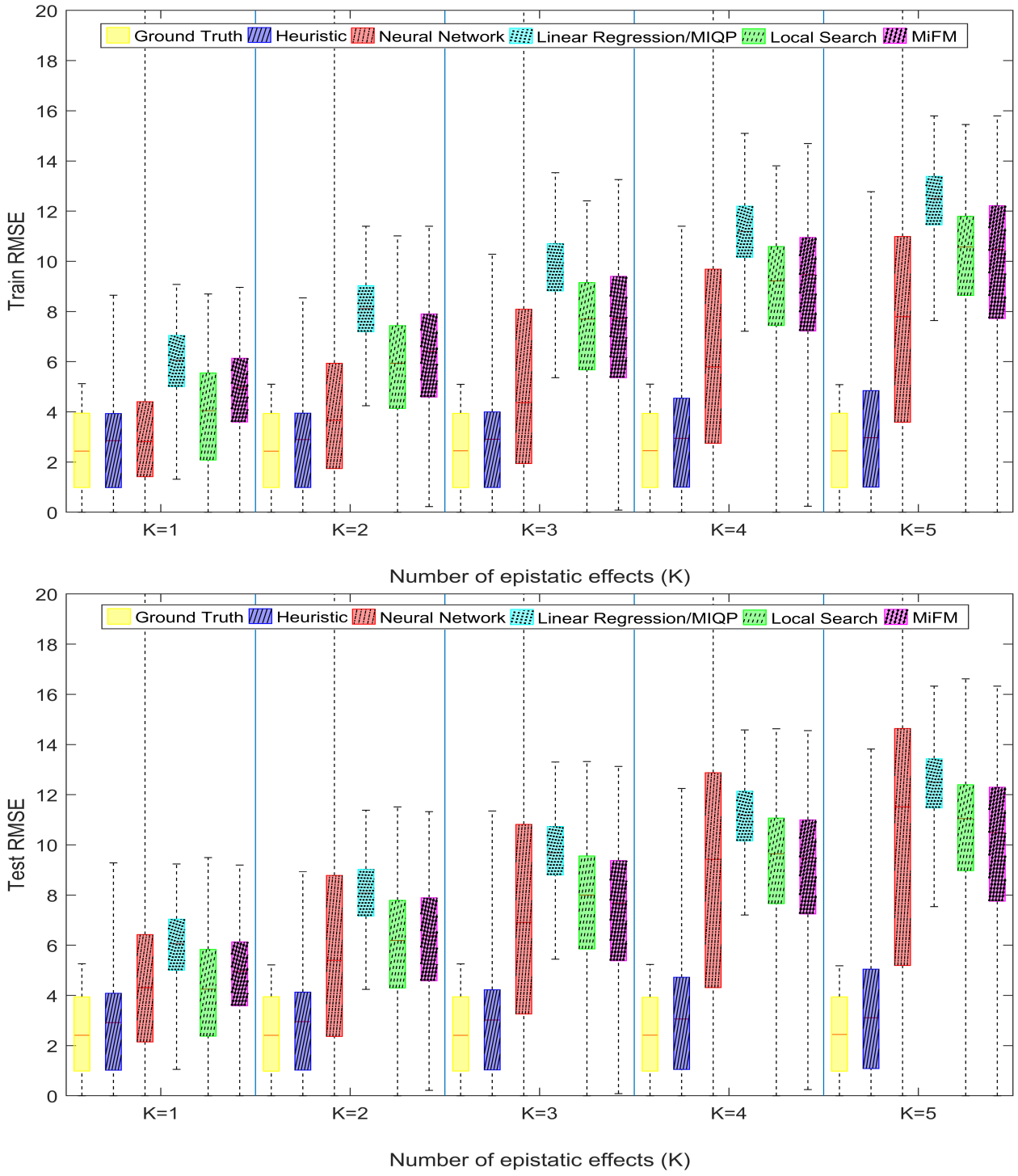

Figure 2.7 Comparing algorithm in terms of the number of epistatic effects (K)

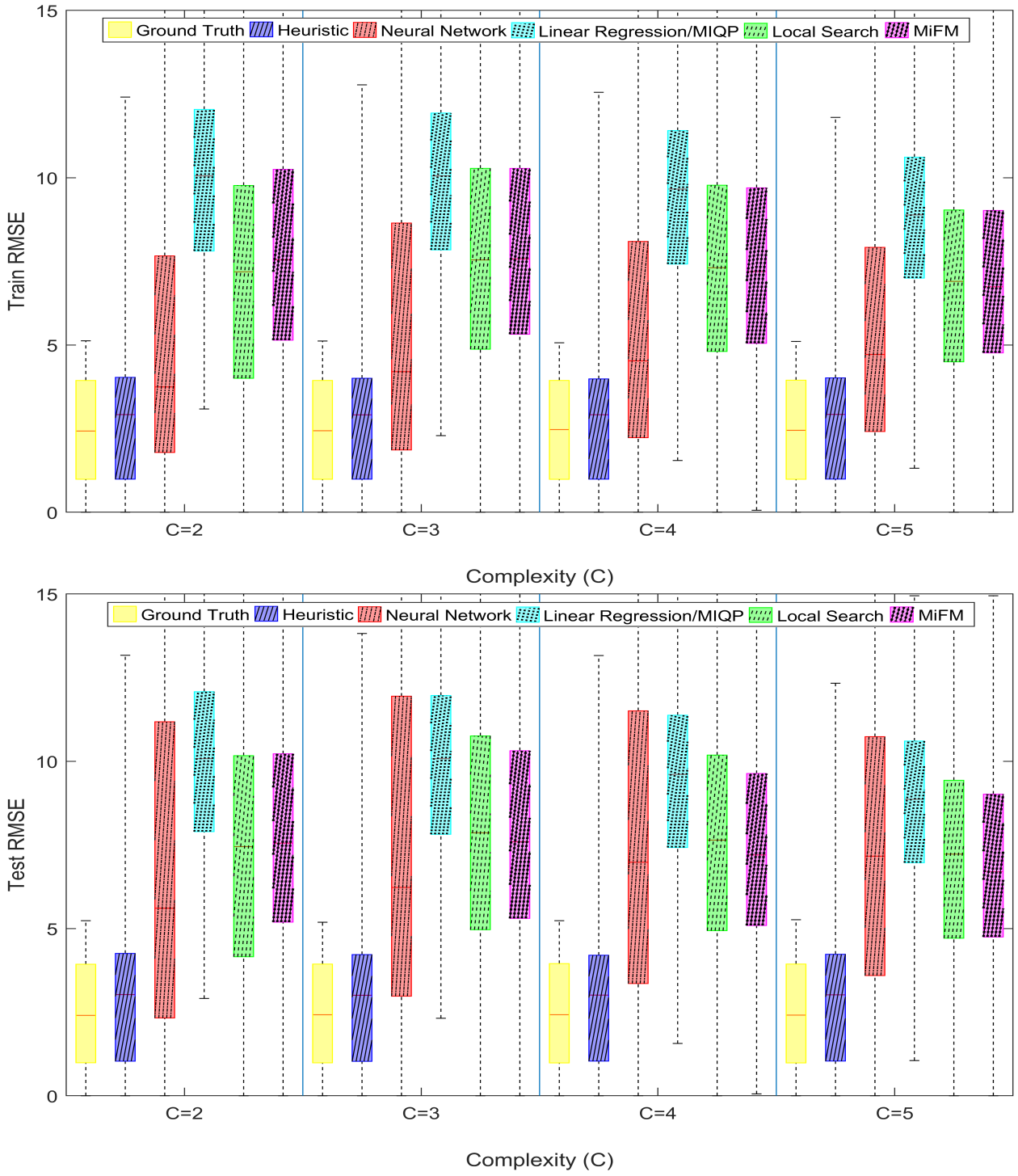

Figure 2.8 Comparing algorithm in terms of maximal complexity (C)

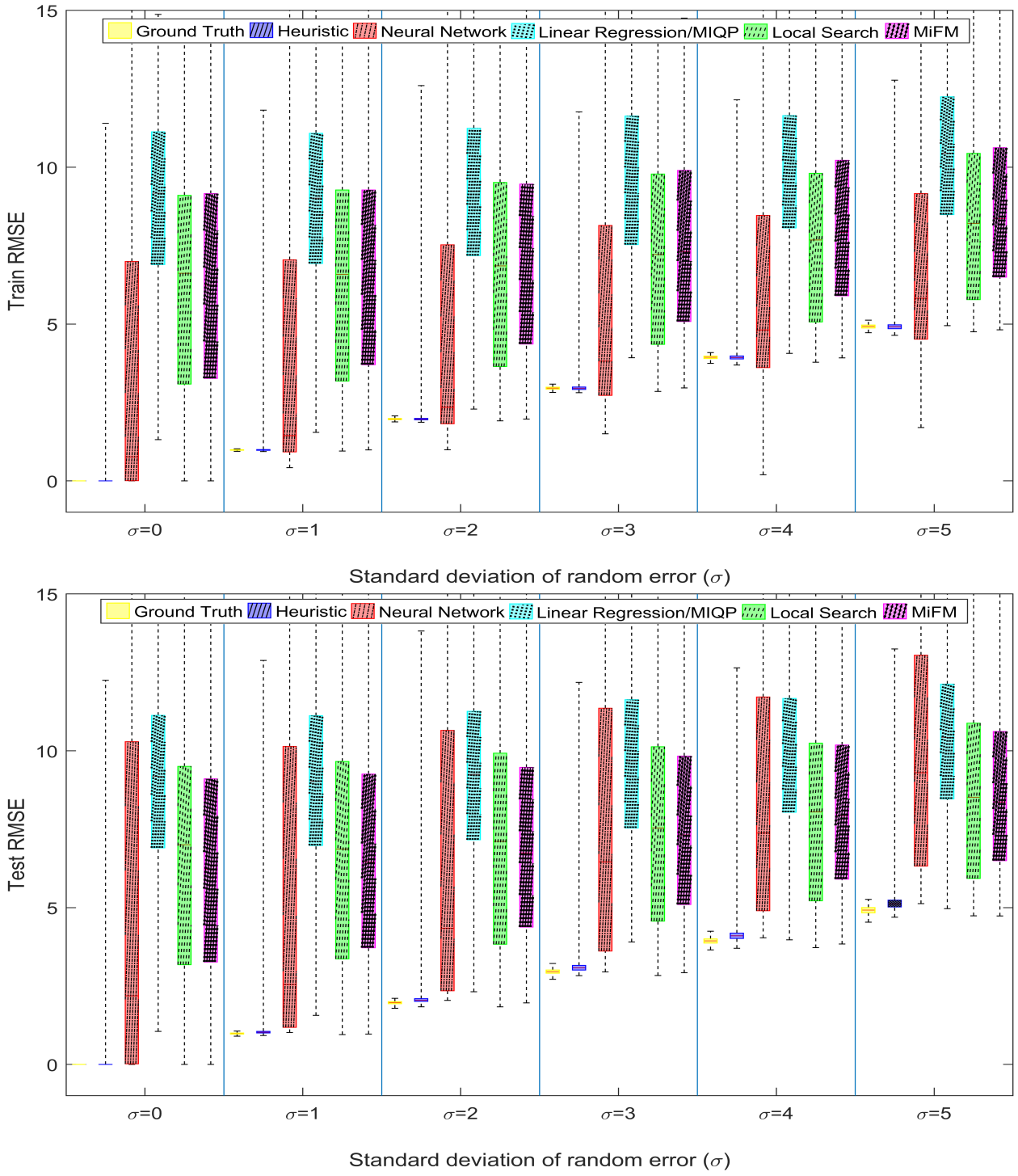

Figure 2.9 Comparing algorithm in terms of standard deviation of random error (σ)

2.6 Conclusion

The contribution of this paper was to propose three approaches for detecting multi-effect and multi-way epistatic interactions. The first approach was to cast the problem as an MIQP model, which can be solved to global optimality using existing algorithms and solvers. The second approach was a local search algorithm that can efficiently find local optimal solutions to the MIQP. The third algorithm was a heuristic algorithm that applies additional search strategies depending on the maximally tolerated computation time.

Effectiveness of the proposed approaches, especially the heuristic algorithm, was tested in a case study using realistic data sets. Computational results suggested that the heuristic algorithm was able to find optimal or close to optimal solutions very quickly, outperforming neural network models, linear regression/MIQP, local search, and MiFM algorithms in terms of both speed and solution quality. Due to its robustness against multiple parameters and lack of overfitting, this algorithm is expected to be able to provide biologically verifiable interpretations of epistases for scientific discoveries.

Future research should focus on two directions. One is to further improve the efficiency of the algorithm so that it can be applied to more genetic data sets with larger numbers of genetic markers. The other is to extend to other cases, such as case-control studies or non-binary explanatory variables.

2.7 References

[1] Bloom, J. S., Kotenko, I., Sadhu, M. J., Treusch, S., Albert, F. W., and Kruglyak, L. (2015). Genetic interactions contribute less than additive effects to quantitative trait variation in yeast. *Nature Communications*, 6:8712.

[2] Chen, S.-H., Sun, J., Dimitrov, L., Turner, A. R., Adams, T. S., Meyers, D. A., Chang, B.-L., Zheng, S. L., Grönberg, H., Xu, J., et al. (2008). A support vector machine approach for detect-

ing gene-gene interaction. *Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society*, 32(2):152–167.

[3] Combarros, O., Cortina-Borja, M., Smith, A. D., and Lehmann, D. J. (2009). Epistasis in sporadic alzheimer's disease. *Neurobiology of Aging*, 30(9):1333–1349.

[4] Evans, D. M., Marchini, J., Morris, A. P., and Cardon, L. R. (2006). Two-stage two-locus models in genome-wide association. *PLoS Genetics*, 2(9):e157.

[5] Fang, Y.-H. and Chiu, Y.-F. (2012). Svm-based generalized multifactor dimensionality reduction approaches for detecting gene-gene interactions in family studies. *Genetic Eepidemiology*, 36(2):88–98.

[6] González-Domínguez, J., Schmidt, B., Kässens, J. C., and Wienbrandt, L. (2014). Hybrid cpu/gpu acceleration of detection of 2-snp epistatic interactions in gwas. In *European Conference on Parallel Processing*, pages 680–691. Springer.

[7] Goudey, B., Rawlinson, D., Wang, Q., Shi, F., Ferra, H., Campbell, R. M., Stern, L., Inouye, M. T., Ong, C. S., and Kowalczyk, A. (2013). Gwis-model-free, fast and exhaustive search for epistatic interactions in case-control gwas. *BMC Genomics*, 14(3):S10.

[8] Guan, B., Zhao, Y., and Sun, W. (2018). Ant colony optimization with an automatic adjustment mechanism for detecting epistatic interactions. *Computational Biology and Chemistry*, 77:354–362.

[9] Gusareva, E. S., Carrasquillo, M. M., Bellenguez, C., Cuyvers, E., Colon, S., Graff-Radford, N. R., Petersen, R. C., Dickson, D. W., John, J. M. M., Bessonov, K., et al. (2014). Genome-wide association interaction analysis for alzheimer's disease. *Neurobiology of Aging*, 35(11):2436–2443.

[10] Han, B. and Chen, X.-w. (2011). bneat: a bayesian network method for detecting epistatic interactions in genome-wide association studies. In *BMC Genomics*, volume 12, page S9. BioMed Central.

[11] Hardison, N. E. and Motsinger-Reif, A. A. (2011). The power of quantitative grammatical evolution neural networks to detect gene-gene interactions. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pages 299–306. ACM.

[12] Jiang, R., Tang, W., Wu, X., and Fu, W. (2009). A random forest approach to the detection of epistatic interactions in case-control studies. *BMC Bioinformatics*, 10(1):S65.

[13] Koo, C. L., Liew, M. J., Mohamad, M. S., Salleh, M., and Hakim, A. (2013). A review for detecting gene-gene interactions using machine learning methods in genetic epidemiology. *BioMed Research International*, 2013.

[14] Leem, S., Jeong, H.-h., Lee, J., Wee, K., and Sohn, K.-A. (2014). Fast detection of high-order epistatic interactions in genome-wide association studies using information theoretic measure. *Computational Biology and Chemistry*, 50:19–28.

[15] Lin, H.-Y., Ann Chen, Y., Tsai, Y.-Y., Qu, X., Tseng, T.-S., and Park, J. Y. (2012). Trm: A powerful two-stage machine learning approach for identifying snp-snp interactions. *Annals of Human Genetics*, 76(1):53–62.

[16] Motsinger, A. A., Dudek, S. M., Hahn, L. W., and Ritchie, M. D. (2006). Comparison of neural network optimization approaches for studies of human genetics. In *Workshops on Applications of Evolutionary Computation*, pages 103–114. Springer.

[17] Özgür, A., Vu, T., Erkan, G., and Radev, D. R. (2008). Identifying gene-disease associations using centrality on a literature mined gene-interaction network. *Bioinformatics*, 24(13):i277–i285.

[18] Padyukov, L. (2013). *Between the lines of genetic code: genetic interactions in understanding disease and complex phenotypes*. Academic Press.

[19] Piriyapongsa, J., Ngamphiw, C., Intarapanich, A., Kulawonganunchai, S., Assawamakin, A., Bootchai, C., Shaw, P. J., and Tongsima, S. (2012). iloci: a snp interaction prioritization technique for detecting epistasis in genome-wide association studies. In *BMC Genomics*, volume 13, page S2. BioMed Central.

[20] Rekaya, R. and Robbins, K. (2009). Ant colony algorithm for analysis of gene interaction in high-dimensional association data. *Revista Brasileira de Zootecnia*, 38(SPE):93–97.

[21] Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., and Moore, J. H. (2001). Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *The American Journal of Human Genetics*, 69(1):138–147.

[22] Ritchie, M. D., Motsinger, A. A., Bush, W. S., Coffey, C. S., and Moore, J. H. (2007). Genetic programming neural networks: A powerful bioinformatics tool for human genetics. *Applied Soft Computing*, 7(1):471–479.

[23] Sapin, E., Keedwell, E., and Frayling, T. (2014). Ant colony optimisation of decision trees for the detection of gene-gene interactions. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 57–61. IEEE.

[24] Schwarz, D. F., König, I. R., and Ziegler, A. (2010). On safari to random jungle: a fast implementation of random forests for high-dimensional data. *Bioinformatics*, 26(14):1752–1758.

[25] Shen, Y., Liu, Z., and Ott, J. (2012). Support vector machines with l 1 penalty for detecting gene-gene interactions. *International Journal of Data Mining and Bioinformatics*, 6(5):463–470.

[26] Sluga, D., Curk, T., Zupan, B., and Lotric, U. (2014). Heterogeneous computing architecture for fast detection of snp-snp interactions. *BMC Bioinformatics*, 15(1):216.

[27] Tang, W., Wu, X., Jiang, R., and Li, Y. (2009). Epistatic module detection for case-control studies: a bayesian model with a gibbs sampling strategy. *PLoS Genetics*, 5(5):e1000464.

[28] Taylor, M. B. and Ehrenreich, I. M. (2014). Genetic interactions involving five or more genes contribute to a complex trait in yeast. *PLoS Genetics*, 10(5):e1004324.

[29] Uppu, S. and Krishna, A. (2018). A deep hybrid model to detect multi-locus interacting snps in the presence of noise. *International Journal of Medical Informatics*, 119:134–151.

[30] Upstill-Goddard, R., Eccles, D., Fliege, J., and Collins, A. (2012). Machine learning approaches for the discovery of gene–gene interactions in disease data. *Briefings in bioinformatics*, 14(2):251–260.

[31] Wan, X., Yang, C., Yang, Q., Xue, H., Fan, X., Tang, N. L., and Yu, W. (2010). Boost: A fast approach to detecting gene-gene interactions in genome-wide case-control studies. *The American Journal of Human Genetics*, 87(3):325–340.

[32] Wan, X., Yang, C., Yang, Q., Xue, H., Tang, N. L., and Yu, W. (2009a). Megasnphunter: a learning approach to detect disease predisposition snps and high level interactions in genome wide association study. *BMC Bioinformatics*, 10(1):13.

[33] Wan, X., Yang, C., Yang, Q., Xue, H., Tang, N. L., and Yu, W. (2009b). Predictive rule inference for epistatic interaction detection in genome-wide association studies. *Bioinformatics*, 26(1):30–37.

[34] Wang, L. and Mehr, M. N. (2018). An optimization approach to epistasis detection. *European Journal of Operational Research*.

[35] Xie, M., Li, J., and Jiang, T. (2011). Detecting genome-wide epistases based on the clustering of relatively frequent items. *Bioinformatics*, 28(1):5–12.

[36] Yang, C., He, Z., Wan, X., Yang, Q., Xue, H., and Yu, W. (2008). Snpharvester: a filtering-based approach for detecting epistatic interactions in genome-wide association studies. *Bioinformatics*, 25(4):504–511.

[37] Yoshida, M. and Koike, A. (2011). Snpinterforest: a new method for detecting epistatic interactions. *BMC Bioinformatics*, 12(1):469.

[38] Yung, L. S., Yang, C., Wan, X., and Yu, W. (2011). Gboost: a gpu-based tool for detecting gene–gene interactions in genome–wide case control studies. *Bioinformatics*, 27(9):1309–1310.

[39] Yurochkin, M., Nguyen, X., et al. (2017). Multi-way interacting regression via factorization machines. In *Advances in Neural Information Processing Systems*, pages 2598–2606.

[40] Zhang, H., Wang, H., Dai, Z., Chen, M.-s., and Yuan, Z. (2012). Improving accuracy for cancer classification with a new algorithm for genes selection. *BMC Bioinformatics*, 13(1):298.

[41] Zhang, X., Huang, S., Zou, F., and Wang, W. (2010). Team: efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics*, 26(12):i217–i227.

[42] Zhang, Y. and Liu, J. S. (2007). Bayesian inference of epistatic interactions in case-control studies. *Nature Genetics*, 39(9):1167.

[43] Zou, F., Huang, H., Lee, S., and Hoeschele, I. (2010). Nonparametric bayesian variable selection with applications to multiple quantitative trait loci mapping with epistasis and gene–environment interaction. *Genetics*, 186(1):385–394.

CHAPTER 3. AN EXPLAINABLE MODEL FOR CROP YIELD PREDICTION

A paper submitted to *Scientific Reports*

Javad Ansarifar, Lizhi Wang, and Sotirios Archontoulis

3.1 Abstract

Crop yield prediction is crucial for global food security yet notoriously challenging due to multitudinous factors that jointly determine the yield, including genotype, environment, management, and their complex interactions. Integrating the power of optimization, machine learning, and agronomic insight, we present an explainable model referred to as the interaction regression model for crop yield prediction, which has three salient properties. First, it achieved a relative root mean square error of 8% or less in three Midwest states (Illinois, Indiana, and Iowa) in the US for both corn and soybean yield prediction, outperforming state-of-the-art machine learning algorithms. Second, it identified about a dozen environment by management interactions for corn and soybean yield, some of which are consistent with conventional agronomic wisdom whereas others challenge such wisdom and require additional analysis or experiment to prove or disprove. Third, it quantitatively dissected crop yield into contributions from weather, soil, management, and their interactions, allowing agronomists to pinpoint the factors that favorably or unfavorably affect the yield of a given location under a given weather and management scenario. The most significant contribution of the proposed prediction model is its capability to produce accurate prediction and explainable insights simultaneously. This was achieved by training the algorithm to select features and interactions that are spatially and temporally robust in order to balance prediction accuracy for the training data and generalizability to the test data.

3.2 Introduction

Predicting crop yield is crucial to addressing emerging challenges in food security, particularly in an era of global climate change. Accurate yield predictions not only help farmers make informed economic and management decisions but also support famine prevention efforts. Underlying crop yield prediction is a fundamental research question in plant biology, which is to understand how plant phenotype is determined by genotype (G), environment (E), management (M), and their interactions (G×E×M) [15, 20, 32, 24, 3, 17]. State-of-the-art crop yield prediction methods fall into three main categories: linear models, machine learning models, and crop models, which have complementary strengths and limitations. Linear models are explainable by quantifying the additive effect of each variable, but they often struggle to achieve high prediction accuracy due to the inability to capture the intrinsically nonlinear interactions among G, E, and M variables.

Machine learning models have been successfully used for crop yield prediction, including stepwise multiple linear regression [19], random forest [33], neural networks [40, 34, 16], convolutional neural networks [51], recurrent neural networks [63], weighted histograms regression [41], ensembles methods [58, 57], interaction based model [5], and association rule mining and decision tree [50]. Most of these studies were based on environmental and managerial variables only, due to lack of publicly available genotype data at the state or national scale. Some studies [50, 28, 8, 27] explored the relationship between genotype and grain yield from regional yield trials from a plant breeding perspective, which would be hard to scale up to statewide or nationwide predictions. Many machine learning algorithms are scalable to large datasets and have reasonably high prediction accuracy. However, due to the black-box nature of these models, prediction accuracy is sensitive to model structure and parameter calibration, and it can prove difficult to explain why predictions are accurate or inaccurate.

Crop models are another type of nonlinear models, including APSIM [35], DSSAT [9], FASSET [10], RZWQM [2], SWAP/WOFOST [21], and SOYGRO [42], which build upon the physiological understanding of plant and soil processes to develop biologically meaningful non-linear equations to predict crop yield and other phenotypes. These models provide explicit (albeit complex) ex-

planations of the interactions between traits and environmental conditions in different phases of the crop growth cycle. They also offer biological insights into causes of phenotypic variation [31]. Nevertheless, the collection of trait measurement data and calibration of model coefficients can be labor intensive and time consuming [11, 38, 47], computation speed could be low [58], and prediction accuracy may not be as high as some machine learning algorithms.

We propose a novel model, the interaction regression model, for crop yield prediction, which attempts to combine the strengths and avoid the limitations of the aforementioned approaches. At the core of this model lies a combinatorial optimization algorithm, which not only selects the most revealing E and M features but also detects their most pronounced interactions; the contributions of these features and interactions to the crop yield are then quantified with a multiple linear regression. To ensure the explainability of the results, we trained our algorithm to find features and interactions that are spatially and temporally robust, which means that they should be consistently predictive of crop yield across all counties in all years. As such, results from this model have the potential to propose biologically and agronomically insightful hypotheses on E×M interactions that can be validated experimentally. A similar concept of robust inference model in spatial-temporal models was presented in Santos and Erniel [52]. A measure of robustness was proposed in Nogueira et al. [44], which was based on the number of overlapping features selected using different subsets of training data. In our approach, the robustness measure is defined as the average prediction performance in multiple validation datasets at different temporal and spatial spectra. As such, our robustness definition allowed the algorithm to strike a balance between prediction accuracy and generalizability.

The proposed model has demonstrated notable performance in a comprehensive case study, in which it was compared with eight other machine learning models to predict corn and soybean yield in 293 counties of the states of Illinois, Indiana, and Iowa from 2015 to 2018. The proposed model not only achieved a less than 8% relative root mean square error (RRMSE) for both corn and soybean in all three states, outperforming all other machine learning models in the case study, but also produced explainable insights. In particular, our model identified 11 G×E interactions for

corn and 12 for soybean, and also dissected the total yield into contributions from weather, soil, management, and their interactions. To test the generalizability of the model in terms of both temporal and spatial extrapolation, we trained the model using historical data from two states up to 2017 and applied it to predict corn yield in a third state for 2018, and the resulting average RRMSE was less than 10%.

3.3 Methods

Let X denote the set of explanatory (including genotype, environment, and management) variables and y the crop yield of a given county for a given year. We propose the interaction regression model to describe the relationship between X and y as follows.

$$\hat{y}_i = \beta_0 + \sum_{j \in \mathcal{P}} X_{i,j}\beta_j + \sum_{m \in \mathcal{M}} b_m Z_{i,m}, \quad \forall i \in \mathcal{N}, \tag{3.1}$$

where,

- $\mathcal{N}$ is the set of sample observations (one sample per county per year),
- $\mathcal{P}$ is the set of explanatory variables,
- $\mathcal{M}$ is the set of interactions,
- $\hat{y}_i$ is predicted crop yield of sample i,
- β_0 is the intercept of crop yield,
- β_j is the additive effect of variable j,
- $X_{i,j}$ is the explanatory variable j of sample i,
- b_m is the effect of interaction m, and
- $Z_{i,m}$ is the interaction variable m of sample i.

Key to equation (4.1) is to decipher the interaction matrix Z from explanatory variables. We use a kernel-based approach to represent the interactions as

$$Z_{i,m} = \sum_{k \in \mathcal{K}} \delta_{m,k} K_k(X_i),$$

where,

- $K_k(\cdot)$ is the type k kernel function,
- $\mathcal{K}$ is the set of kernel functions that we use to describe nonlinear relationships between explanatory variables and crop yield, and
- $\delta_{m,k}$ is a binary variable indicating whether interaction m is best described by the type k kernel ($\delta_{m,k} = 1$) or not ($\delta_{m,k} = 0$).

In order to solve Equation (4.1), we propose an approach that consists of three major steps: data pre-processing, robust feature and interaction selection, and linear regression, as illustrated in Figure 3.1. Key elements of the three steps are summarized as follows.

3.3.1 Step 1: Data Pre-processing.

We collected weather data from the Iowa Environmental Mesonet [22], soil data from the Gridded Soil Survey Geographic Database [18], and management and yield performance data from the National Agricultural Statistics Service [55] for all 293 counties of the states of Illinois, Indiana, and Iowa from 1990 to 2018. Weather variables include precipitation (Prcp, mm), solar radiation (Srad, MJ/m^2), maximum temperature (Tmax, C°), and minimum temperature (Tmin, C°) from weeks 13 (late March) to 52 (late December). Soil variables include dry bulk density (BDdry, g cm^{-3}), clay percentage (clay, %), soil pH (pH), drained upper limit (dul, mm.mm^{-1}), soil saturated hydraulic conductivity (ksat, mm/day), drained lower limit (ll, mm.mm^{-1}), organic matter (om, %), sand percentage (sand, %), and saturated volumetric water content (sat, mm.mm^{-1}) at nine different depths of soil: 0-5, 5-10, 10-15, 15-30, 30-45, 45-60, 60-80, 80-100, and 100-120 cm. Management variables include acres planted at the county-level, weekly cumulative percentage of planted and

harvested acreages. We also created additional variables using the weather and management data based on agronomic insight to help enhance the performance of the model, such as growing degree days, number of rainy days, and heat units. Due to the lack of publicly available genotype data, we extracted two new variables using additional data from the National Agricultural Statistics Service [55] to account for the trend of genetic improvements [20]: (1) trend of historical yields and (2) trend of population density for corn and pod count for soybean. These two variables were put in the category of management variables. All variables were normalized to the $[0, 1]$ interval.

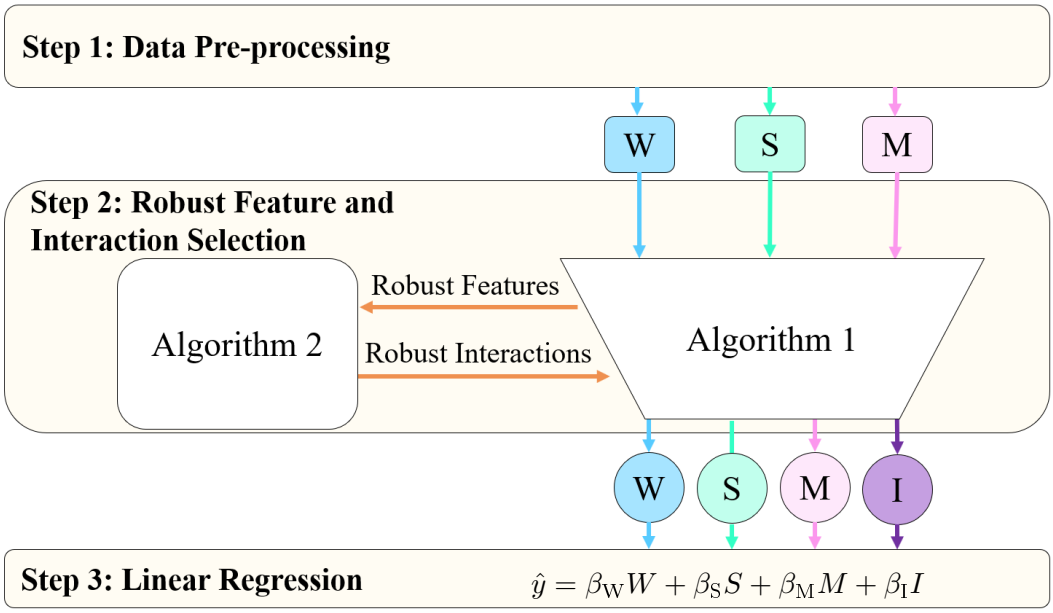

Figure 3.1 Illustration of the proposed explainable crop yield prediction model. Step 1 is data pre-processing. In step 2, Algorithms 1 and 2 select robust features and interactions, which are then used in step 3 to predict the crop yield with a multiple linear regression model. Here, $\hat{y}$ is the predicted yield, β_{W}, β_{S}, and β_{M} are, respectively, the additive effects of weather, soil, and management features, whereas β_{I} is the effect of E×M interactions.

3.3.2 Step 2: Robust Feature and Interaction Selection.

To avoid overfitting, we selected a subset of all explanatory variables (features) to predict crop yield. We applied elastic net regularization model to select a set of high-quality features for each category of weather, soil, and management, and then we used forward and backward stepwise selection to identify features and interaction that are spatially and temporally robust across different counties over different years. These robust features and interactions were selected using a similar algorithm from our previous study[6], which was modified to iterate between exploring new interactions and cross-validating their performances. Such process continues until a set of robust features and interactions has been discovered that lead to good prediction accuracy on the training data and generalizability on the validation data. The way interactions were represented in our model differs from the classical factorial interaction. However, they are also similar in the sense that our algorithm explores all possible factorial combinations to identify the most effect interactions to include in the model.

3.3.3 Step 3: Linear Regression.

The last step of the prediction model is a multiple linear regression, which attributes crop yield to additive contributions from weather, soil, management, and their interactions. As such, this prediction model combines the strengths of explainability of linear regression, prediction accuracy of machine learning, and agronomic insights of crop models.

More details about the kernel functions in Equation (4.1) and the algorithm for solving it are provided in Appendix 1.

3.4 Prediction Results

3.4.1 Prediction Accuracy Comparison with Other Machine Learning Models.

We compared the performance of the proposed algorithm with that of eight other machine learning algorithms from the literature: linear regression was implemented in R; stepwise regression

was implemented in R using the MASS package [48]; LASSO, ridge, and elastic net were implemented in R using the glmnet package [26]; random forest was implemented in R using the ranger package [62]; extreme gradient boosting (XGBoost) was implemented in R using the xgboost package [14]; and neural network was implemented in Python using the Sklearn package [46]. We fed all original explanatory variables as input to these eight algorithms. The linear regression algorithm uses all features without interaction selection; stepwise regression, Lasso regression, ridge regression, and elastic net have their default feature selection settings in the software packages without interaction selection; random forecast, xgboost, and neural network use different modeling structures for feature and interaction selection. As such, the different performances of these algorithms can be attributed to how they select features and interactions from the same set of explanatory data.

All nine algorithms were deployed to predict both corn and soybean yields in the states of Illinois, Indiana, and Iowa from 2015 to 2018. To predict yield for the test year t, the training data included all the explanatory (weather, soil, and management) and response (crop yield) data from 1990 to year $t-1$. A 10-fold CV over training and validation partitions was applied to tune the hyperparameters using a grid search approach. Prediction errors for two crops over four test years using nine algorithms are summarized in Table 3.1. More comparison in terms of the relative RMSE (RRMSE), the relative squared error (RSE), the mean absolute error (MAE), the relative absolute error (RAE), and the coefficient of determination (R^2) of nine models are reported in Appendix 2. These results suggested that the proposed model outperformed other models for all test years for both corn and soybean in all evaluation criteria. The test root mean square errors (RMSE) are also lower than what has been reported in the literature [63, 41, 50, 58]. In terms of the computation time, the proposed approach took approximately two hour for each test year, which was comparable with the neural network model.

Table 3.1 RMSE (in t/ha) of nine algorithms for corn and soybean yield prediction over four test years.

Model	Corn Test Year				Soybean Test Year			
	2015	2016	2017	2018	2015	2016	2017	2018
Linear Regression	1.39	1.33	1.19	0.96	0.52	0.48	0.42	0.43
Stepwise Regression	1.37	1.13	1.16	0.97	0.42	0.34	0.35	0.36
Lasso Regression	1.41	1.31	1.21	0.92	0.42	0.42	0.31	0.31
Ridge Regression	1.32	1.29	0.99	0.95	0.41	0.43	0.34	0.32
Elastic Net	1.25	1.26	1.03	0.93	0.40	0.40	0.32	0.33
Random Forest	1.30	1.20	1.06	0.94	0.34	0.37	0.28	0.39
XGBoost	1.50	1.37	1.24	1.08	0.43	0.46	0.40	0.44
Neural Network	1.24	0.82	0.95	0.93	0.40	0.37	0.31	0.40
Interaction Regression	**1.02**	**0.81**	**0.90**	**0.81**	**0.29**	**0.27**	**0.23**	**0.27**

3.4.2 Prediction Performance with Known Weather After Growing Season.

Figure 3.2 illustrates the prediction performance of the proposed model after the end of the growing season when all the weather data have been observed. These results indicate that the proposed model has an RRMSE lower than 8% in all three states (and most of the counties) over multiple years for both corn and soybean. In reference, prediction accuracy of other recent studies ranged from 7.6% mean absolute percentage error for corn using deep neural networks [37] to 16.7% RRMSE for corn using random forest [33]. According to the comparative table in Shahhosseini et al. (2020)[56], the RRMSE of the crop model was around 14-20%.

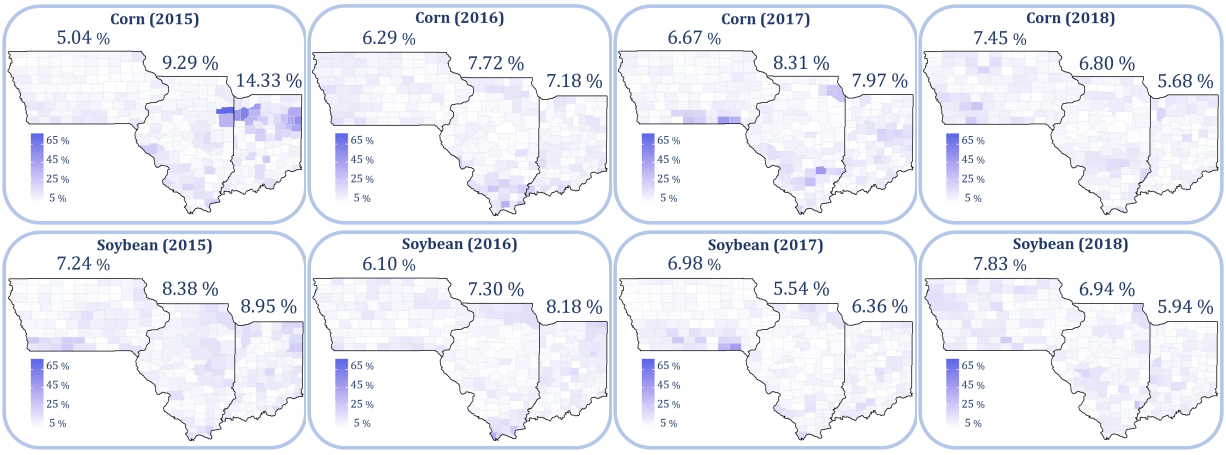

Figure 3.2 RRMSE for corn and soybean yield prediction from 2015 to 2018.

3.4.3 Prediction Performance with Updating Weather During Growing Season.

Crop yield prediction during the growing season is informative for farmers to make economic or management decisions, but it is also very challenging due to weather uncertainty. Our model was able to provide weekly predictions by integrating continuously updated weather data with future weather scenarios. The prediction accuracy is expected to improve over time as more actual weather observations become available to replace weather predictions. Our previous work using a crop model suggested that weather uncertainty decreased by 60% by mid July in Iowa for both corn and soybean [7].

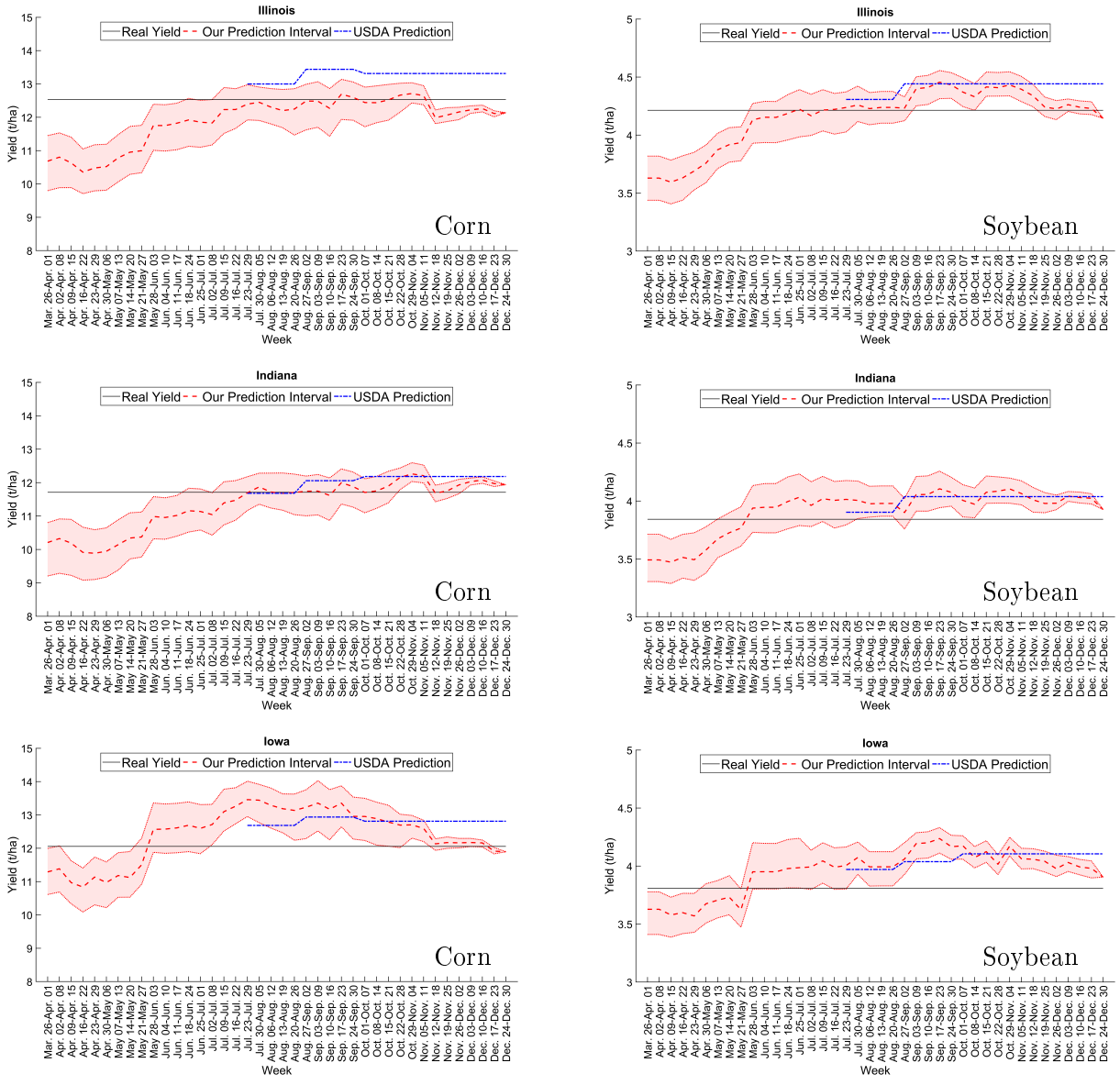

Figure 3.3 State-level predictions of corn and soybean during the growing season for three states in 2018. USDA predictions were released in August, September, and October. Our model provided weekly predictions based on observed weather information; prediction intervals were constructed using historical weather scenarios for yet-to-be-observed weather.

Figure 3.3 shows the predictions of corn and soybean yield during the growing season of 2018 in the three states, updated weekly to incorporate new weather data. The dashed red curve is the median prediction, and the pink interval is defined by the first and third quantiles under multiple weather scenarios, constructed using historical weather data. The dotted blue curves are USDA predictions, which were released in August, September, and October of 2018 at the state level. The solid black line indicates the actual state average yield, which was announced by USDA in February 2019. Compared with the USDA predictions, results from he proposed model have three advantages: (1) interval predictions throughout growing season with weekly updates, (2) county level (as opposed to state level) predictions, and (3) higher accuracy. The pattern of increased yield prediction from April to July was caused by weather and planting time in 2018, and it varied across different counties. Our prediction continues to update until the end of December, which is more than two months after the end of the growing season. This is because the model is able to capture factors that affect crop yield from crop maturity to harvest, such as adverse weather conditions during harvesting.

3.4.4 Temporal and Spatial Extrapolation Performance.

To show the performance of our model in the temporal and spatial extrapolation of yield, the prediction performance of the proposed Interaction-Regression model for corn and soybean in unseen counties at the test year 2018 are reported in Table 3.2. We created four datasets by removing the historical dataset of some counties from the training and validation sets. For the first three datasets, we removed data for Illinois (IL), Indiana (IN), and Iowa (IA), respectively; for the last dataset, we randomly picked 100 out of the 293 counties and removed all their data from training and validation sets. After training the model for each dataset, we used the learned model to predict crop yield of the unseen counties in the year 2018. The results suggest that the proposed approach has a satisfactory prediction performance in both temporal and spatial extrapolation.

Table 3.2 RMSE in t/ha (and RRMSE in %) of the interaction regression model for the extrapolation of crop yield for unseen counties at the year 2018. Each row shows the dataset by removing all historical information of counties. First test set refers to the prediction of counties with historical datasets in training and validation set at the test year 2018 (temporal extrapolation). Second test set refers to the prediction of unseen counties with no historical dataset in training and validation set at the test year 2018 (temporal and spatial extrapolation).

Crop	Training and validation sets	Test set	Training RMSE (RRMSE)	Validation RMSE (RRMSE)	Test RMSE (RRMSE)
Corn	IA and IN	IA and IN	0.56 (6.19%)	1.20 (10.3%)	1.52 (12.82%)
		IL			0.83 (6.67%)
	IA and IL	IA and IL	0.60 (6.61%)	0.82 (6.80%)	1.15 (9.37%)
		IN			0.79 (6.79%)
	IL and IN	IL and IN	0.59 (6.75%)	0.66 (5.93%)	0.71 (5.90%)
		IA			1.08 (8.98%)
	193 random counties	193 random counties	0.62 (6.85%)	0.68 (5.89%)	0.75 (6.23%)
		The other 100 counties			0.75 (6.30%)
Soybean	IA and IN	IA and IN	0.19 (6.51%)	0.20 (5.42%)	0.30 (7.86%)
		IL			0.37 (8.94%)
	IA and IL	IA and IL	0.19 (6.54%)	0.18 (4.81%)	0.30 (7.55%)
		IN			0.64 (16.77%)
	IL and IN	IL and IN	0.20 (6.87%)	0.18 (4.97%)	0.24 (6.09%)
		IA			0.85 (22.47%)
	193 random counties	193 random counties	0.20 (6.95%)	0.18 (4.96%)	0.30 (7.71%)
		The other 100 counties			0.29 (7.39%)

3.5 Explainable Insights

The proposed model was able to provide not only accurate predictions but also explainable insights, which could help farmers, breeders, and agronomists better understand the complex and interactive relationship among environment and management.

3.5.1 Additive and Interactive Effects.

Our model selected 202 robust features and 11 two-way interactions to predict the corn yield. Out of the 202 features, 155 were for weather, 37 for soil, and 10 for management. In reference, the total number of variables is 613 (including 440 for weather, 90 for soil, 83 for management), thus the total number of possible two-way interactions is $613^2 = 375,769$ (quadratic effects are considered self-interactions [4, 39]). These features and interactions were carefully selected to balance prediction accuracy with spatial and temporal consistency. As such, the same set of features and interactions apply to all counties in the three states for all years between 2015 and 2018. Similarly, our model selected 160 robust features (including 91 for weather, 59 for soil, and 10 for management) and 12 two-way interactions to predict the soybean yield. The contributions of the selected features and interactions for corn and soybean are visualized in Figure 3.4 in two circular graphs, in which the curves inside the inner circle indicate the variables involved in the two-way interactions, the bars in the first layer around the circle represent the effects of the interactions, and the bars in the second layer show the additive effects of the features. Positive and negative effects are illustrated with red and blue colors, respectively. A close-up view of the interactions are shown in Figure 3.5, in which all 11 interactions for corn and 12 for soybean are numbered.

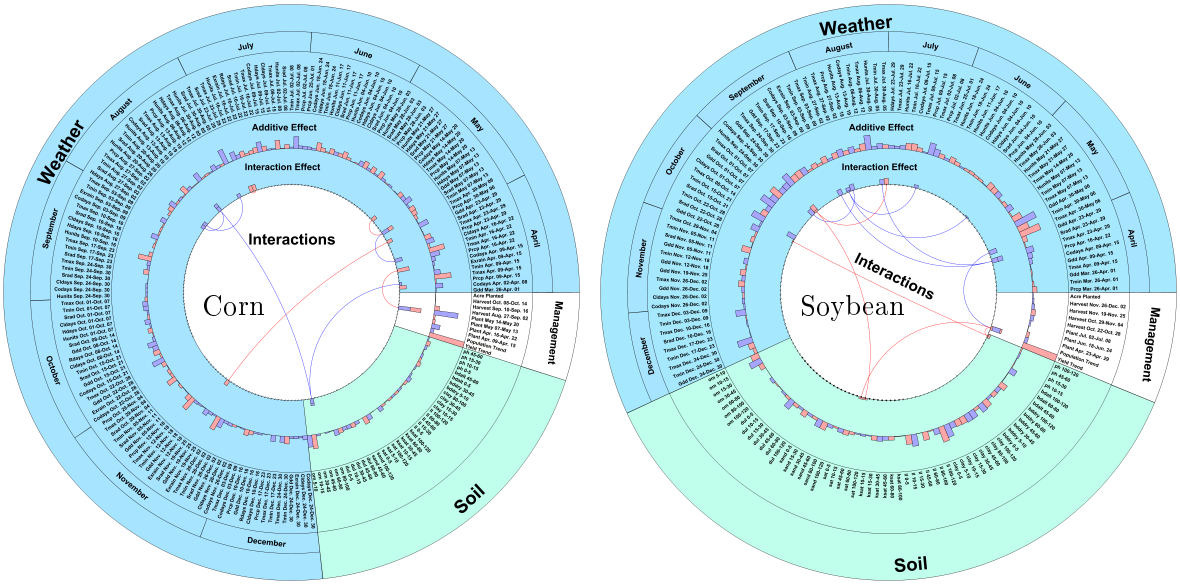

Figure 3.4 Additive and interactive effects for corn (left) and soybean (right). Curves inside the inner circle connect the two variables involved in the two-way interactions. The first layer outside the circle shows the effects of the interactions, and the second layer shows the additive effects of the variables. Positive and negative effects are illustrated with red and blue bars, respectively.

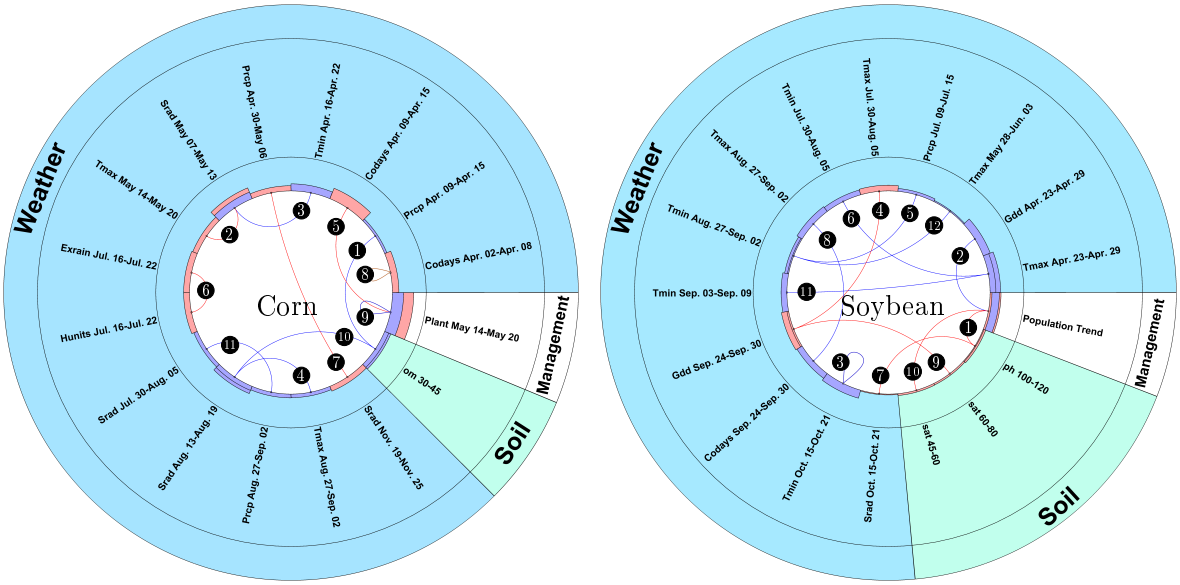

Figure 3.5 Interactions for corn (left) and soybean (right) that were discovered by the proposed model. Curves inside the inner circle connect the two variables involved in the interactions. The first layer outside the circle shows the positive (red) or negative (blue) effects of the interactions.

We explain the contributions of weather ($\beta_{\mathrm{W}}W$), soil ($\beta_{\mathrm{S}}S$), management ($\beta_{\mathrm{M}}M$), and their interactions ($\beta_{\mathrm{I}}I$) in all counties in 2015 and 2018 as violin plots in Figure 3.6. These results identified several high-impact features, including temperature, precipitation, soil organic matter, drained upper limit of soil, planting time, and yield trend. It was also revealed that weather conditions in earlier weeks of the growing season have more influences on yield than later ones, and that late planting time is associated with lower yield performance. These findings are consistent with results from field experimental studies [7, 36, 12, 23, 49, 45, 43].

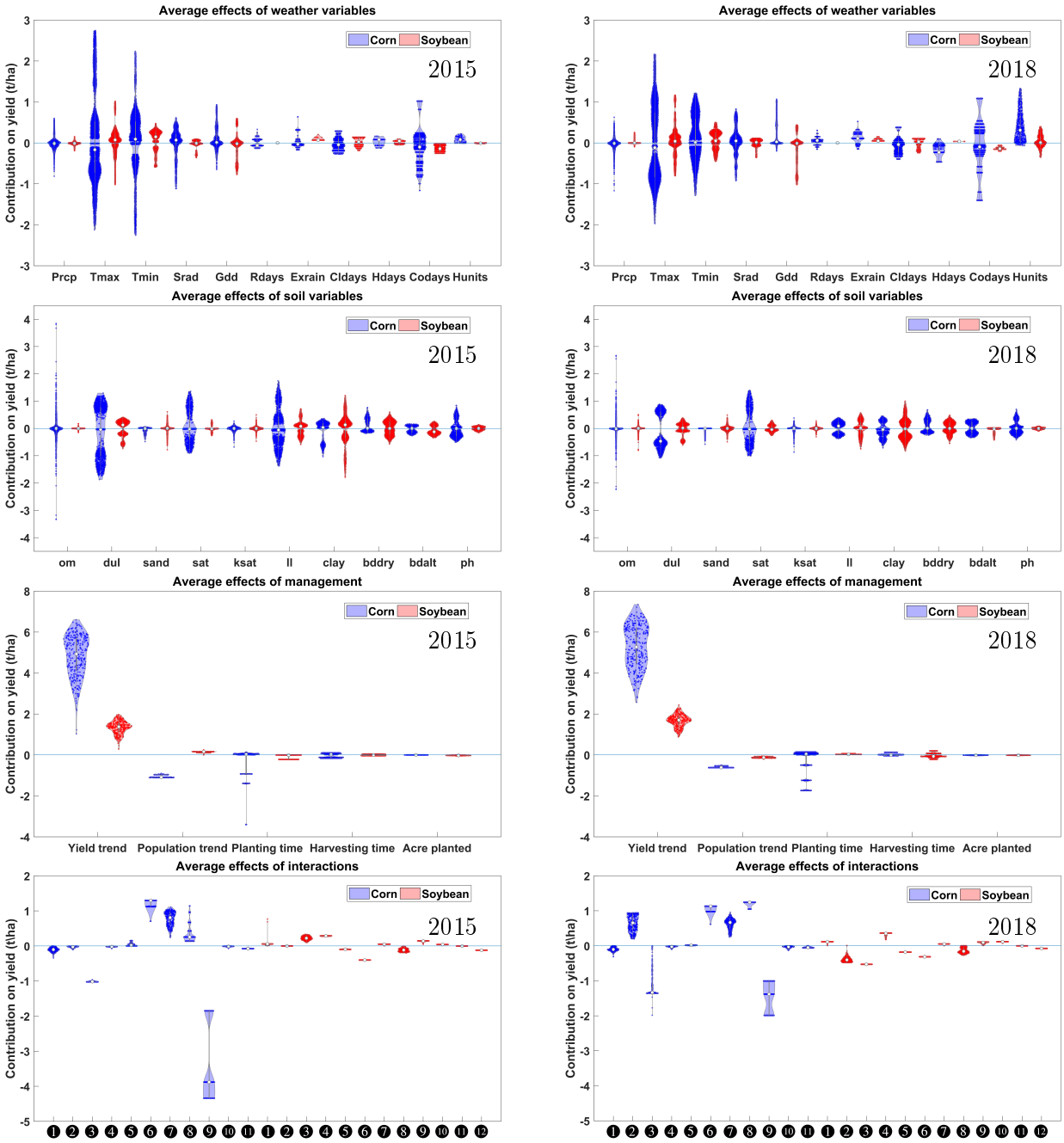

Figure 3.6 Violin plots of estimated contributions of weather (first row), soil (second row), management (third row) and interaction (fourth row) variables on corn and soybean yield in 2015 (left) and 2018 (right). Each dot on a violin plot represents a county level observation.

3.5.2 Insightful Interactions.

Figure 3.7 illustrates three of the interactions for corn using partial dependence plots, which is a popular way to show the marginal effect that one or two features have on the predicted outcome of a machine learning model.

- Two-way interaction ❹ for corn: the combination of low solar radiation and high maximum temperature during the late grain filling period negatively affects corn yields. This is consistent with agronomic intuition, as low solar radiation limits the energy for photosynthesis, and high maximum temperatures are associated with additional yield losses through tissue respiration and increased evapotranspiration stress.

- Self interaction ❽ for corn: average yield drops from 9.455 to 9.15 t/ha as the number of cold days in the week of April 2 increases from 0 to 4. This is insightful because the soil organic matter mineralization and soil water evaporation will slow down in low temperature, leading to delayed field operations due to reduced production of nitrogen and wetter soil surface. The upward trend of yield as the number of cold days increases from 4 to 7 days is counter-intuitive biologically, but it may reveal an important agronomic insight: when the low temperatures last long enough, farmers may start to take actions (e.g., more fertilization and irrigation) to offset its negative impact on corn yield.

- Self interaction ❾ for corn: completing planting by May 14 is ideal for the yield, and leaving 50% of planting unfinished by May 20 may reduce the yield by 1.25 t/ha. This is consistent with the well-known benefit of early planting [12]. It was also validated in 2019, when the weather-caused delay in planting in IL and IN led to decreased yields [55].

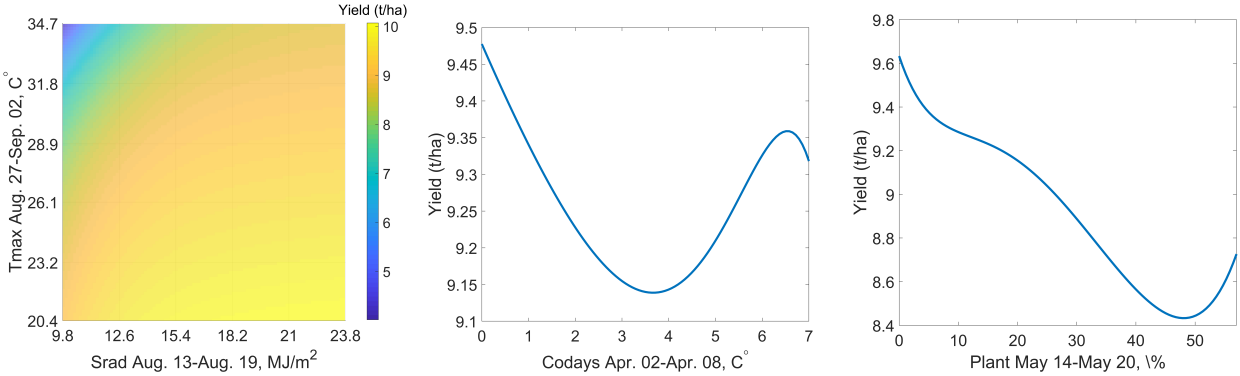

Figure 3.7 Partial dependence plots of interactions ❹ (left), ❽ (center), and ❾ (right) for corn.

Figure 3.8 illustrates two of the interactions for soybean using partial dependence plots.

- Self interaction ❸ for soybean: lower temperature, even near freezing, in mid- to late-October is favorable for soybean yield.
- Two-way interaction ❺ for soybean: high precipitation in mid July makes the yield sensitive to night temperature in late August; warmer nights may lead to a 0.45 t/ha higher yield than cooler nights. It has been reported that higher temperature will negatively impact soybean yield [61, 64]; our results further suggest that precipitation may also affect the extent of such impact. A possible interpretation is that higher temperature decelerates leaf senescence and increases remobilization of nitrogen and dry matter from vegetative tissues to grains, and such process may be more sensitive to temperature at a higher level of soil moisture.

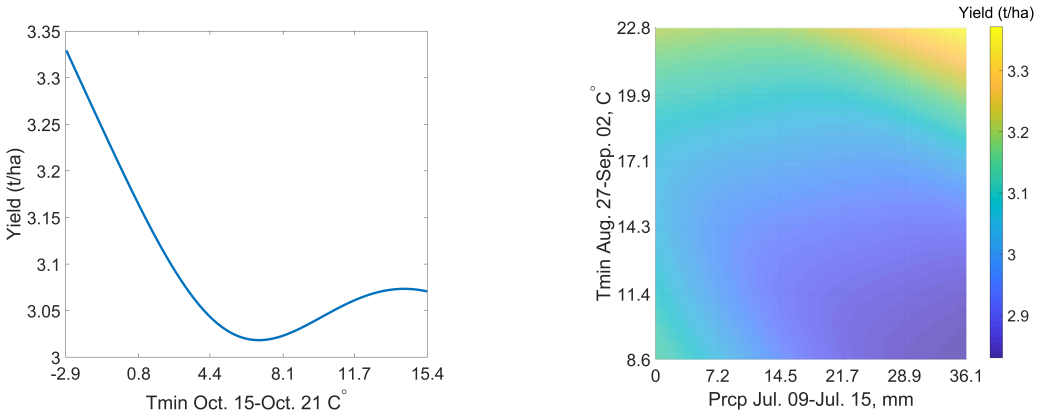

Figure 3.8 Partial dependence of interactions ❸ (left) and ❺ (right) for soybean.

3.5.3 Dissection of Crop Yield.

Breakdowns of observed yields in three states from 2015 to 2018 to contributions of weather ($\beta_{\mathrm{W}}W$), soil ($\beta_{\mathrm{S}}S$), management ($\beta_{\mathrm{M}}M$), and their interactions ($\beta_{\mathrm{I}}I$) are shown in Figures 3.9 and 3.10 for corn and soybean, respectively. These contributions differ by county and change over time. In 2015, weather was the deciding variable for the yield, whereas interactions played a more important role in 2018. Due to the relatively static nature and lack of dramatic changes across the three Midwest states, soil variables demonstrated a lower effect on crop yield than the dynamic weather, management, and their interactions [47, 65].

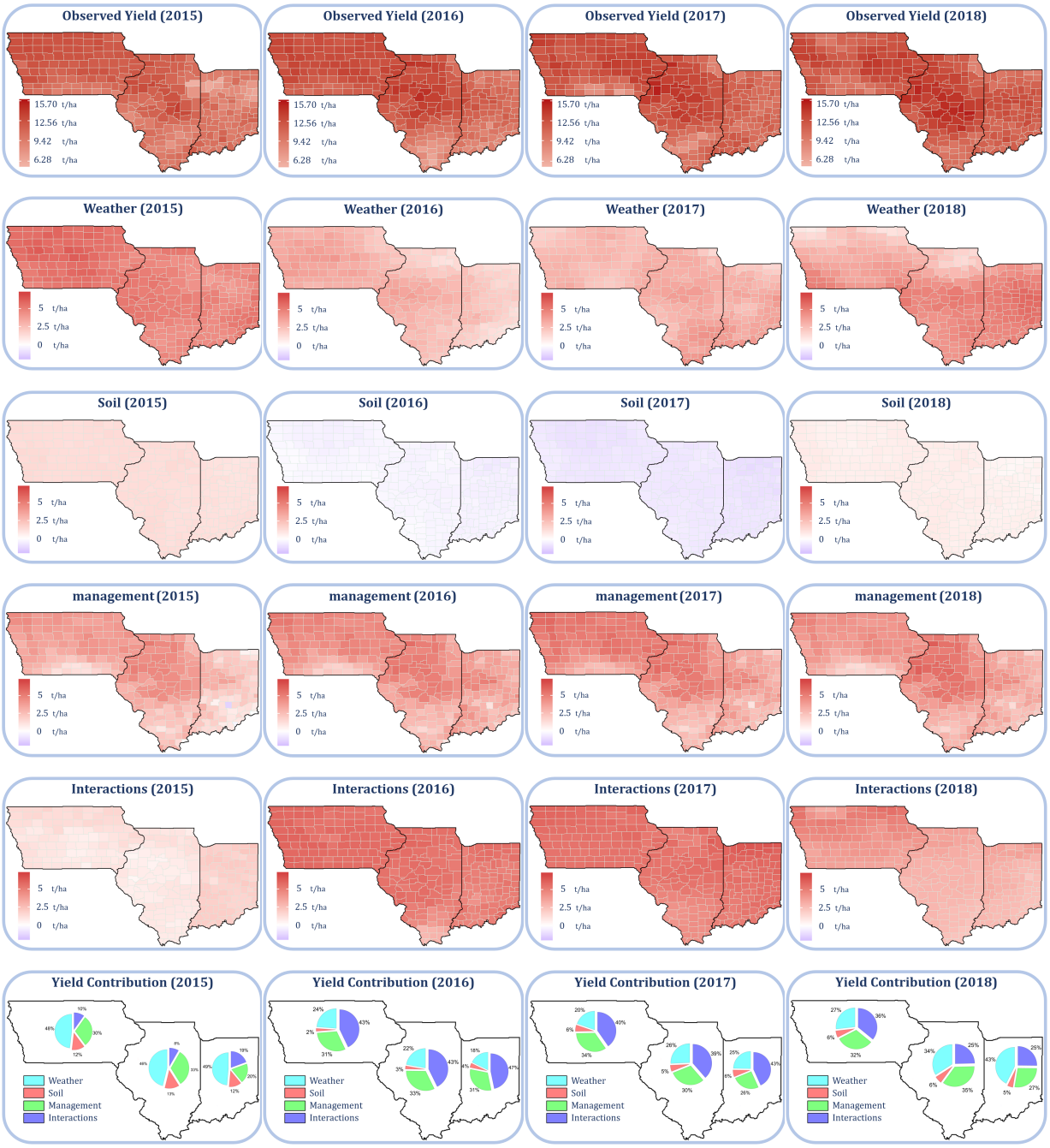

Figure 3.9 Breakdown of observed corn yield in three states from 2015 to 2018 to contributions of weather ($\beta_{\mathrm{W}}W$), soil ($\beta_{\mathrm{S}}S$), management ($\beta_{\mathrm{M}}M$), and their interactions ($\beta_{\mathrm{I}}I$).

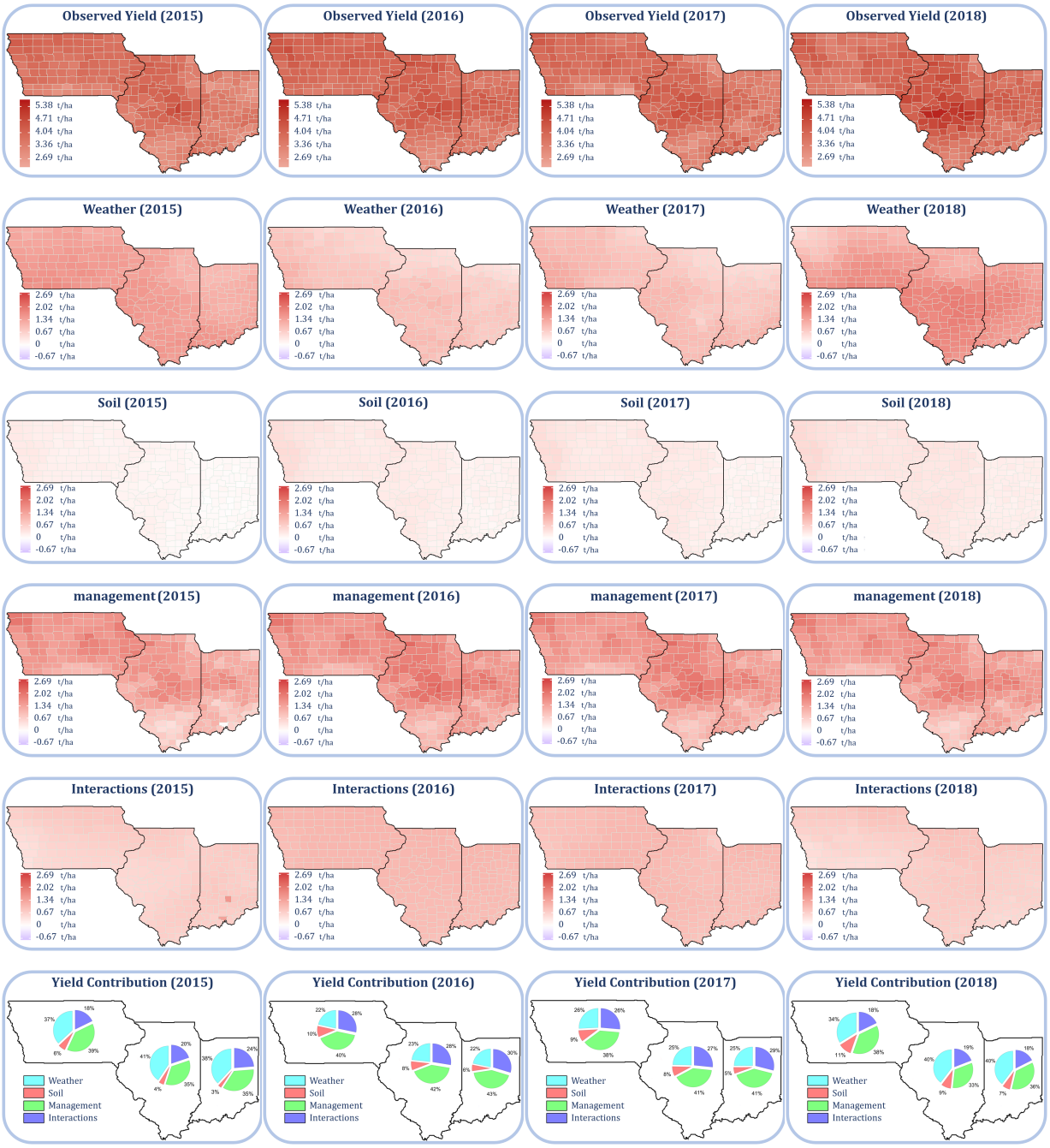

Figure 3.10 Breakdown of observed soybean yield in three states from 2015 to 2018 to contributions of weather ($\beta_{\mathrm{W}}W$), soil ($\beta_{\mathrm{S}}S$), management ($\beta_{\mathrm{M}}M$), and their interactions ($\beta_{\mathrm{I}}I$).

3.6 Conclusion

We proposed an explainable model for crop yield prediction, which made three major contributions. First, it outperformed state-of-the-art machine learning algorithms with respect to prediction accuracy in a comprehensive case study, which used historical data of three Midwest states from 1990 to 2018. Second, it was able to identify about a dozen E×M interactions for corn and soybean yield, which are spatially and temporally robust and can be used to form counter-intuitive, insightful, and testable hypotheses. Third, it was able to explain the contributions of weather, soil, management, and their interactions to crop yield. Achieving these three contributions simultaneous is particularly significant, since no other crop yield prediction algorithms have been able to satisfactorily address both prediction accuracy and explainability.

The proposed model and computational experiments are not without limitations. For example, the robust feature and interaction selection algorithms were heuristic in nature, which can find high quality solutions efficiently but do not guarantee global optimality. The performance of the algorithm may be further improved by applying more advanced techniques for hyperparameter tuning [13]. Due to lack of publicly available information on genotype and management, the G, W, S, and M data used in our case study may be disproportional to their true contributions to crop yield. However, the proposed modeling approach was designed for both discrete and continuous explanatory variables and capable of analyzing all G, W, S, and M variables and their interactions. Future research should explore the possibility of including additional data (such as high-dimensional genotype data, plant traits, detailed management strategies, and satellite images) to further improve prediction accuracy and make more biologically and agronomically insightful discoveries.

3.7 References

[1] Abendroth, L. J., Elmore, R. W., Boyer, M. J., and Marlay, S. K. (2011). Corn growth and development.

[2] Ahuja, L. and Ma, L. (2011). *Methods of introducing system models into agricultural research*. American Society of Agronomy.

[3] Alminana, M., Escudero, L., Landete, M., Monge, J., Rabasa, A., and Sánchez-Soriano, J. (2010). Wische: A DSS for water irrigation scheduling. *Omega*, 38(6):492–500.

[4] Alvarez, R. and Grigera, S. (2005). Analysis of soil fertility and management effects on yields of wheat and corn in the rolling pampa of Argentina. *Journal of Agronomy and Crop Science*, 191(5):321–329.

[5] Ansarifar, J., Akhavizadegan, F., and Wang, L. (2020). Performance prediction of crosses in plant breeding through genotype by environment interactions. *Scientific Reports*, 10(1):1–11.

[6] Ansarifar, J. and Wang, L. (2019). New algorithms for detecting multi-effect and multi-way epistatic interactions. *Bioinformatics*, 35(24):5078–5085.

[7] Archontoulis, S. V., Castellano, M. J., Licht, M. A., Nichols, V., Baum, M., Huber, I., Martinez-Feria, R., Puntel, L., Ordóñez, R. A., Iqbal, J., et al. (2020). Predicting crop yields and soil-plant nitrogen dynamics in the US corn belt. *Crop Science*, 60(2):721–738.

[8] Basnet, B. R., Crossa, J., Dreisigacker, S., Pérez-Rodríguez, P., Manes, Y., Singh, R. P., Rosyara, U. R., Camarillo-Castillo, F., and Murua, M. (2019). Hybrid wheat prediction using genomic, pedigree, and environmental covariables interaction models. *The Plant Genome*, 12(1):1–13.

[9] Basso, B., Liu, L., and Ritchie, J. T. (2016). A comprehensive review of the CERES-wheat,-maize and -rice models' performances. In *Advances in Agronomy*, volume 136, pages 27–132. Elsevier.

[10] Bassu, S., Asseng, S., Motzo, R., and Giunta, F. (2009). Optimising sowing date of durum wheat in a variable mediterranean environment. *Field Crops Research*, 111(1-2):109–118.

[11] Bassu, S. et al. (2014). How do various maize crop models vary in their responses to climate change factors? *Global Change Biology*, 20(7):2301–2320.

[12] Baum, M., Archontoulis, S., and Licht, M. (2019). Planting date, hybrid maturity, and weather effects on maize yield and crop stage. *Agronomy Journal*, 111(1):303–313.

[13] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305.

[14] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

[15] Cooper, M., Tang, T., Gho, C., Hart, T., Hammer, G., and Messina, C. (2020). Integrating genetic gain and gap analysis to predict improvements in crop productivity. *Crop Science*.

[16] Crane-Droesch, A. (2018). Machine learning methods for crop yield prediction and climate change impact assessment in agriculture. *Environmental Research Letters*, 13(11):114003.

[17] Dai, Z. and Li, Y. (2013). A multistage irrigation water allocation model for agricultural land-use planning under uncertainty. *Agricultural Water Management*, 129:69–79.

[18] Database, G. S. S. G. https://gdg.sc.egov.usda.gov.

[19] Drummond, S. T., Sudduth, K. A., Joshi, A., Birrell, S. J., and Kitchen, N. R. (2003). Statistical and neural methods for site–specific yield prediction. *Transactions of the ASAE*, 46(1):5.

[20] Duvick, D. (2005). Genetic progress in yield of United States maize (zea mays l.). *Maydica*, 50(3/4):193.

[21] Eitzinger, J., Trnka, M., Hösch, J., Žalud, Z., and Dubrovskỳ, M. (2004). Comparison of ceres, wofost and swap models in simulating soil water content during growing season under different soil conditions. *Ecological Modelling*, 171(3):223–246.

[22] Environmental Mesonet, I. https://mesonet.agron.iastate.edu.

[23] Fan, Y., Li, H., and Miguez-Macho, G. (2013). Global patterns of groundwater table depth. *Science*, 339(6122):940–943.

[24] Filippi, C., Mansini, R., and Stevanato, E. (2017). Mixed integer linear programming models for optimal crop selection. *Computers & Operations Research*, 81:26–39.

[25] Forecast and of Cropping sysTemS (FACTS), A. https://crops.extension.iastate.edu/facts/weather-tool.

[26] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1.

[27] González-Camacho, J. M., Crossa, J., Pérez-Rodríguez, P., Ornella, L., and Gianola, D. (2016). Genome-enabled prediction using probabilistic neural network classifiers. *BMC Genomics*, 17(1):208.

[28] González-Camacho, J. M., Ornella, L., Pérez-Rodríguez, P., Gianola, D., Dreisigacker, S., and Crossa, J. (2018). Applications of machine learning methods to genomic selection in breeding wheat for rust resistance. *The Plant Genome*, 11(2):1–15.

[29] Hastie, T. J. and Pregibon, D. (2017). Generalized linear models. In *Statistical Models in S*, pages 195–247. Routledge.

[30] Hawkins, E., Fricker, T. E., Challinor, A. J., Ferro, C. A., Ho, C. K., and Osborne, T. M. (2013). Increasing influence of heat stress on french maize yields from the 1960s to the 2030s. *Global Change Biology*, 19(3):937–947.

[31] Heslot, N., Akdemir, D., Sorrells, M., and Jannink, J.-L. (2014). Integrating environmental covariates and crop modeling into the genomic selection framework to predict genotype by environment interactions. *Theoretical and Applied Genetics*, 127(2):463–480.

[32] Hipólito, J., Boscolo, D., and Viana, B. F. (2018). Landscape and crop management strategies to conserve pollination services and increase yields in tropical coffee farms. *Agriculture, Ecosystems & Environment*, 256:218–225.

[33] Jeong, J. H., Resop, J. P., Mueller, N. D., Fleisher, D. H., Yun, K., Butler, E. E., Timlin, D. J., Shim, K.-M., Gerber, J. S., Reddy, V. R., et al. (2016). Random forests for global and regional crop yield predictions. *PLoS One*, 11(6).

[34] Kaul, M., Hill, R. L., and Walthall, C. (2005). Artificial neural networks for corn and soybean yield prediction. *Agricultural Systems*, 85(1):1–18.

[35] Keating, B. A., Carberry, P. S., Hammer, G. L., Probert, M. E., Robertson, M. J., Holzworth, D., Huth, N. I., Hargreaves, J. N., Meinke, H., Hochman, Z., et al. (2003). An overview of APSIM, a model designed for farming systems simulation. *European Journal of Agronomy*, 18(3-4):267–288.

[36] Kessler, A., Archontoulis, S. V., and Licht, M. A. (2020). Soybean yield and crop stage response to planting date and cultivar maturity in Iowa, USA. *Agronomy Journal*, 112(1):382–394.

[37] Kim, N., Ha, K.-J., Park, N.-W., Cho, J., Hong, S., and Lee, Y.-W. (2019). A comparison between major artificial intelligence models for crop yield prediction: Case study of the midwestern United States, 2006–2015. *ISPRS International Journal of Geo-Information*, 8(5):240.

[38] Lamsal, A., Welch, S., Jones, J., Boote, K., Asebedo, A., Crain, J., Wang, X., Boyer, W., Giri, A., Frink, E., et al. (2017). Efficient crop model parameter estimation and site characterization using large breeding trial data sets. *Agricultural Systems*, 157:170–184.

[39] Leeper, R., Runge, E., and Walker, W. (1974). Effect of plant-available stored soil moisture on corn yields. i. constant climatic conditions 1. *Agronomy Journal*, 66(6):723–727.

[40] Liu, J., Goering, C., and Tian, L. (2001). A neural network for setting target corn yields. *Transactions of the ASAE*, 44(3):705.

[41] Marko, O., Brdar, S., Panic, M., Lugonja, P., and Crnojevic, V. (2016). Soybean varieties portfolio optimisation based on yield prediction. *Computers and Electronics in Agriculture*, 127:467–474.

[42] Monsi, M. and Saeki, T. (2005). On the factor light in plant communities and its importance for matter production. *Annals of Botany*, 95(3):549.

[43] Nichols, V. A., Ordonez, R. A., Wright, E. E., Castellano, M. J., Liebman, M., Hatfield, J. L., Helmers, M., and Archontoulis, S. V. (2019). Maize root distributions strongly associated with water tables in Iowa, USA. *Plant and Soil*, 444(1-2):225–238.

[44] Nogueira, S., Sechidis, K., and Brown, G. (2017). On the stability of feature selection algorithms. *The Journal of Machine Learning Research*, 18(1):6345–6398.

[45] Pasley, H. R., Camberato, J. J., Cairns, J. E., Zaman-Allah, M., Das, B., and Vyn, T. J. (2020). Nitrogen rate impacts on tropical maize nitrogen use efficiency and soil nitrogen depletion in eastern and southern Africa. *Nutrient Cycling in Agroecosystems*, pages 1–12.

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

[47] Puntel, L. A., Pagani, A., and Archontoulis, S. V. (2019). Development of a nitrogen recommendation tool for corn considering static and dynamic variables. *European Journal of Agronomy*, 105:189–199.

[48] Ripley, B. et al. (2011). Mass: support functions and datasets for venables and ripley's mass. *R Package Version*, pages 7–3.

[49] Rizzo, G., Edreira, J. I. R., Archontoulis, S. V., Yang, H. S., and Grassini, P. (2018). Do shallow water tables contribute to high and stable maize yields in the US corn belt? *Global Food Security*, 18:27–34.

[50] Romero, J. R., Roncallo, P. F., Akkiraju, P. C., Ponzoni, I., Echenique, V. C., and Carballido, J. A. (2013). Using classification algorithms for predicting durum wheat yield in the province of Buenos Aires. *Computers and Electronics in Agriculture*, 96:173–179.

[51] Russello, H. (2018). Convolutional neural networks for crop yield prediction using satellite images. *IBM Center for Advanced Studies*.

[52] Santos, J. and Barrios, E. (2019). Robust inference in semiparametric spatial-temporal models. *Communications in Statistics-Simulation and Computation*, pages 1–20.

[53] Schauberger, B., Archontoulis, S., Arneth, A., Balkovic, J., Ciais, P., Deryng, D., Elliott, J., Folberth, C., Khabarov, N., Müller, C., et al. (2017). Consistent negative response of us crops to high temperatures in observations and crop models. *Nature Communications*, 8(1):1–9.

[54] Schlenker, W. and Roberts, M. J. (2009). Nonlinear temperature effects indicate severe damages to us crop yields under climate change. *Proceedings of the National Academy of Sciences*, 106(37):15594–15598.

[55] Service, N. A. S. https://quickstats.nass.usda.gov.

[56] Shahhosseini, M., Hu, G., and Archontoulis, S. V. (2020). Forecasting corn yield with machine learning ensembles. *arXiv preprint arXiv:2001.09055*.

[57] Shahhosseini, M., Hu, G., and Pham, H. (2019a). Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. *arXiv preprint arXiv:1908.05287*.

[58] Shahhosseini, M., Martinez-Feria, R. A., Hu, G., and Archontoulis, S. V. (2019b). Maize yield and nitrate loss prediction with machine learning algorithms. *arXiv preprint arXiv:1908.06746*.

[59] Venables, W. N. and Ripley, B. D. (2013). *Modern applied statistics with S-PLUS*. Springer Science & Business Media.

[60] Walsh, M. K., Backlund, P. W., Buja, L., DeGaetano, A., Melnick, R., Prokopy, L., Takle, E., Todey, D., and Ziska, L. (2020). Climate indicators for agriculture. *USDA Technical Bulletin 1953*, pages 1–70.

[61] Wilhelm, W. and Wortmann, C. S. (2004). Tillage and rotation interactions for corn and soybean grain yield as affected by precipitation and air temperature. *Agronomy Journal*, 96(2):425–432.

[62] Wright, M. N. and Ziegler, A. (2015). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *arXiv preprint arXiv:1508.04409*.

[63] You, J., Li, X., Low, M., Lobell, D., and Ermon, S. (2017). Deep Gaussian process for crop yield prediction based on remote sensing data. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[64] Zhao, C., Liu, B., Piao, S., Wang, X., Lobell, D. B., Huang, Y., Huang, M., Yao, Y., Bassu, S., Ciais, P., et al. (2017). Temperature increase reduces global yields of major crops in four independent estimates. *Proceedings of the National Academy of Sciences*, 114(35):9326–9331.

[65] Zipper, S. C., Soylu, M. E., Booth, E. G., and Loheide, S. P. (2015). Untangling the effects of shallow groundwater and soil texture as drivers of subfield-scale yield variability. *Water Resources Research*, 51(8):6338–6358.

[66] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

3.8 Appendix 1: Details of Prediction Model

The prediction model can be formulated in Equation (3.2) as follows:

$$\hat{y}_i = \beta_0 + \sum_{j \in \mathcal{P}} X_{i,j}\beta_j + \sum_{m \in \mathcal{M}} b_m Z_{i,m}, \quad \forall i \in \mathcal{N}. \tag{3.2}$$

Key to model element in Equation (3.2) is to decipher the interaction matrix Z from the input data. In this paper, we use a kernel-based approach to represent the interactions as

$$Z_{i,m} = \sum_{k \in \mathcal{K}} \delta_{m,k} K_k(r_{i,m}(X_i, \alpha_m)),$$

where vector $\alpha_m \in \{0, 0.5, 1\}^{|\mathcal{P}|}$ indicates variables that trigger the interaction m; $r_{i,m}$ defines relative difference between involved variables in interactions m of county i; $K(\cdot)$ is the kernel function; and where $\delta_{m,k}$ is a binary variable indicating whether interaction m is best described by the type k kernel ($\delta_{m,k} = 1$) or not ($\delta_{m,k} = 0$). The proposed model specify the best type of kernel by satisfying the constraint $\sum_{k \in \mathcal{K}} \delta_{m,k} = 1$. Vector α_m specifies which variables trigger interaction m by assigning one value among three options $\{0, 0.5, 1\}$ to $\mathcal{P}$ variables. If $\alpha_{m,j} = 0.5$, then variable j is not involved in interaction m. If $\alpha_{m,j} \neq 0$, then variable j trigger interaction m. We define the interactions such that, the proposed model is able to capture the non-linear self-effects of variables (self-interaction) as well as two-way interactions between variables on yield. Therefore, $r_{i,m}$ of two-way interaction between two variables j and l ($\sum_{j \in \mathcal{P}} |\alpha_{m,j} - 0.5| = 1, \alpha_{m,j} \neq 0.5, \alpha_{m,l} \neq 0.5, j < l$) is defined as

$$r_{i,m}(X_{i,:}, \alpha_m) = (2\alpha_{m,j} - 1)(X_{i,j} + \alpha_{m,j} - 1) + (2\alpha_{m,l} - 1)(-X_{i,l} - \alpha_{m,l} + 1)$$

where the $r_{i,m}$ of self interaction of variable j ($\sum_{j \in \mathcal{P}} |\alpha_{m,j} - 0.5| = 0.5, \alpha_{m,j} \neq 0.5$) is defined as

$$r_{i,m}(X_{i,:}, \alpha_m) = (2\alpha_{m,j} - 1)(X_{i,j} + \alpha_{m,j} - 1)$$

In this research, kernel function $K(\cdot)$ has six possible variants:

$$K_k(r_{i,m}) = \begin{cases} \text{Linear kernel: } r_{i,m}^2 & k = 1 \\ \text{Squared exponential kernel: } \sigma_f^2 \exp\left(-\frac{1}{2}\frac{r_{i,m}^2}{\sigma_l^2}\right) & k = 2 \\ \text{Exponential kernel: } \sigma_f^2 \exp\left(-\frac{r_{i,m}}{\sigma_l^2}\right) & k = 3 \\ \text{Matern } 3/2\text{: } \sigma_f^2 \left(1 + \frac{\sqrt{3} r_{i,m}}{\sigma_l}\right) \exp\left(-\frac{\sqrt{3} r_{i,m}}{\sigma_l}\right) & k = 4 \\ \text{Matern } 5/2\text{: } \sigma_f^2 \left(1 + \frac{\sqrt{5} r_{i,m}}{\sigma_l} + \frac{\sqrt{5} r_{i,m}^2}{\sigma_l^2}\right) \exp\left(-\frac{\sqrt{5} r_{i,m}}{\sigma_l}\right) & k = 5 \\ \text{Rational quadratic kernel: } \sigma_f^2 \left(1 + \frac{r_{i,m}^2}{2\theta\sigma_l^2}\right)^{-\theta} & k = 6 \end{cases}.$$

Kernel function has three positive-valued parameters: σ_f, σ_l, and θ, which are signal standard deviation, characteristic length scale, and scale-mixture parameters, respectively. The non-linearity of yield in relation to the predictors comes from the kernel function that each interaction has.

3.8.1 Step 1: Data Preprocessing.

We collected weather, soil, management, and yield performance data from publicly available sources for all counties of the states of Illinois, Indiana, and Iowa from 1990 to 2018.

- Weather data were collected from the Iowa Environmental Mesonet [22], which included four daily surface weather parameters at 1 km^2 spatial resolution: precipitation (Prcp, mm), solar radiation (Srad, MJ/m^2), maximum temperature (Tmax, C°), and minimum temperature (Tmin, C°). Weather data from January to March were excluded, and only weeks 13 (late March) to 52 (late December) data were used in the model.

- Soil data were acquired from the Gridded Soil Survey Geographic Database [18], which included ten parameters at 1 km^2 spatial resolution: dry bulk density (BDdry, g cm^{-3}), clay percentage (clay, %), soil pH (pH), drained upper limit (dul, $mm.mm^{-1}$), soil saturated hydraulic conductivity (ksat, mm/day), drained lower limit (ll, $mm.mm^{-1}$), organic matter (om, %), sand percentage (sand, %), and saturated volumetric water content (sat, $mm.mm^{-1}$). All of these ten parameters were available at nine different depths of soil: 0-5, 5-10, 10-15, 15-30, 30-45, 45-60, 60-80, 80-100, and 100-120 cm.

- Management data were acquired from the National Agricultural Statistics Service [55], which included acres planted at the county-level, the weekly cumulative percentage of planting process and harvested fields at the state-level. More management data, including the weekly cumulative percentage of silking and emerging processes for corn and the weekly cumulative percentage of blooming and emerging processes for soybean, were collected from the National Agricultural Statistics Service [55]. However, we found that these management variables did not improve the prediction accuracy, since Algorithm 1 did not select them as robust features,

which is probably because management data are for the state level, whereas yield prediction is at the county level. Due to the lack of publicly available genotype data, we constructed two new features, i.e., the trend of historical yields and trend of population density for corn and pod count for soybean from the National Agricultural Statistics Service [55] to represent the trend of genetic improvements. We combined these features with management data.

- Yield performance data were also acquired from the National Agricultural Statistics Service [55], which included observed average yield performance between 1990 and 2018 for corn and soybean for all 293 counties in the states of Illinois, Indiana, and Iowa.

We also estimated additional features using the weather and management data based on agronomic insight to help enhance the performance of the model. The following weather variables were calculated from the raw weather data and added to the dataset:

- Growing degree days (Gdd, C°), which is $\max\{0, \text{mean(Tmax,Tmin)} - 10\}$, which is a largely used by agronomists and faster to track crop development [1].
- Number of rainy days (Rdays), which defined as the number of days with rain above 5 mm and below 24 mm in a week [60, 25].
- Number of extreme rainy days (Exrain), which is the number of days with rain above 24 mm in a week [47].
- Number of heat days (Hdays), which is the number of days with Tmax above 34 C° in a week [30, 54, 53].
- Number of cold days (Codays), which is the number of days with Tmin below 5 C° in a week [60, 25].
- Number of cloudy days (Cldays), which is the number of days with solar radiation below 10 MJ/m^2 in a week [60, 25].
- Heat units (Hunits), which are the summation of $\max\{0, \text{Tmax} - 34\}$ of a week [30, 54, 53].

3.8.2 Step 2: Robust Feature and Interaction Selection.

To avoid overfitting, we selected a subset of all explanatory variables (features) to predict crop yield, so that in Equation (4.1) we have $\beta_i = 0$ for all variables that are not selected. We also required the selected features to be spatially and temporally robust across different counties over different years. The performance of our feature selection algorithm was evaluated using a time-wise F-fold (4-fold in our case study) cross validation, as shown in Figure 3.11. Each fold $f \in \{1, ..., F\} = \mathcal{F}$ is corresponding to a particular test year for prediction. For each fold, we considered the partition of data related to two previous years from a test year as the validation set and dataset corresponding to the rest of the years to 1990 as a training set. We denote indices set $\mathcal{N}_f^{\mathrm{Tr}}, \mathcal{N}_f^{\mathrm{V}}, \mathcal{N}_f^{\mathrm{Te}}$ for training, validation, and test datasets for each fold $f \in \mathcal{F}$.

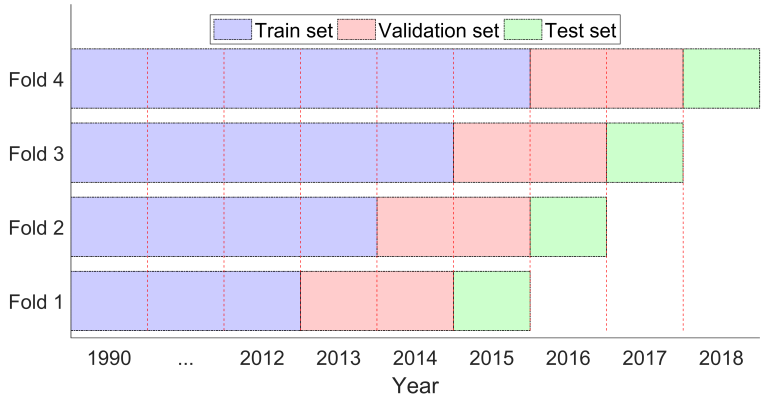

Figure 3.11 Partition of training, validation, and test datasets for cross-validation.

We cast the prediction problem as the following optimization model. The objective function (3.3) is our definition of the robustness measure. For any prediction $\hat{y}$ and parameters α and δ, $\zeta_{\mathrm{CV}}^{\mathrm{V}}(\hat{y})$ measures the average RMSE for all F folds of validation datasets. This definition captures temporal and spatial robustness by ensuring, respectively, that the same set of features $\mathcal{P}$ is used for different test years and that the same set of β^f is used for all counties in the same fold.

$$\min \quad \zeta_{\mathrm{CV}}^{\mathrm{V}} = \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} \sqrt{\frac{1}{|\mathcal{N}_f^{\mathrm{V}}|} \sum_{i \in \mathcal{N}_f^{\mathrm{V}}} (y_i - \hat{y}_i)^2} \tag{3.3}$$

$$\text{s.t.} \quad \hat{y}_i = \beta_0^f + \sum_{j \in \mathcal{P}} X_{i,j} \beta_j^f + \sum_{m \in \mathcal{M}} b_m^f Z_{i,m} \quad i \in \{\mathcal{N}_f^{\mathrm{Tr}}, \mathcal{N}_f^{\mathrm{V}}\}, f \in \mathcal{F} \tag{3.4}$$

$$\begin{bmatrix} \beta_0^f \\ \beta^f \\ b^f \end{bmatrix} = \left[\left(\tilde{X}^f \right)^\top \tilde{X}^f \right]^{-1} (\tilde{X}^f)^\top y \qquad f \in \mathcal{F} \tag{3.5}$$

$$\tilde{X}_{i,:}^f = [\mathbf{1}, X_{i,:}, Z_{i,:}] \qquad i \in \mathcal{N}_f^{\mathrm{Tr}}, f \in \mathcal{F} \tag{3.6}$$

$$Z_{i,m} = \sum_{k \in \mathcal{K}} \delta_{m,k} K_k(r_{i,m}(X_i, \alpha_m)) \qquad \forall i \in \mathcal{N}_f, m \in \mathcal{M} \tag{3.7}$$

$$\sum_{k \in \mathcal{K}} \delta_{m,k} = 1 \qquad \forall m \in \mathcal{M} \tag{3.8}$$

Model (3.3)-(3.8) cannot be solved exactly as a mathematical programming model due to its complex constraints, thus we designed two new algorithms in Step 2 to solve it heuristically, as illustrated in Figure 3.12. First, the elastic net regularization model[66] is applied to select a set of high-quality features for each fold and each category of soil, weather, and management features. Using the common features of all folds as a starting point, Algorithm 1 and Algorithm 2 are iteratively deployed to find a set of robust features and interactions. Algorithm 1 attempts to improve the robustness measure (3.3) using a stepwise linear regression approach [59, 29] in both backward and forward directions. Algorithm 2 detects interactions among the features identified by Algorithm 1. The interaction of these two algorithms was designed to maximize the robustness measure by balancing validation RMSE and training RMSE. These two algorithms iterate until the termination condition is met, when a set of features and interactions have been found that are temporally and spatially robust. Details of Algorithms 1 and 2 are explained as follows.

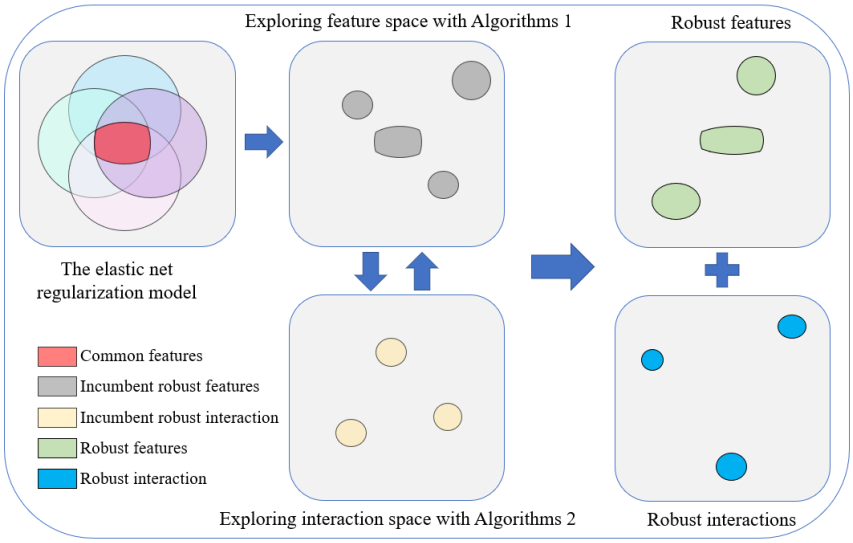

Figure 3.12 Diagram of step 2. First, the elastic net regularization model is used to select features from each of the folds and use their common features as a starting point for robust features. Then, Algorithm 1 tries to improve the robustness measure (3.3) using a stepwise linear regression approach in both backward and forward directions, and Algorithm 2 explores potentially significant interactions among these features. The final output of step 2 is a set of features and interactions that are temporally and spatially robust.

Algorithm 3 `Robust feature and interaction selection algorithm`

Input: Dataset $(X \in [\mathbb{B}, \mathbb{R}]^{|\mathcal{N}|\times|\mathcal{P}|}, y \in \mathbb{R}^{|\mathcal{N}|\times 1})$, high-quality features P_f^S, P_f^W, and P_f^M for each fold $f \in \mathcal{F}$.

Output: Robust feature set $\mathcal{P}^*$ and interactions $\alpha^* \in \{0, 0.5, 1\}^{|\mathcal{M}|\times|\mathcal{P}^*|}$ and their kernel function variable $\delta^* \in \mathbb{B}^{|\mathcal{M}|\times|\mathcal{K}|}$.

Initialize robust features set $\mathcal{P}^* = \cap_{f\in\mathcal{F}}(P_f^S \cup P_f^W \cup P_f^M)$ and empty sets j_a and j_r as candidate features for adding and removing, respectively.

Apply `Interaction`$(X_{\mathcal{N},\mathcal{P}^*}, y_{\mathcal{N}})$ algorithm to get $\alpha, \delta, \zeta_{\text{CV}}^{\text{Tr}}, \zeta_{\text{CV}}^{\text{V}}$. Initialize $\alpha^* = \alpha, \delta^* = \delta, \zeta_{\text{CV}}^{\text{Tr}*} = \zeta_{\text{CV}}^{\text{Tr}}, \zeta_{\text{CV}}^{\text{V}*} = \zeta_{\text{CV}}^{\text{V}}$.

for each $j \notin \mathcal{P}^*$ **do**

  Apply `Interaction`$(X_{\mathcal{N},\mathcal{P}^*\cup j}, y_{\mathcal{N}})$ algorithm to get $\alpha, \delta, \zeta_{\text{CV}}^{\text{Tr}}, \zeta_{\text{CV}}^{\text{V}}$.

  if $\zeta_{\text{CV}}^{\text{V}} < \zeta_{\text{CV}}^{\text{V}*}$ **then**

    Update $\alpha^* \leftarrow \alpha, \delta^* \leftarrow \delta, \zeta_{\text{CV}}^{\text{Tr}*} \leftarrow \zeta_{\text{CV}}^{\text{Tr}}, \zeta_{\text{CV}}^{\text{V}*} \leftarrow \zeta_{\text{CV}}^{\text{V}}$, and $j_a \leftarrow j$.

  end if

end for

for each $j \in \mathcal{P}^*$ **do**

  Apply `Interaction`$(X_{\mathcal{N},(\mathcal{P}^*\cup j_a)\setminus j}, y_{\mathcal{N}})$ algorithm to get $\alpha, \delta, \zeta_{\text{CV}}^{\text{Tr}}, \zeta_{\text{CV}}^{\text{V}}$.

  if $\zeta_{\text{CV}}^{\text{V}} < \zeta_{\text{CV}}^{\text{V}*}$ **then**

    Update $\alpha^* \leftarrow \alpha, \delta^* \leftarrow \delta, \zeta_{\text{CV}}^{\text{Tr}*} \leftarrow \zeta_{\text{CV}}^{\text{Tr}}, \zeta_{\text{CV}}^{\text{V}*} \leftarrow \zeta_{\text{CV}}^{\text{V}}$, and $j_r \leftarrow j$.

  end if

end for

if $\mathcal{P}^* = (\mathcal{P}^* \cup j_a) \setminus j_r$ **then**

  C(1): Finish.

else

  C(2): Update $\mathcal{P}^* \leftarrow (\mathcal{P}^* \cup j_a) \setminus j_r$; reset j_a and j_r sets as empty sets, and go to line 5.

end if

Algorithm 4 `Interaction algorithm`

1: **Input:** Dataset $(X \in [\mathbb{B}, \mathbb{R}]^{|\mathcal{N}|\times|\hat{\mathcal{P}}|}, y \in \mathbb{R}^{|\mathcal{N}|\times 1})$, training set $\mathcal{N}_f^{\mathrm{Tr}}$ and validation set $\mathcal{N}_f^{\mathrm{V}}$ for each fold $f \in \mathcal{F}$.

2: **Output:** $\alpha^* \in \{0, 0.5, 1\}^{M\times|\hat{\mathcal{P}}|}, \delta^* \in \mathbb{B}^{M\times|\mathcal{K}|}, \zeta_{\mathrm{CV}}^{\mathrm{Tr}}, \zeta_{\mathrm{CV}}^{\mathrm{V}}$ as local optimal interactions, their kernel function variable, CV training RMSE, and CV validation RMSE, respectively.

3: Initialize the incumbent solution $\alpha^* = 0.5^{1\times|\hat{\mathcal{P}}|}, \delta^* = 0^{1\times|\mathcal{K}|}$, and $M = 1$. Then, go to Step 1.

4: Initialize the current solution as $\hat{\alpha}_{i,j} = \begin{cases} \alpha^*_{m,j}, & \text{if } m \leq M-1 \\ 0.5, & \text{otherwise.} \end{cases}, \forall m \in \{1, ..., M\}, j \in \hat{\mathcal{P}}.$

5: Identify the 2-hop neighborhood of $\hat{\alpha}$ as follows.

$$\mathcal{A}(M) = \{\alpha \in \{0, 0.5, 1\}^{M\times|\hat{\mathcal{P}}|} : \|\alpha_{i,:} - \hat{\alpha}_{i,:}\|_0 \leq 2, \forall i \in \{1, ..., M\}; \sum_{j\in\hat{\mathcal{P}}} |\alpha_{i,j} - 0.5| \leq 1, \forall i \in \{1, ..., M\}\}$$

6: Evaluate $\zeta_{\mathrm{CV}}^{\mathrm{Tr}} = \zeta(X, y, \mathcal{N}_f^{\mathrm{Tr}}, \mathcal{N}_f^{\mathrm{Tr}}, \alpha, \delta)$ for all $\alpha \in \mathcal{A}(M)$. Let $\bar{\alpha}$ and $\bar{\delta}$ be optimal solutions: $\{\bar{\alpha}, \bar{\delta}\} = \underset{\alpha\in\mathcal{A}(M)}{\arg\min}\, \zeta(X, y, \mathcal{N}_f^{\mathrm{Tr}}, \mathcal{N}_f^{\mathrm{Tr}}, \alpha, \delta)$.

7: **if** $\zeta(X, y, \mathcal{N}_f^{\mathrm{Tr}}, \mathcal{N}_f^{\mathrm{V}}, \bar{\alpha}, \bar{\delta}) < \zeta(X, y, \mathcal{N}_f^{\mathrm{Tr}}, \mathcal{N}_f^{\mathrm{V}}, \alpha^*, \delta^*)$ **then**

8: **C(1):** Update $\alpha^* \leftarrow \bar{\alpha}$ and $\delta^* \leftarrow \bar{\delta}$; reset $M \leftarrow M + 1$, and go to line 4.

9: **else**

10: **C(2):** Finish.

11: **end if**

The objective of Algorithm 1 is to identify a set of robust features to minimize $\zeta_{\mathrm{CV}}^{\mathrm{Tr}}$, which is the average of training RMSEs. In line 3, the common features of different folds are used as the starting point for the set of robust features. In the two "for" loops, new features are added or removed using a stepwise linear regression approach to further improve the robustness measure $\zeta_{\mathrm{CV}}^{\mathrm{Tr}}$.

The objective of Algorithm 2 is to detect interactions among the features from Algorithm 1 to optimize the robustness measure $\zeta_{\mathrm{CV}}^{\mathrm{Tr}}$. It explores the 2-hop neighborhood of interactions space with all combinations of the kernel functions to optimize the robustness measure. This is an extended version of the algorithm in Ansarifar and Wang (2019) [6] by including not only discrete (genetic)

variables but also continues (environment and management) features. This is achieved by normalizing all continues variables to the [0, 1] internal and then using the six kernel functions to capture the potential nonlinear relationship between pairs of features.

3.8.3 Step 3: Linear Regression.

The interaction matrix Z augments the input dataset X with additional features, which helps fit the crop yield with a multiple linear regression model. As such, the model first deploys a powerful optimization engine to identify complex interactions, and then delivers explainable prediction results that can attribute crop yield to additive and interactive contributions of individual explanatory variables.

3.9 Appendix 2: Additional Results

Prediction performance of the proposed prediction algorithm for corn and soybean in three states over four test years is reported in Table 3.3.

All nine algorithms were deployed to predict both corn and soybean yields in the states of Illinois, Indiana, and Iowa from 2015 to 2018. To predict yield for the test year t, the training data included all the explanatory (weather, soil, and management) and response (crop yield) data from 1990 to year $t-1$. Cross validation was used to tune hyperparameters for all algorithms. Prediction errors for two crops over four test years using nine algorithms are summarized in Table 3.1. Prediction comparisons in terms of the relative RMSE (RRMSE), the relative squared error (RSE), the mean absolute error (MAE), the relative absolute error (RAE), and the coefficient of determination (R^2) of nine models are reported in Tables 3.4-3.8, respectively.

Table 3.3 RMSE in t/ha (and RRMSE in %) of the interaction regression model for corn and soybean in three states over four test years.

Crop	Dataset	Test Year			
		2015	2016	2017	2018
Corn	Train	0.61 (6.97%)	0.62 (6.99%)	0.62 (6.95%)	0.63 (6.99%)
	Validation	1.67 (15.01%)	1.01 (9.08%)	0.92 (8.30%)	0.89 (7.63%)
	Test (3 states)	1.02 (9.60%)	0.81 (7.06%)	0.90 (7.66%)	0.81 (6.73%)
	Test (Illinois)	0.99 (9.29%)	0.88 (7.72%)	0.99 (8.31%)	0.85 (6.80%)
	Test (Indiana)	1.37 (14.33%)	0.75 (7.18%)	0.88 (7.97%)	0.66 (5.68%)
	Test (Iowa)	0.59 (5.04%)	0.79 (6.29%)	0.82 (6.67%)	0.89 (7.45%)
Soybean	Train	0.21 (7.26%)	0.21 (7.26%)	0.21 (7.22%)	0.21 (7.16%)
	Validation	0.27 (8.06%)	0.30 (8.58%)	0.28 (7.75%)	0.26 (7.05%)
	Test (3 states)	0.29 (8.16%)	0.27 (7.18%)	0.23 (6.31%)	0.27 (6.97%)
	Test (Illinois)	0.30 (8.38%)	0.28 (7.30%)	0.20 (5.54%)	0.29 (6.94%)
	Test (Indiana)	0.30 (8.95%)	0.30 (8.18%)	0.22 (6.36%)	0.22 (5.94%)
	Test (Iowa)	0.27 (7.24%)	0.24 (6.10%)	0.26 (6.98%)	0.29 (7.83%)

Table 3.4 RRMSE (in %) of nine algorithms for corn and soybean yield prediction over four test years.

	Corn Test Year				Soybean Test Year			
Model	2015	2016	2017	2018	2015	2016	2017	2018
Linear Regression	13.05	11.57	10.13	8.01	14.51	12.53	11.50	10.88
Stepwise Regression	12.86	9.87	9.87	8.05	11.78	8.93	9.68	9.33
Lasso Regression	13.15	11.40	10.25	7.64	11.84	10.89	8.51	8.00
Ridge Regression	12.36	11.22	8.39	7.87	11.46	11.20	9.28	8.21
Elastic Net	11.71	10.96	8.77	7.69	11.32	10.41	8.82	8.57
Random Forest	12.16	10.45	9.03	7.81	9.64	9.66	7.64	10.03
XGBoost	14.06	11.93	10.58	8.99	12.05	11.94	10.89	11.14
Neural Network	11.56	7.12	8.05	7.74	11.28	9.73	8.58	10.10
Interaction Regression	**9.60**	**7.06**	**7.66**	**6.73**	**8.16**	**7.18**	**6.31**	**6.97**

Table 3.5 RSE of nine algorithms for corn and soybean yield prediction over four test years.

Model	Corn Test Year				Soybean Test Year			
	2015	2016	2017	2018	2015	2016	2017	2018
Linear Regression	0.86	0.61	0.62	0.49	1.72	1.60	1.10	0.76
Stepwise Regression	0.84	0.44	0.59	0.49	1.13	0.81	0.78	0.56
Lasso Regression	0.88	0.59	0.64	0.44	1.14	1.20	0.60	0.41
Ridge Regression	0.77	0.57	0.43	0.47	1.07	1.27	0.71	0.43
Elastic Net	0.69	0.54	0.46	0.45	1.04	1.10	0.65	0.47
Random Forest	0.75	0.50	0.49	0.46	0.76	0.95	0.48	0.64
XGBoost	1.00	0.65	0.68	0.61	1.18	1.45	0.99	0.79
Neural Network	0.68	0.23	0.39	0.45	1.04	0.96	0.61	0.65
Interaction Regression	**0.46**	**0.22**	**0.35**	**0.34**	**0.54**	**0.52**	**0.33**	**0.31**

Table 3.6 MAE (in t/ha) of nine algorithms for corn and soybean yield prediction over four test years.

Model	Corn Test Year				Soybean Test Year			
	2015	2016	2017	2018	2015	2016	2017	2018
Linear Regression	1.06	1.07	0.95	0.78	0.42	0.39	0.34	0.34
Stepwise Regression	1.11	0.91	0.94	0.79	0.34	0.28	0.28	0.29
Lasso Regression	1.08	1.06	0.96	0.74	0.34	0.33	0.24	0.25
Ridge Regression	1.05	1.06	0.76	0.77	0.32	0.35	0.27	0.25
Elastic Net	0.95	1.02	0.79	0.74	0.32	0.33	0.25	0.26
Random Forest	0.98	0.95	0.84	0.76	0.28	0.29	0.22	0.31
XGBoost	1.20	1.09	0.98	0.85	0.34	0.37	0.33	0.36
Neural Network	1.01	0.66	0.72	0.73	0.32	0.30	0.25	0.32
Interaction Regression	**0.74**	**0.66**	**0.69**	**0.65**	**0.23**	**0.22**	**0.17**	**0.22**

Table 3.7 RAE of nine algorithms for corn and soybean yield prediction over four test years.

Model	Corn Test Year				Soybean Test Year			
	2015	2016	2017	2018	2015	2016	2017	2018
Linear Regression	0.86	0.75	0.78	0.68	1.28	1.34	1.03	0.88
Stepwise Regression	0.90	0.64	0.77	0.70	1.05	0.96	0.87	0.74
Lasso Regression	0.87	0.75	0.79	0.65	1.05	1.15	0.75	0.66
Ridge Regression	0.85	0.74	0.62	0.68	0.98	1.21	0.81	0.64
Elastic Net	0.77	0.72	0.65	0.65	0.97	1.13	0.77	0.67
Random Forest	0.80	0.67	0.68	0.67	0.85	1.01	0.67	0.81
XGBoost	0.97	0.77	0.80	0.75	1.05	1.27	1.00	0.93
Neural Network	0.82	0.46	0.59	0.65	0.97	1.03	0.77	0.83
Interaction Regression	**0.60**	**0.46**	**0.56**	**0.58**	**0.71**	**0.76**	**0.53**	**0.56**

Table 3.8 R^2 of nine algorithms for corn and soybean yield prediction over four test years.

Model	Corn Test Year				Soybean Test Year			
	2015	2016	2017	2018	2015	2016	2017	2018
Linear Regression	0.13	0.38	0.37	0.50	-0.72	-0.60	-0.10	0.23
Stepwise Regression	0.15	0.55	0.40	0.50	-0.13	0.18	0.21	0.43
Lasso Regression	0.11	0.40	0.35	0.55	-0.14	-0.20	0.39	0.58
Ridge Regression	0.22	0.42	0.56	0.52	-0.07	-0.27	0.28	0.56
Elastic Net	0.30	0.45	0.53	0.54	-0.04	-0.10	0.34	0.52
Random Forest	0.24	0.49	0.50	0.53	0.23	0.04	0.51	0.35
XGBoost	-0.01	0.34	0.31	0.38	-0.18	-0.45	0.01	0.20
Neural Network	0.31	0.76	0.60	0.54	-0.04	0.03	0.38	0.34
Interaction Regression	**0.53**	**0.77**	**0.64**	**0.65**	**0.45**	**0.47**	**0.66**	**0.68**

Figure 3.13 provides complementary information to Figure 3.6 on the estimated contributions of weather and soil to crop yield.

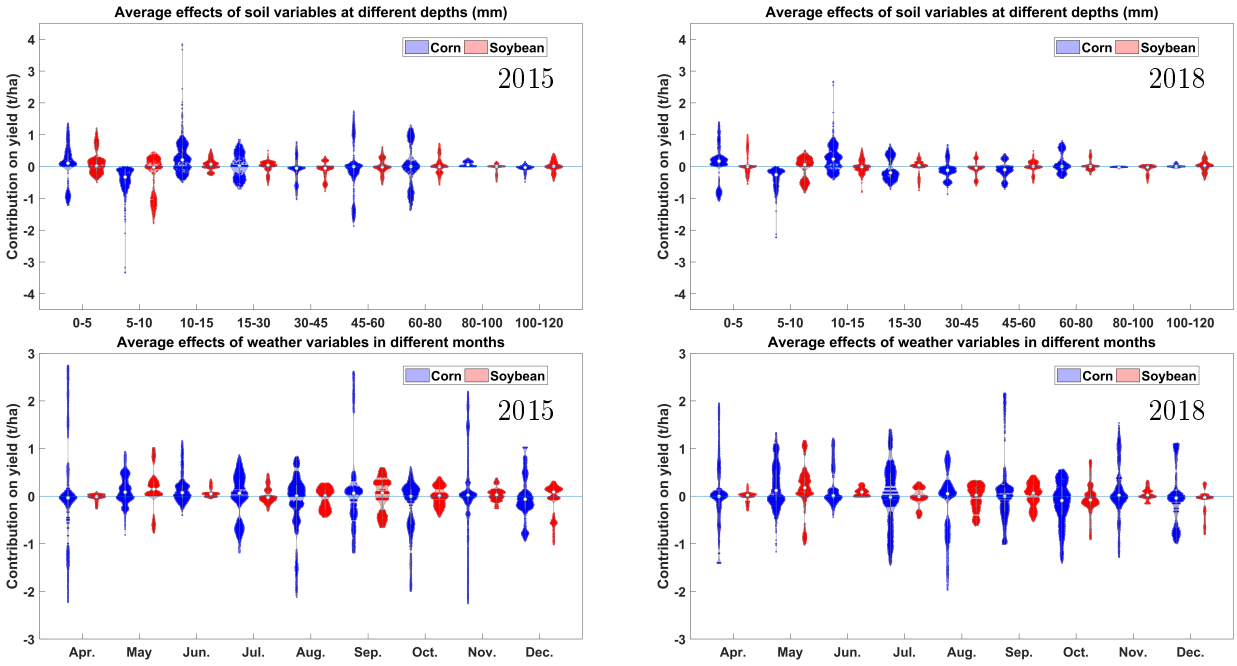

Figure 3.13 Violin plots of estimated contributions of soil variables (top) and weather variables (bottom) on corn and soybean yield in 2015 (left) and 2018 (right). Each dot on a violin plot represents a county level observation.

CHAPTER 4. PERFORMANCE PREDICTION OF CROSSES IN PLANT BREEDING THROUGH GENOTYPE BY ENVIRONMENT INTERACTIONS

A paper accepted by *Scientific Reports*

Javad Ansarifar, Faezeh Akhavizadegan, Lizhi Wang

4.1 Abstract

Performance prediction of potential crosses plays a significant role in plant breeding, which aims to produce new crop varieties that have higher yields, require fewer resources, and are more adaptable to the changing environments. In the 2020 Syngenta crop challenge, Syngenta challenged participants to predict the yield performance of a list of potential breeding crosses of inbreds and testers based on their historical yield data in different environments. They released a dataset that contained the observed yields for 294,128 corn hybrids through the crossing of 593 unique inbreds and 496 unique testers across multiple environments between 2016 and 2018. To address this challenge, we designed a new predictive approach that integrates random forest and an optimization model for G $\times$ E interaction detection. Our computational experiment found that our approach achieved a relative root-mean-square-error (RMSE) of 0.0869 for the validation data, outperforming other state-of-the-art models such as factorization machine and extreme gradient boosting tree. Our model was also able to detect genotype by environment interactions that are potentially biologically insightful. This model won the first place in the 2020 Syngenta crop challenge in analytics.

4.2 Introduction

Meeting the food demands of the world's growing population is one of the most significant challenges that society is facing, especially due to the continuously changing climate [18]. Vari-

ous approaches have been proposed to improve food production and security, including optimizing planting regime, sustainable farming practices, traits introgression, and modeling of plant physiology and ecology. In particular, optimizing the plant breeding process has been recognized as a promising area to improve global agrarian output with limited resources [30, 13, 24]. One of the most challenging decisions that plant breeders have to make is the selection of breeding parents for crosses [8]. For hybrid plant breeding, breeders make the best biparental crosses with high-yield potentials and test the hybrids' yield performance by planting them in multiple locations and weathers. The empirical breeding process of predicting, planting, and evaluating biparental combinations is expensive, labor-intensive, and time-consuming, which is why scientists are turning to artificial crosses to help the breeders predict and select promising breeding parents for hybridization. The 2020 Syngenta crop challenge was a recent effort by the agriculture industry to address such a challenge with realistic datasets. The goal of this challenge is to predict the yield performance of inbred-tester combinations in a given test set.

Many classic models have been used for prediction and selection of parents for crosses, including, clustering technique [33] as analysis of genetic diversity of hybrids, mixed models [8, 3, 4], best linear unbiased prediction (BLUP) [7, 26], ridge regression and the genomic best linear unbiased predictor (GBLUP) [34], and regression methods such as ridge [16, 17, 28] as predictor of cross performance of untested crosses, genetic relationship [5] as assessment of yield performance of hybrid combinations.

More recently, machine learning models have been applied to predict yield performances of crosses. For example, González-Camacho et al. [15] developed random forest, neural networks, and support vector machine (SVM) for predicting genomic performance. Montesinos-López et al. [25] applied SVM, neural network, and BLUP in the genomic selection process. A probabilistic neural network was applied for genome-based prediction of corn and wheat in González-Camacho et al. [14]. Basnet et al. [6] and Jiang et al. [20] developed G $\times$ E interactions models for grain yield prediction using the genomic general combining ability (GCA) and specific combining ability (SCA) and their interactions with environments. Acosta-Pech et al. [1] were the first to propose an extension of the models of Technow et al. [32] and Massman et al. [23] by combing

the G × E model with the reaction norm model proposed by Jarquín et al. [19]. They used an interaction-based model with the interactions between SCA and GCA effects and environment for genomic predictions. State-of-the-art machine learning models have also been used for crop yield prediction, including stepwise multiple linear regression [11], neural networks [11, 21], convolutional neural networks [31, 36], recurrent neural networks [36], multiple regression [21], random forest [27], weighted histograms regression [22], and association rule mining and decision tree [29].

In this paper, we propose a new model for predicting the yield performance of new hybrids based on historical data of other hybrids. This model integrates a random forest with a combinatorial optimization-based interaction-detection model and attempts to combine their strengths. The random forest model [9] is known for its capability to approximate general form nonlinear relationships among the variables. On the other hand, the interaction-detection model originated from a recently published algorithm [2] that has been shown to be particularly effective in detecting epistatic type of interactions. Our model extends that algorithm to the detection of genotype by environment interactions (G × E).

Our computational results using the 2020 Syngenta crop challenge data suggested that the proposed model can accurately predict the performance of untested cross combinations of inbreds and testers. Moreover, results of our prediction model can also reveal biologically meaningful insights, such as the best hybrids for specific environments.

4.3 Problem Definition

Most of the effort in a breeding program is related to evaluating inbreds by crossing to another inbred known as a tester. According to the problem statement of the 2020 Syngenta crop challenge, "it is a plant breeder's job to identify the best parent combinations by creating experimental hybrids and assessing the hybrids' performance by 'testing' it in multiple environments to identify the hybrids that perform best." While the yield performance of a hybrid is largely related to the parents, it is also affected by many factors that are hard to predict, such as heterosis and interactions between genotype and the environment.

The objective of the 2020 Syngenta crop challenge was to design a model for predicting the yield performance of a list of inbred-tester combinations based on historical datasets that included yield, genetic group, and pedigree information of hybrids collected in different environments over a number of years. If successful, this challenge will stimulate novel design of predictive models and algorithms for yield prediction of inbred-tester combinations and progeny testing of inbreds, which will help breeders make the most promising crosses without having to rely on large-scale trial-and-error that is expensive, labor intensive, and time consuming. The 2020 Syngenta crop challenge released the following dataset for commercial corn.

Training Dataset

- Yield: Historical yield performances were measured for 10,919 unique biparental hybrids. To provide realistic data without revealing proprietary information, actual yield values were anonymized to make the average and standard deviation of yields approximately 1.0 and 0.1, respectively. The range of the yields was from 0.0472 to 1.8001.

- Genetic clusters: No genetic marker information was available, but the genetic clusters of 593 unique inbreds and 496 unique testers were provided. Syngenta grouped the inbreds and testers into some clusters according to their genetic similarities using internal methods. There were 14 inbred clusters and 13 tester clusters.

- Environment: Out of a total of $593 \times 496 = 294{,}128$ possible combinations of inbred-tester crosses, the training data included 10,919 unique hybrids that were planted across 280 locations between 2016 and 2018, each year with a unique set of weather conditions. The information that we had for the environment is 280 location IDs and 3 years such that there were 599 unique location-weather combinations in the training set. The total number of unique hybrids-location-weather combinations was 155,765, some of which had multiple replications, so the total number of yield records was 199,476. However, this training dataset represents

only 0.08% of all possible $593 \times 496 \times 280 \times 3 = 247{,}067{,}520$ hybrids-location-weather combinations.

Test Dataset: The test dataset includes a set of inbred-tester combinations whose yield performances need to be predicted. The environments in which these hybrids would be grown were not specified in the crop challenge.

Evaluation Criteria: The evaluation criteria for the 2020 Syngenta crop challenge in analytics were "accuracy of the predicted values in the test set based on root mean squared error, simplicity and intuitiveness of the solution, clarity in the explanation, and the quality and clarity of the finalist's presentation at the 2020 INFORMS Conference on Business Analytics and Operations Research." Our model won the first place in this competition. For this paper, we evaluated the proposed model in terms of prediction accuracy. Because we did not have access to the ground truth yield of the test dataset, we divided the given dataset to training and validation subsets using 10-fold cross-validation (CV). Then, we used the average performance of the proposed model as the evaluation criteria.

4.4 Method

4.4.1 Data Preprocessing

We defined the input variable X as one-hot coding of hybrid-location-weather combinations and the output variable y as the corresponding yield. To accommodate this definition, four types of training data were converted to binary using the one-hot coding preprocessing: inbred and tester indices, genetic cluster, location ID, and weather. For those hybrid-location-weather combinations with multiple replications, the average yield was used as the output data. As such, the training data has a dimension of 155,765 observations by 1,399 (593 inbreds + 496 testers + 14 inbred clusters + 13 tester clusters + 280 locations + 3 years of weather) one-hot coding variables.

4.4.2 Proposed Model and Algorithm

We proposed a hybrid model for this challenge, which combines random forest with G × E interaction detection techniques. The overview of the model is diagramed in Figure 4.1. This model consists of three main components: a random forest model that captures the complex nonlinear relationship between input and output variables, a G × E interaction detection model that captures interactions among hybrid, location, and weather variables, and another random forest model that utilizes the interactions to augment the prediction performance of the first random forest model. Details of these components are described in the rest of this section.

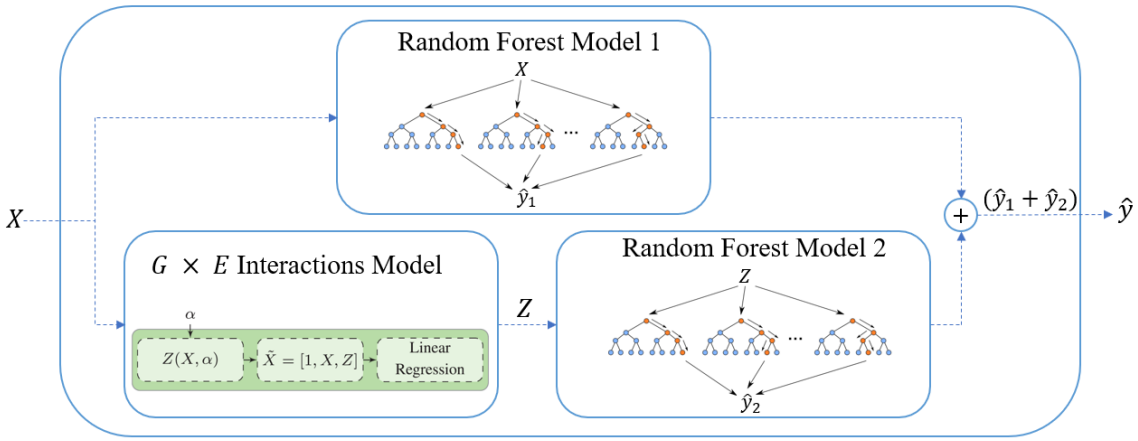

Figure 4.1 The test process of proposed model.

Random Forest Model 1

Random forest [9] is an ensemble learning model that can be used for classification or regression by constructing a multitude of decision trees. To grow each tree, a random subset of features is selected along with replacement sampling (bootstrap sampling) used to select different subsets of the observations. Therefore, observations in the dataset that were not included in the bootstrapped samples are considered as out-of-bag observations, and the performance of the tree is evaluated by the average out-of-bag error. Due to the builtin component of cross-validation, the random forest is less prone to overfitting.

The random forest model 1 takes the one-hot matrix X as input and predicts the corresponding yield performance $\hat{y}$ as output. This model is sensitive to three hyperparameters: the number of trees should be large enough to stabilize the error rate and small enough to be tractable; the number of features controls tree correlation, and the node size (minimum size of terminal nodes) determines the complexity of the individual trees. A 10-fold CV was used to partition dataset to training and validation subsets. For each fold, we used the training subset for training and parameter tuning. A 5-fold CV over train partition for each fold was applied to tune the parameters. Table 4.1 gives the values of these hyperparameters using a 5-fold CV over the whole dataset to get the best values that lead to good performance on the validation dataset.

Table 4.1 Tuned hyperparameters for the random forest model 1

Hyperparameters	Value
Number of trees	1000
Number of features	100
Node size	10

G $\times$ E Interactions Model

The random forest model has the capability to approximate nonlinear relationships among the variables. It grows many classification trees by randomly selecting subsets of features. As such, this model is ineffective in discovering specific combinations of features that have the most significant interactions. Therefore, we also introduced a combinatorial optimization-based model to augment the random forest by strategically searching for G $\times$ E interactions.

The G $\times$ E interactions model was designed to detect interactions among specific hybrid, location, and weather variables. This model is built off of a recently published algorithm [2], which was designed to detect genetic interactions in the form of epistases. The algorithm was found to be effective in detecting multiple interactions involving multiple variables. The G $\times$ E interactions model considers yield as a linear function of input variables and their interactions, shown as follows.

$$\hat{y}_i = \beta_0 + \sum_{j=1}^{p} X_{i,j}\beta_j + \sum_{k=1}^{K} b_k Z_{i,k} + \epsilon_i. \quad \forall i \in \{1, ..., n\} \tag{4.1}$$

Here,

- $X_{i,j} \in \{0, 1\}$ is the one-hot input variable j of observation i,
- $\hat{y}_i$ is the yield of observation i,
- $Z_{i,k} \in \{0, 1\}$ indicates whether or not observation i receives interaction k,
- β_0, β_j, and b_k are the effects of baseline, variable j, and interaction k, respectively, and
- ϵ_i is random noise for observation i.

In this model, the interactions are defined by a matrix α, which has a dimension of $K \times p$, where K is the number of interactions that the proposed model tries to decipher and p is the number of variables. Each column of this matrix corresponds to a variable and each row corresponds to an interaction. Moreover, each element of matrix α can take three possible values $0, 0.5, 1$. If $\alpha_{k,j} = 0$, then interaction k requires that variable j be 0 ($X_{i,j} = 0$) for any individual i to receive this effect. If $\alpha_{k,j} = 1$, then interaction k requires that variable j be 1 ($X_{i,j} = 1$) for any individual i to receive this effect. If $\alpha_{k,j} = 0.5$, then variable j is not involved in interaction k. Given matrix α, the matrix Z can be subsequently calculated to determine whether or not the individuals receive the interactions. The dimension of the binary matrix Z is $n \times K$, with each row corresponding to one individual and each column corresponding to one interaction. If $Z_{i,k} = 1$, then individual i receives the interaction k, and $Z_{i,k} = 0$ otherwise. This complex relationship can be captured mathematically as: individual i receives interaction k ($Z_{i,k} = 1$) if and only if $X_{i,j} + \alpha_{k,j} \neq 1$, or equivalently $X_{i,j} = \alpha_{k,j}$, for each variable j.

The key to model (4.1) is to find Z from a given training dataset $(X^{\text{Train}}, y^{\text{Train}})$, which requires the estimation of the number of interactions and the combination of variables that are involved in each interaction. When Z has been determined, model (4.1) reduces to a multiple linear regression that is easy to solve and interpret. Figure 4.2 illustrates an over-simplified example of G $\times$ E

interactions on corn yield. The given training data gives the yield of $n = 8$ corn plants with all possible combinations of $p = 3$ variables: high-yield (1) or low-yield (0) gene, fertile (1) or infertile (0) soil, wet (1) or dry (0) weather. No random noise was added to simplify the illustration. The figure shows the solution to the model (4.1). Matrix Z has three columns, indicating three interactions.

- The first interaction is triggered by infertile soil ($\alpha_{1,2} = 0$) and dry weather ($\alpha_{1,3} = 0$), reducing yield by 1 ($b_1 = -1$). Plants #3 and #4 receive this effect, indicated by the first column of matrix Z.

- The second interaction is triggered by high yield gene ($\alpha_{2,1} = 1$) and fertile soil ($\alpha_{2,2} = 1$), increasing yield by 1 ($b_2 = 1$). Plants #1 and #5 receive this effect, indicated by the second column of matrix Z.

- The third interaction is triggered by high yield gene ($\alpha_{3,1} = 1$) and wet weather ($\alpha_{3,3} = 1$), increasing yield by 2 ($b_3 = 2$). Plants #5 and #7 receive this effect, indicated by the third column of matrix Z.

The rest of the solution indicates that the baseline yield is $\beta_0 = 2$, the high yield gene, and wet weather contribute additional $\beta_1 = 1$ and $\beta_3 = 2$, respectively, and the fertile soil has no additive effect ($\beta_2 = 0$).

In our model, a similar approach is used to detect interactions among hybrid, soil, and weather at a much larger scale with $n = 155,765$ and $p = 1,399$. To overcome the computational challenges, we used a similar heuristic algorithm as in [2], which had three desirable features: (1) it used cross-validation to avoid-overfitting; (2) it was able to find local optimal solutions efficiently; and (3) it could be parameterized to balance computation time and solution quality.

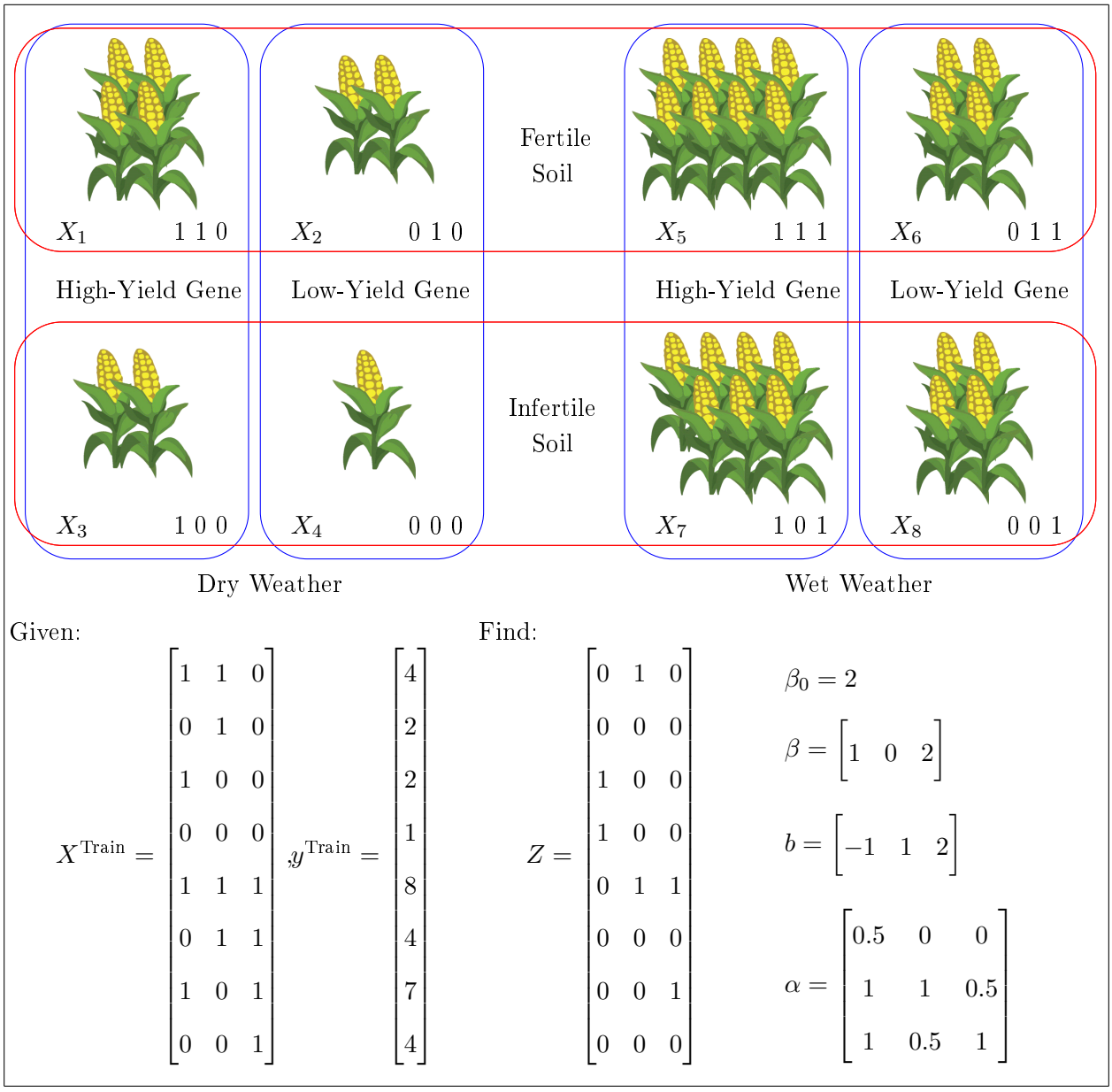

Figure 4.2 An illustrative example of G × E interactions.

Random Forest Model 2

Although the interaction model can decipher the interactions between binary predictors, it cannot find more complex nonlinear function of interactions. Hence, we feed the results of the G × E model into another random forest to identify more complex nonlinear interactions. Random forest model 2 was designed to predict the residual prediction from random forest model 1. Let $\hat{y}_1$

and $\hat{y}_2$ denote the predictions from random forest models 1 and 2. The overall model output $\hat{y}_1 + \hat{y}_2$ will provide a more accurate prediction of yield, y, than $\hat{y}_1$ if $\hat{y}_2$ can be trained to estimate $y - \hat{y}_1$.

To achieve this objective, we feed matrix Z from the G $\times$ E interactions model to random forest 2 to predict not only linear G $\times$ E interactions described in matrix Z but also more complex and nonlinear interactions. This model is trained using the residual of $y - \hat{y}_1$ to improve its accuracy. The tuned hyperparameters for the random forest model 2 are reported in Table 4.2. The same process as the random forest model 1 was applied to tune hyperparameters.

Table 4.2 Tuned hyperparameters for the random forest model 2

Hyperparameters	Value
Number of trees	1000
Number of features	20
Node size	10

The proposed model combines the strengths of combinatorial optimization in identifying G $\times$ E interactions and random forest in producing accurate predictions using complex and nonlinear functions. As such, it is a trade-off between insight and accuracy. It will be shown in the computational experiments that this hybrid model produced more insightful and accurate predictions than using either model alone.

4.5 Quantitative Results

In this section, we report the results of our computational experiments, which were designed to test the performance of the proposed algorithm with respect to other benchmark approaches.

4.5.1 Prediction Accuracy

To show the performance of the proposed model, it was compared with models from the literature, which are summarized as follows:

- A multiple linear regression model was trained using the glmnet [12] package in R statistical software (version 3.4.4).
- The multi-way interacting regression via factorization machines (MiFM) [37] was implemented in Python by the authors.
- An extreme gradient boosting tree (XGBoost) [10] model was trained using the xgboost [10] package in R, which was an efficient and scalable implementation of gradient boosting framework. Three hyperparameters were tuned using 5-fold cross validation (without data leakage): "nrounds", "eta", and "gamma".
- A G × E interactions model [2] was implemented in MATLAB (Version 2018a), which used heuristic algorithms to detect multi-way and multi-effect epistasis (interactions between binary variables). It is equivalent to the G × E interactions model without integrating with the random forest models.
- A random forest [9] was trained using the ranger [35] packages in R, which was an ensemble of decision trees and trains with the bagging method, equivalent to the random forest model 1 without the interaction model and the random forest model 2 in our proposed model. Three hyperparameters were tuned using 5-fold cross-validation: the number of trees, number of features, and node size.
- The proposed model was implemented in MATLAB (Version 2018a).

Three metrics were used for evaluating and comparing the predictive models' performances: RMSE, which presents the difference between predicted and observed values, Mean Absolute Error (MAE), which measures the average magnitude of the prediction errors, without considering their direction, and R^2, the coefficient of determination defined as the proportion of the variance in the response variable that is explained by independent variables. Because the ground truth of the test dataset was never released, we partitioned the training dataset into training and validation subsets in a 10-fold CV manner. For each fold, we tuned the parameters and trained the models using

the training set, and then their performances were evaluated using the validation set. We made sure that no validation data was leaked in the model training process. The average RMSE, MAE, and R^2 values over 10 partitions for the six algorithms are reported in Table 4.3. These results indicate that the proposed model outperformed other algorithms in all measures. Since the random forest model was part of our proposed model and it outperformed the first four machine learning algorithms, these results indicated the effectiveness of both the random forest method and our G × E interactions detection model.

Table 4.3 Average RMSE, MAE, and R^2 of six algorithms for yield prediction. A 10-fold cross-validation on the training dataset was used for algorithm performance evaluation, since the ground truth yield of the test dataset was never released.

Model	**Train**			**Validation**		
	RMSE	**MAE**	R^2	**RMSE**	**MAE**	R^2
Linear regression	0.1016	0.1009	0.1047	0.1026	0.0851	0.0866
Factorization machine	0.0740	0.0676	0.4855	0.0984	0.0765	0.1578
Xgboost	0.0790	0.0735	0.4581	0.0996	0.0806	0.1388
G × E	0.0740	0.0706	0.4902	0.0980	0.0744	0.1623
Random forest	0.0737	0.0673	0.5283	0.0976	0.0723	0.1738
Proposed model	0.0548	0.0523	0.7386	0.0869	0.0648	0.3448

The performance of the proposed model is also illustrated in figure 4.3, which plots the average predicted yields against actual observations for all inbreds and testers. The results suggest that our proposed model's prediction is close to the observation, both on average and in terms of probability density distributions.

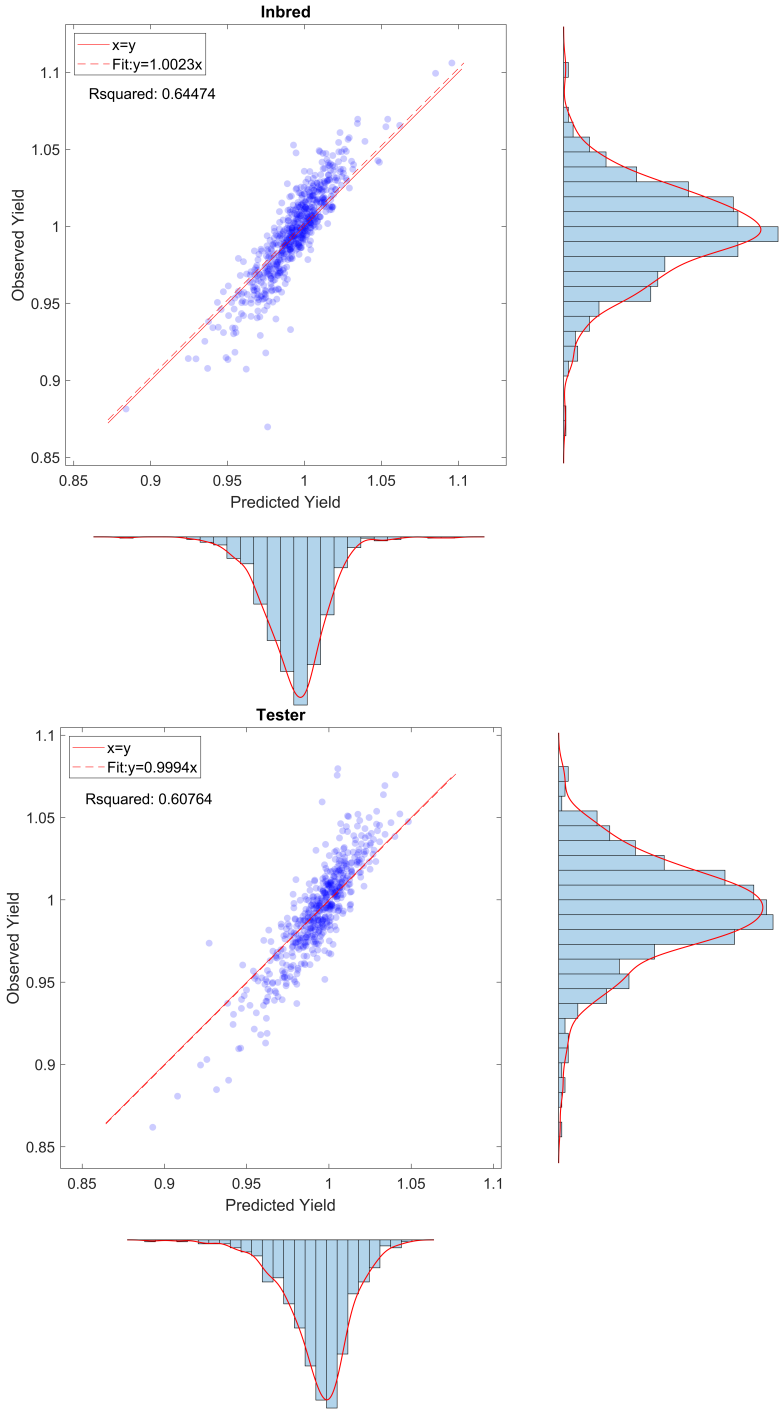

Figure 4.3 The up and down plots indicate the plots of the average observed yield versus the average predicted yield for performances of inbreds and testers, respectively.

We also examined the consistency of top and bottom inbreds and testers selected based on our prediction model against those based on observations. Out of the top 29 (5%) inbreds among all 593 inbreds with the highest average yield selected by our model, 21 of them were consistent with those selected based on actual observations. Similarly, out of the top 24 (5%) testers among all 496 testers with the highest average yield selected by our model, 17 of them were consistent with those selected based on actual observations. The counterpart consistency ratios for the bottom 5% inbreds and bottom 5% testers are $\frac{22}{29}$ and $\frac{16}{24}$, respectively. The predicted and observed average yield for the 14 inbred clusters and 13 tester clusters are summarized in Table 4.4.

Table 4.4 Predicted and observed average yield of 14 inbred clusters and 13 tester clusters.

Average	**Inbred cluster**													
Yield	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Predicted	1.010	1.010	1.011	0.997	1.007	0.991	1.002	0.993	0.997	0.990	0.986	0.991	0.992	0.998
Observed	1.006	1.020	1.007	0.992	1.003	0.981	0.999	0.988	0.990	0.991	0.984	0.992	0.996	0.996
Average	**Tester cluster**													
Yield	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	
Predicted	0.999	1.002	0.999	0.992	1.004	0.993	1.005	1.005	0.998	0.981	0.999	0.992	1.001	
Observed	0.995	0.996	0.994	0.992	1.003	0.997	1.001	1.001	0.998	0.980	1.005	0.975	0.996	

4.5.2 Genotype and Environment Interactions

The proposed model was able to provide not only accurate yield prediction but also genotype and environment interactions that could be biologically insightful. Figures 4.4 and 4.5 show the two-way and three-way interactions between variables, respectively. The results indicate that weather variables involve in more interactions following soil and genotype.

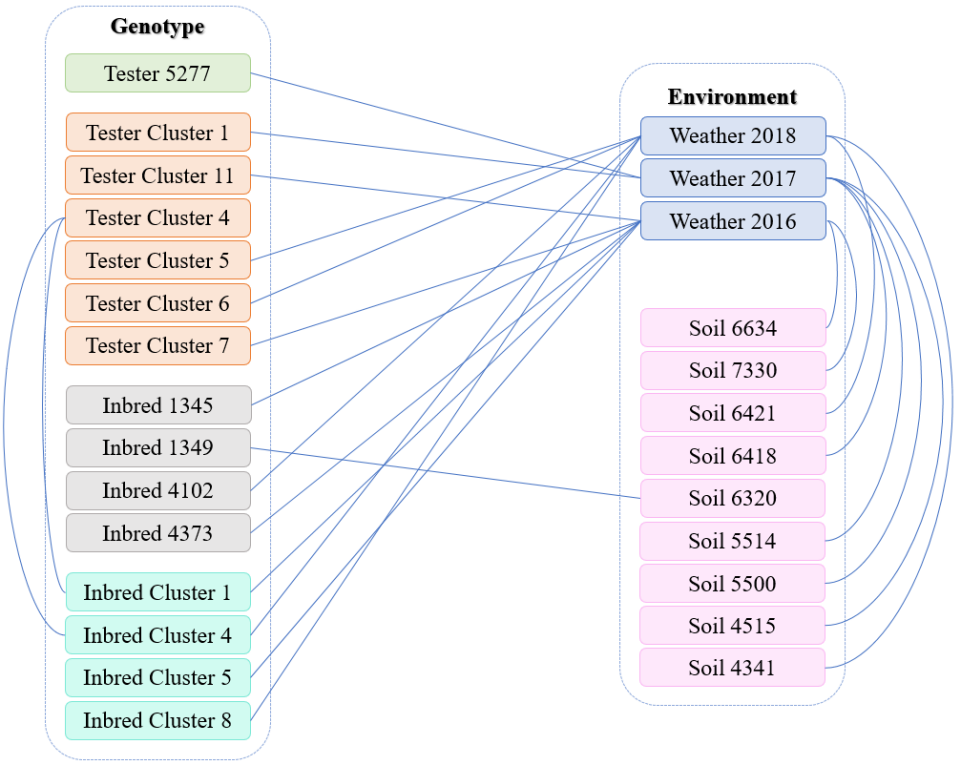

Figure 4.4 Two-way interactions. Each line shows the two-way interaction between two variables.

4.5.3 Optimal Biparental Crosses

To shed light on optimal biparental crosses between the given inbreds and testers, we used the proposed model to predict the yield performance of all combinations of testers and inbreds in different years and locations. Then, we ranked them based on average yield performance over all years and locations. The results of the top and bottom 5% of inbred-tester combinations (combinations of top and bottom 29 inbreds with top and bottom 24 testers) are illustrated in Figure 4.6, which can help breeders predict the most promising crosses. The average yields for four combinations of crosses are given in Table 4.5. These results appear to suggest that testers have a slightly higher weight in determining the yield performance of their progeny.

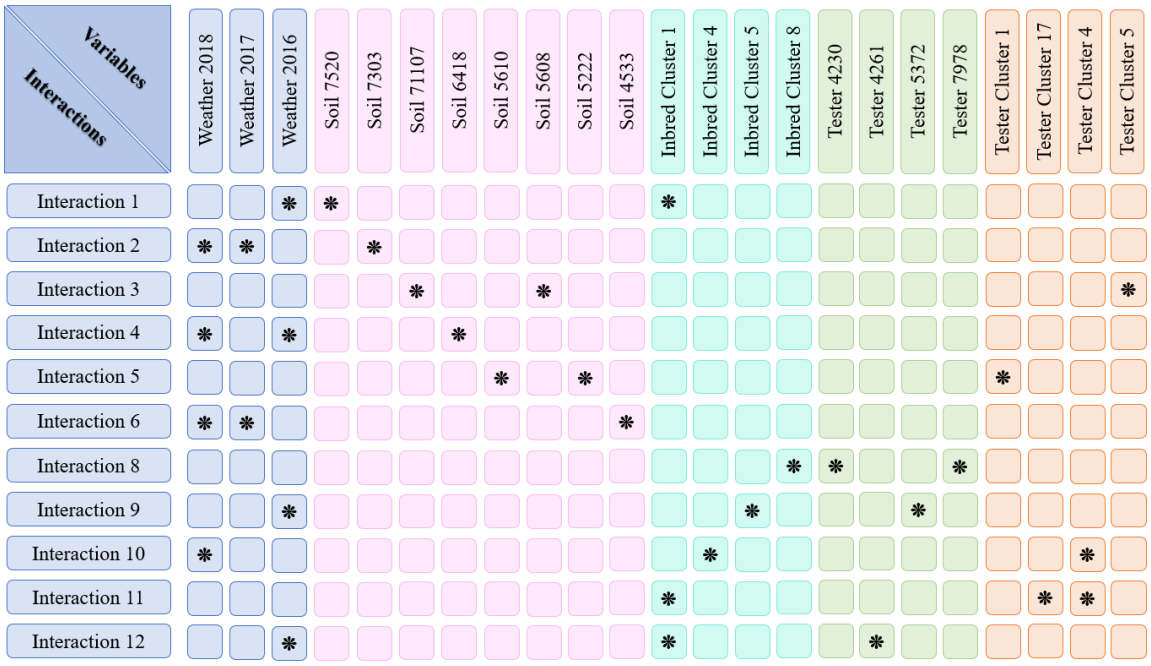

Figure 4.5 Three-way interactions. Each row indicates the three-way interaction between three variables. The star markers in each row indicate which variables involve in the interaction.

Table 4.5 Average yield performance of combinations of high- and low-yield testers and inbreds.

	High-yield Tester	Low-yield Tester
Low-yield Inbred	1.0098	0.9457
High-yield Inbred	1.0625	0.9789

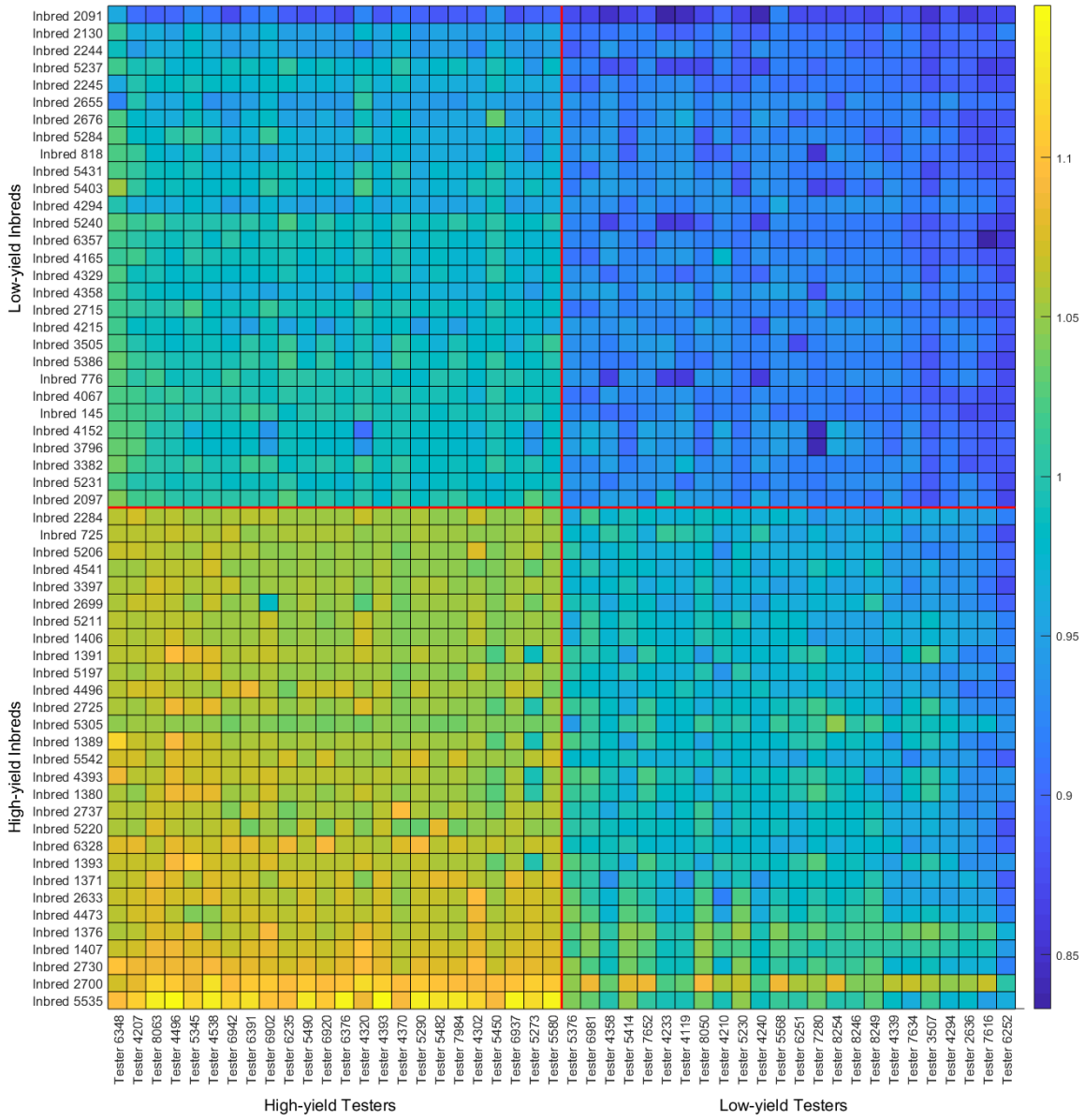

Figure 4.6 Predicted yield performances for combinations of the top and bottom 5% of inbreds and testers.

4.6 Conclusion

We proposed a new model to address the 2020 Syngenta crop challenge, which combines random forest with an G $\times$ E interactions model to predict yield performance of inbreds and testers based

on historical yield data in multiple years and environments. Random forest model has been found to be an effective and powerful machine learning model for prediction, yet it has its limitations in the degrees and types of interactions among the predictors. Based on a recently published algorithm for detecting multi-way and multi-effect epistatic effects, the G × E interactions model captures both linear and nonlinear interactions of the genotype by environment effects. The combination of random forest and the G × E interactions model was found to be effective in predicting yield performances of inbred-tester combinations in our computational study using 10-fold validation, achieving a 0.0869 validation RMSE, 0.0648 validation MAE, and 0.3448 R-squared value, outperforming four other popular machine learning algorithms as the benchmark. Moreover, our proposed model was also more explainable than other machine learning models by yielding genotype by environment interactions. Results from our proposed model will be able to help breeders test progeny and identify the best parent combinations to produce new hybrids with improved yield performances.

4.7 References

[1] Acosta-Pech, R., Crossa, J., de los Campos, G., Teyssèdre, S., Claustres, B., Pérez-Elizalde, S., and Pérez-Rodríguez, P. (2017). Genomic models with genotype× environment interaction for predicting hybrid performance: an application in maize hybrids. *Theoretical and Applied Genetics*, 130(7):1431–1440.

[2] Ansarifar, J. and Wang, L. (2019). New algorithms for detecting multi-effect and multi-way epistatic interactions. *Bioinformatics*, 35(24):5078–5085.

[3] Balzarini, M. (2002). 23 applications of mixed models in plant breeding. *Quantitative Genetics, Genomics, and Plant Breeding*, page 353.

[4] Balzarini, M. G. (2000). Biometrical models for predicting future performance in plant breeding. *Ph.D. Dissertation., Louisiana State University,Baton Rouge, LA, USA.*

[5] Barbosa-Neto, J., Sorrells, M., and Cisar, G. (1996). Prediction of heterosis in wheat using coefficient of parentage and rflp-based estimates of genetic relationship. *Genome*, 39(6):1142–1149.

[6] Basnet, B. R., Crossa, J., Dreisigacker, S., Pérez-Rodríguez, P., Manes, Y., Singh, R. P., Rosyara, U. R., Camarillo-Castillo, F., and Murua, M. (2019). Hybrid wheat prediction using genomic, pedigree, and environmental covariables interaction models. *The Plant Genome*, 12(1).

[7] Bernardo, R. (1996). Best linear unbiased prediction of maize single-cross performance. *Crop Science*, 36(1):50–56.

[8] Bertan, I., Carvalho, F., and Oliveira, A. d. (2007). Parental selection strategies in plant breeding programs. *Journal of Crop Science and Biotechnology*, 10(4):211–222.

[9] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[10] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd Acm Sigkdd International Conference on knowledge Discovery and Data Mining*, pages 785–794. ACM.

[11] Drummond, S. T., Sudduth, K. A., Joshi, A., Birrell, S. J., and Kitchen, N. R. (2003). Statistical and neural methods for site–specific yield prediction. *Transactions of the ASAE*, 46(1):5.

[12] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1.

[13] Godfray, H. C. J., Beddington, J. R., Crute, I. R., Haddad, L., Lawrence, D., Muir, J. F., Pretty, J., Robinson, S., Thomas, S. M., and Toulmin, C. (2010). Food security: The challenge of feeding 9 billion people. *Science*, 327(5967):812–818.

[14] González-Camacho, J. M., Crossa, J., Pérez-Rodríguez, P., Ornella, L., and Gianola, D. (2016). Genome-enabled prediction using probabilistic neural network classifiers. *BMC Genomics*, 17(1):208.

[15] González-Camacho, J. M., Ornella, L., Pérez-Rodríguez, P., Gianola, D., Dreisigacker, S., and Crossa, J. (2018). Applications of machine learning methods to genomic selection in breeding wheat for rust resistance. *The Plant Genome*, 11(2).

[16] Hoerl, A. E. and Kennard, R. W. (2000). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86.

[17] Hofheinz, N., Borchardt, D., Weissleder, K., and Frisch, M. (2012). Genome-based prediction of test cross performance in two subsequent breeding cycles. *Theoretical and Applied Genetics*, 125(8):1639–1645.

[18] Huai, J. (2017). Dynamics of resilience of wheat to drought in australia from 1991–2010. *Scientific Reports*, 7(1):9532.

[19] Jarquín, D., Crossa, J., Lacaze, X., Du Cheyron, P., Daucourt, J., Lorgeou, J., Piraux, F., Guerreiro, L., Pérez, P., Calus, M., et al. (2014). A reaction norm model for genomic selection using high-dimensional genomic and environmental data. *Theoretical and applied genetics*, 127(3):595–607.

[20] Jarquin, D., Howard, R., Liang, Z., Gupta, S. K., Schnable, J. C., and Crossa, J. (2019). Enhancing hybrid prediction in pearl millet using genomic and/or multi-environment phenotypic information of inbreds. *Frontiers in Genetics*, 10.

[21] Kaul, M., Hill, R. L., and Walthall, C. (2005). Artificial neural networks for corn and soybean yield prediction. *Agricultural Systems*, 85(1):1–18.

[22] Marko, O., Brdar, S., Panic, M., Lugonja, P., and Crnojevic, V. (2016). Soybean varieties portfolio optimisation based on yield prediction. *Computers and Electronics in Agriculture*, 127:467–474.

[23] Massman, J. M., Gordillo, A., Lorenzana, R. E., and Bernardo, R. (2013). Genomewide predictions from maize single-cross data. *Theoretical and applied genetics*, 126(1):13–22.

[24] McCouch, S., Baute, G. J., Bradeen, J., Bramel, P., Bretting, P. K., Buckler, E., Burke, J. M., Charest, D., Cloutier, S., Cole, G., et al. (2013). Agriculture: feeding the future. *Nature*, 499(7456):23.

[25] Montesinos-López, O. A., Martín-Vallejo, J., Crossa, J., Gianola, D., Hernández-Suárez, C. M., Montesinos-López, A., Juliana, P., and Singh, R. (2019). A benchmarking between deep learning, support vector machine and bayesian threshold best linear unbiased prediction for predicting ordinal traits in plant breeding. *G3: Genes, Genomes, Genetics*, 9(2):601–618.

[26] Panter, D. and Allen, F. (1995). Using best linear unbiased predictions to enhance breeding for yield in soybean: I. choosing parents. *Crop Science*, 35(2):397–405.

[27] Parmley, K. A., Higgins, R. H., Ganapathysubramanian, B., Sarkar, S., and Singh, A. K. (2019). Machine learning approach for prescriptive plant breeding. *Scientific Reports*, 9(1):1–12.

[28] Piepho, H.-P. (2009). Ridge regression and extensions for genomewide selection in maize. *Crop Science*, 49(4):1165–1176.

[29] Romero, J. R., Roncallo, P. F., Akkiraju, P. C., Ponzoni, I., Echenique, V. C., and Carballido, J. A. (2013). Using classification algorithms for predicting durum wheat yield in the province of buenos aires. *Computers and Electronics in Agriculture*, 96:173–179.

[30] Rosegrant, M. W. and Cline, S. A. (2003). Global food security: challenges and policies. *Science*, 302(5652):1917–1919.

[31] Russello, H. (2018). Convolutional neural networks for crop yield prediction using satellite images. *IBM Center for Advanced Studies*.

[32] Technow, F., Riedelsheimer, C., Schrag, T. A., and Melchinger, A. E. (2012). Genomic prediction of hybrid performance in maize with models incorporating dominance and population specific marker effects. *Theoretical and Applied Genetics*, 125(6):1181–1194.

[33] Van Beuningen, L. and Busch, R. (1997). Genetic diversity among north american spring wheat cultivars: III. cluster analysis based on quantitative morphological traits. *Crop Science*, 37(3):981–988.

[34] VanRaden, P. M. (2008). Efficient methods to compute genomic predictions. *Journal of dairy science*, 91(11):4414–4423.

[35] Wright, M. N. and Ziegler, A. (2015). ranger: A fast implementation of random forests for high dimensional data in c++ and r. *ArXiv Preprint ArXiv:1508.04409*.

[36] You, J., Li, X., Low, M., Lobell, D., and Ermon, S. (2017). Deep gaussian process for crop yield prediction based on remote sensing data. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[37] Yurochkin, M., Nguyen, X., et al. (2017). Multi-way interacting regression via factorization machines. In *Advances in Neural Information Processing Systems*, pages 2598–2606.

CHAPTER 5. SCHEDULING PLANTING TIME THROUGH DEVELOPING AN OPTIMIZATION MODEL AND ANALYSIS OF TIME SERIES GROWING DEGREE UNITS

A paper submitted by *Scientific Reports*

Javad Ansarifar, Faezeh Akhavizadegan, Lizhi Wang

5.1 Abstract

Producing higher-quality crops within shortened breeding cycles ensures global food availability and security, but this improvement intensifies logistical and productivity challenges for seed industries in the year-round breeding process due to the storage limitations. In the 2021 Syngenta crop challenge in analytics, Syngenta raised the problem to design an optimization model for the planting time scheduling in the 2020 year-round breeding process so that there is a consistent harvest quantity each week. They released a dataset that contained 2569 seed populations with their planting windows, required growing degree units for harvesting, and their harvest quantities at two sites. To address this challenge, we developed a new framework that consists of a weather time series model and an optimization model to schedule the planting time. A deep recurrent neural network was designed to predict the weather into the future, and a Gaussian process model on top of the time-series model was developed to model the uncertainty of forecasted weather. The proposed optimization models also scheduled the seed population's planting time at the fewest number of weeks with a more consistent weekly harvest quantity. Using the proposed optimization models can decrease the required capacity by 69% at site 0 and up to 51% at site 1 compared to the original planting time.

5.2 Introduction

Global food availability and sustainability are two of the most fundamental challenges due to the growing population and running out of agricultural land required to produce food for people and livestock [20, 17]. Additional challenges include increasingly variable growing conditions and climate change [50, 47]. Data-driven strategies increase and improve the productivity and sustainability of agriculture by proposing appropriate and adaptive management practices (e.g. planting, irrigation, fertilizing, tilling, harvesting, and management), scheduling the activities in the agriculture field, and breeding plants with the highest-yielding genetics [52, 50]. Although applying new methods and analytical approaches helps seed industries to produce higher-quality crops within shortened breeding cycles, ultimately ensuring required food for global food security, it comes with an unprecedented new set of challenges.

Recently, this improvement intensifies logistical and productivity issues for seed industries in the year-round breeding process due to the storage capacity limitations and erratic and inconsistent weekly harvest quantities. One of the most important decisions in management practices is scheduling planting time, which has significant implications in field crops' development and productivity, crop model applications, and in acquiring adaptation strategies for future climate change. Although implementing an optimal planting time reduces the negative impact on the environment and maximizes crop yield [4, 32, 43, 49], it hinders seed industries by increasing the storage requirement and logistic risk incurred during the seed production in the year-round breeding process [11].

Seed industries use analysis of a suite of management practices to identify the optimum schedule among the possibilities for management practices, including planting, irrigation, fertilizing, tilling, and harvesting. However, the complexities among management practices decision, resources, availability of seed, uncertain environment, and policies and procedures have led to the necessity of proposing a decision-making framework for management practices that consider logistic and storage limitation, seed production process, and environmental uncertainty. The 2021 Syngenta crop challenge in analytics was launched to find a critical decision in sustainable agriculture to optimally

schedule the planting of seeds to ensure that when ears are harvested, facilities are not over capacity and that there is a consistent number of ears each week.

Estimation of crop planting time has received special attention among industry players, researchers, and academic actors. In-depth reviews of proposed approaches in crop planning problems have been published by Lowe and Preckel [31] and Ahumada and Villalobos [2]. Several methods have been proposed and reported in the literature for addressing crop planting problems because of the complex and nonlinear relationships between planting time and profitability of agricultural products and uncertainties of the environment. The majority of studies in the literature review have been conducted research on the scheduling of crop planting time in farmers' fields during the summer growing season. In contrast to their work, we schedule planting time of different seed populations to produce seed for farmers in the year-round breeding at a seed industry level.

Three planting time scheduling methods have been proposed in the literature, including predefined and constant planting time, mathematical programming models, and statistical analysis methods [54]. In the first approach, based on long term observations, the constant and fixed planting time is derived as representing typical average planting time [16, 9, 13, 15, 43]. The second approach is the application of mathematical programming by adjusting the farming system with different management practices to optimize the planting time scheduling with limited resources. Mathematical programming methods include a linear programming model [38], genetic algorithm for a weighted sum method [56], simple heuristic allocation policy [7], heuristic selection algorithm using automatic fuzzy clustering [18], a strict mathematical framework using fuzzy set theory [44], calibrated crop model using a genetic algorithm [57], and integration of demand fuzzy time series modeling and linear programming methods [23, 53]. The planting scheduling algorithm was developed to optimize planting time based on the nearest distance to customers and the availability of greenhouses and open fields [39]. Closest to our model, Li et al. [30] developed a fuzzy-based linear multi-objective programming model under uncertainty for crop planting structure planning. The third approach is statistical analysis methods that determine the best planting time by measuring the yield and other objective function response to planting time. They include a segmented-

linear regression model [14], analysis of variance [5, 46], rule-based methods [12, 34], STICS model [8], DSSAT model [25], CERES-Maize model [51], calibrated RZWQM2 model [3], APSIM model [6, 26, 22], progression model and simulation analysis [58], nonlinear model [27], and the greatest likelihood of planting based on cumulative heat units [36].

The majority of studies have attempted to shed light on the planting time scheduling in farmers' fields during the growing season using different sets of tools and methods. However, seed companies need to know the planting time of seeds beyond the conventional growing season (for year-round breeding) to keep up with genetic improvement in the production cycle. Based on this literature gap, we focus on optimally scheduling the planting time of different seed populations in the year-round breeding process so that there is a consistent number of harvested ears each week. First, to address weather information uncertainty during the year-round breeding process, the weather must be predicted based on time-series analysis of historical weather information. Then, we developed the optimization model to schedule the planting of seed population within a few harvest weeks with a more consistent weekly harvest quantity. To show our model's performance, the 2021 Syngenta crop challenge data was used for our computational results in different cases.

5.3 Problem definition

The year-round breeding process of commercial corn as one of the world's most significant and planted crops is illustrated in Figure 5.1. When seed populations arrived, they must wait for planting until their scheduled planting date is reached. After planting, they have to be mature enough to harvest. Growing degree units (GDUs) are a heuristic measurement in phenology that gardeners and farmers use to predict the crop development stages (e.g. emergence stage and maturity stage) by reaching the accumulated GDUs to a certain amount [33, 55]. For several crops in different regions, a relationship between crop development stages and accumulated GDUs has been conducted [10, 40, 45]. GDUs are computed by taking the integral of warmth above a base temperature, approximately the average of the daily minimum and maximum temperatures.

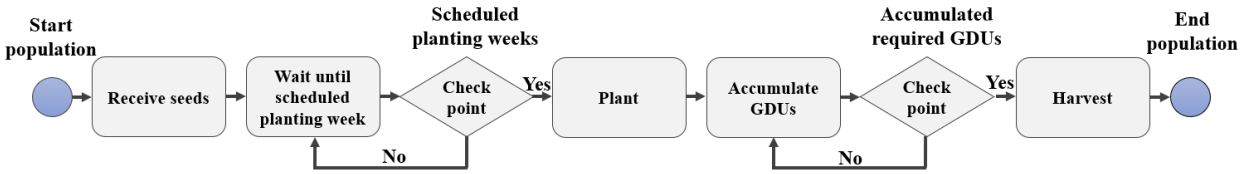

Figure 5.1 The year-round breeding process

The goal of this paper is to develop the model to optimize the planting time of seed populations (see Figure 5.2) to address logistical and productivity challenges because of capacity limitations and inconsistent weekly harvest quantities.

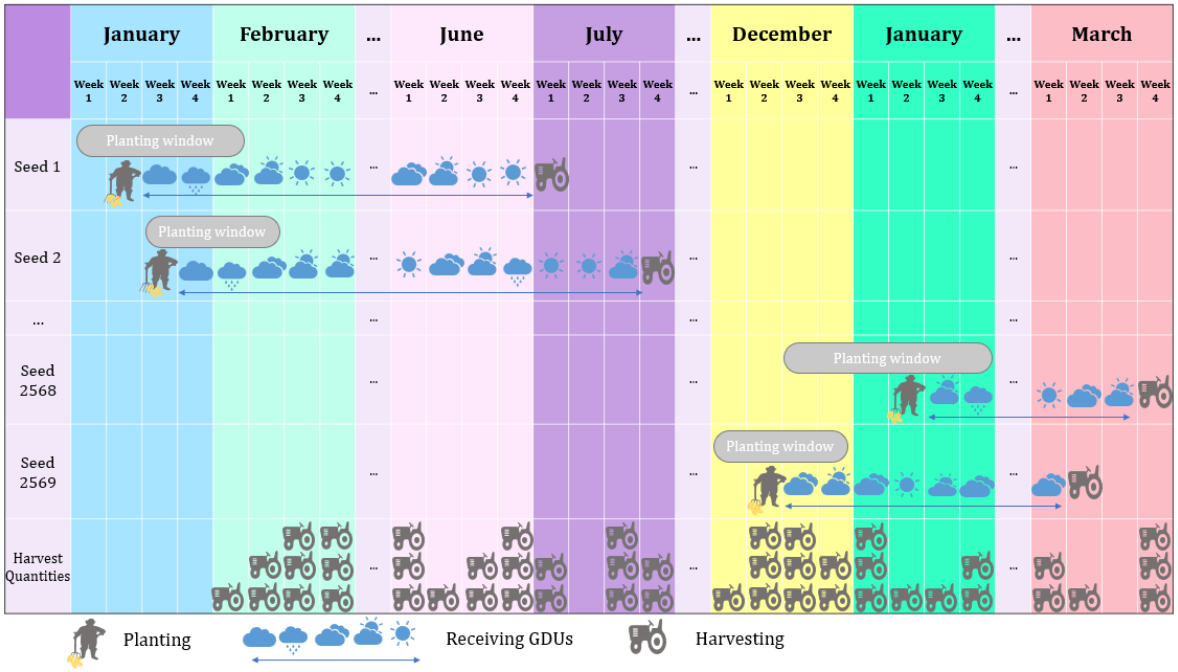

Figure 5.2 The year-round breeding process

5.3.1 Data

The 2021 Syngenta crop challenge provided the Data, which includes seed populations, planting windows, required GDUs, harvest quantity, and historical GDUs information to optimize commercial corn's year-round breeding process.

Calendar information: This challenge is to schedule the seed populations' planting times during 2020. Week index is starting from the first week of January 2020. Each week runs from Sunday – Saturday.

Seed population information: This dataset includes 2569 seed populations, the planting site of each seed population, planting windows, the required number of GDUs in Celsius needed for the harvest, and the harvest quantity of each seed population. There are two different cases with specific population's harvest quantity distribution. Syngenta simulated harvest quantities based on normal distributions for this challenge so that cases 1 and 2 follow normal distributions N(250,100) and N(350,150), respectively. However, in the real world, to estimate harvest quantities, we have to use predictive models based on historical information. Moreover, case 1 has the capacities, while there is no capacity limitation in case 2, and we are looking to determine the lowest possible capacity required. There are two different sites with different capacities. Site 0 has a capacity of 7,500 ears, and site 1 has a capacity of 6,000 ears at each week in case 1.

Historical weather information: This dataset includes historical GDUs in Celsius accumulated for each day for both sites during the last 11 years (2009-2019). Time series techniques can be used with this information to predict the GDUs in 2020.

5.3.2 Objective function

The objective of case 1 for the 2021 Syngenta crop challenge in analytics was to optimize each seed population's planting time at the fewest number of weeks so that the seed industry has a consistent weekly harvest quantity and capacity limitation constraint is met at each week. The objective of case 2 was to optimize each seed population's planting time so that the seed industry has a consistent weekly harvest quantity at the lowest possible capacity.

5.4 Method

We developed a hybrid framework for this challenge, which combined the time series prediction model and optimization model to schedule the planting time of seed population in a one-year breeding process. The overview of this framework is diagrammed in Figure 5.3. This model includes two components: a weather prediction model that forecasts time series of GDUs for 2020 from historical GDUs information and an optimization model that finds optimal scheduling for planting seed populations to ensure a consistent weekly harvest. Details of two components are explained in the rest of this section.

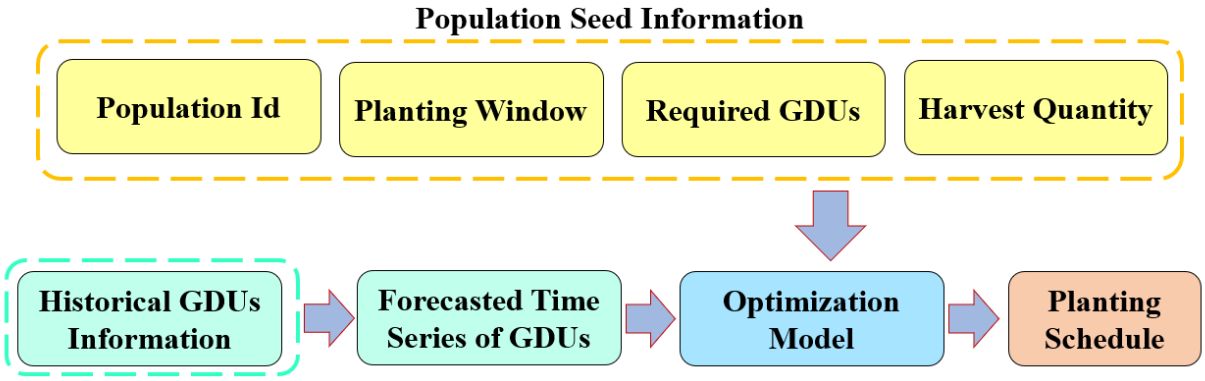

Figure 5.3 The overview of the proposed framework

Weather prediction model

The harvesting date of seed population is determined based on accumulated GDUs in Celsius that seed population received during its planting time. Because the goal is to optimize the planting time for the next years (in this challenge is 2020 and 2021) and the weather information has not yet been observed, each site's GDUs for each calendar day of 2020 and 2021 must be predicted using the time series prediction model. Recent deep learning models have indicated high prediction accuracy in sequence processing and time series problems, particularly recurrent neural network. But, forecasting several steps into the future (in this problem for the next 2 years) is challenging

with regards to keeping the forecasted weather within a reasonable range based on the historical information. Hence, we need a new predictive model to estimate uncertainty in the future. In this paper, a deep recurrent neural network (RNN) using long short-term memory(LSTM) network [21] was designed to capture nonlinear and temporal aspects of the GDUs. Moreover, to address the uncertainty of forecasted GDUs, we trained a Gaussian Process model [42] to predict LSTM's residual errors. Details of LSTM's structure for the prediction of GDUs and the RIO model are described in the rest of this section.

Prediction model design

We designed a new time-series prediction using LSTM network model and a fully connected neural network model to forecast GDUs using historical information. Figure 5.4 indicates the outline of the proposed model. LSTM is an improved version of a RNN model employed widely to classify, process, and predict time-series problems. The main advantage of the LSTM over the conventional RNN model is that LSTM network solve the vanishing gradients problem because of using multiple gates instead of recurrent hidden neurons in their architectures. Also, the main difference between the LSTM models and conventional deep neural network is that LSTM is able to remember temporal dependency and patterns over time due to existing feedback connections in its structure. The structure of the LSTM is illustrated in Figure 5.5.

In the LSTM structure, each time step has a cell with multiple gates as the cell's memory that manages, updates, and controls the flow of information throughout the network. The output of one cell at each time step is the next cell's input at the following time step. LSTM network contains three gates: forget gate, input gate, and output gate. The first sigmoid layer is known as the forget gate that is responsible for deciding what information must be kept and yield to cell state and what useless information must be forgotten. The input gate composes of the combination of the first tanh and the second sigmoid layers, which update the cell state with new encoded information. The output gate that consists of the second tanh and the third sigmoid layers controls the information flow. It decides and encodes part of the cell state as input in the following time step.

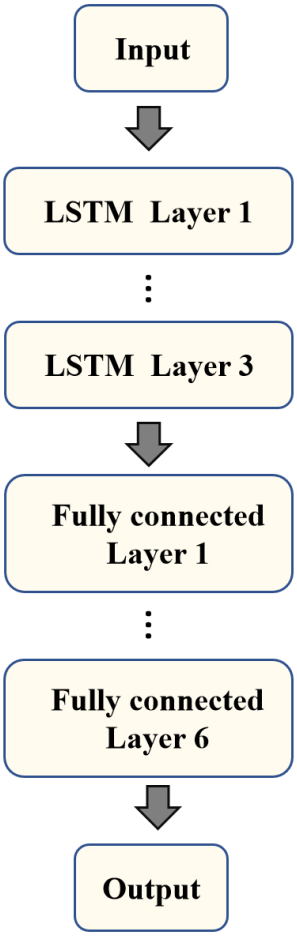

Figure 5.4 Outline of the predictive model structure.

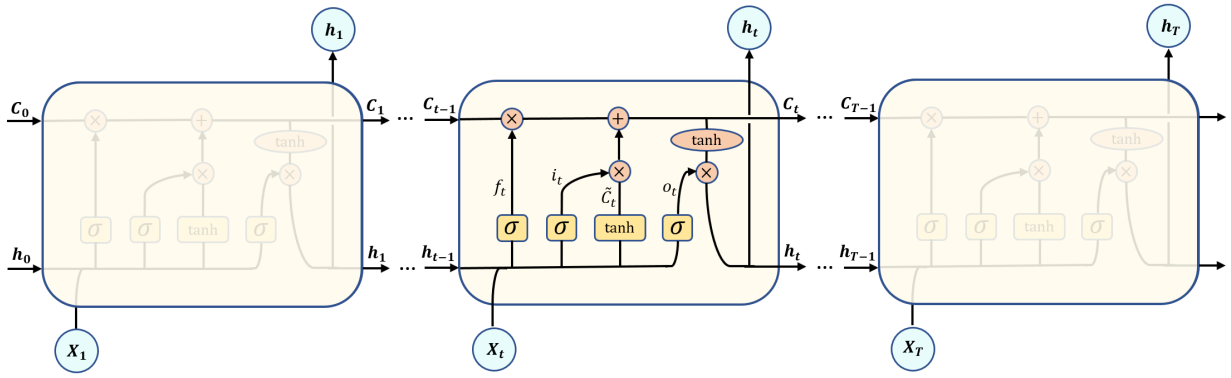

Figure 5.5 The overview of LSTM's structure. The first sigmoid layer is forget gate layer with output $f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f)$. The second sigmoid layer as part of the input gate layer has output $i_t = \sigma(W_i.[h_t - 1, x_t] + b_i)$. The first tanh layer as part of the input generates a vector of new candidate values $\tilde{C}_t = tanh(W_c.[h_t - 1, x_t] + b_c)$. The old cell state C_{t-1} is calculated in the current cell t by $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$. The third sigmoid layer as part of the output gate layer calculates output $o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$. The result of the output gate calculated by second tanh layer as $h_t = o_t * tanh(C_t)$.

We used historical GDUs in Celsius accumulated for each day for both sites from 2009 to 2019 to train the time series prediction model by acquiring short term and long-term dependencies between sequence of GDUs. The previous 30 days of GDU are fed into the proposed time-series model to predict the GDU at the current day such that the proposed network is trained to predict GDU one day in the future. To make predictions far into the future, we can apply the trained model sequentially over the previous predicted GDU. A major limitation of this approach, however, is the dependency on previous predictions, which accumulates prediction errors over time [41]. In the next section, we introduce an auxillary model to improve the performance of the GDU prediction model by estimating and compensating the residual error.

Modeling uncertainty in weather prediction

The idea of the uncertainty estimation of the proposed model is to design another predictive model to estimate the residual error of the proposed network. To develop a more robust estimation far into the future, the Bayesian model can be integrated with the proposed time-series model to measure uncertainty [35, 28, 19, 48]. This paper utilizes another machine learning model to predict the uncertainty directly by predicting residual error and augment the estimated error to the proposed model's prediction. This method is known as RIO (Residual estimation with an I/O kernel) [41]. The RIO's structure is described in the algorithm 5. In this approach, a modified Gaussian Process regression model (GP) [41] is trained to estimate the original residual errors in the training data set. This modified GP uses a new kernel (I/O kernel) that makes use of both inputs and outputs of the proposed time-series model to capture its behavior by estimating the residual error of the proposed time-series model. This I/O kernel is composed of the input kernel that corresponds to the training set and the output kernel that corresponds to the original model's prediction.

To construct the I/O kernel for our proposed model, the last LSTM layer's output and the predicted GDU from the proposed model are fed into the kernel of RIO as input kernel and output kernel. After training the modified GP with the I/O kernel, we estimate a Gaussian distribution for the residual error of the proposed time-series model such that we can compute both the mean and the standard deviation prediction of GDU. The future estimation of GDUs is calculated via Monte Carlo rollouts. Instead of predicting GDU at each day in the future and feeding the predicted value back into the proposed time-series model to predict the next day, we take a sample from the Gaussian distribution returned by RIO. Then this sample is fed back into the model to predict the next day. Sampling from the Gaussian distribution helps uncertainty estimation in the predictions, and we can create several weather scenarios by taking multiple samples from the Gaussian distribution. For this paper, we generated 25 weather scenarios by sampling 25 times from the Gaussian distribution of all days of two test years (2020 and 2021).

Algorithm 5 `RIO Algorithm`

Input: Training set $D = \{(X, Y)\} = \{(x_i, y_i)\}_{i=1}^{n}$ where x_i and y_i is the previous 30 days of GDU at day i and GDu at day i, T as number of days for forecasting.

Output: $\tilde{Y} = \{\tilde{y}_i\}_{i=1}^{T}$ as the prediction of T days far into the future.

Train phase

Train the proposed network by feeding $D = \{(X, Y)\}$ and estimate Y as $\hat{Y}$.

Compute residual error $E = Y - \hat{Y}$.

Feed X back into the network and extract output of the last LSTM layer as $g(X)$.

Train Gaussian process regression using new data set $\{g(X), E\}$ to estimate residual error for x and its prediction $\hat{y}$ as Gaussian distribution $\mathcal{N}(\bar{\hat{e}}, \mathrm{var}(\hat{e}))$, where

$$\bar{\hat{e}} = k((g(x), \hat{y}), (g(X), \hat{Y}))k((g(X), \hat{Y}), (g(X), \hat{Y}))^{-1}E$$

$$\mathrm{var}(\hat{e}) = k((g(x), \hat{y}), (g(x), \hat{y})) - k((g(x), \hat{y}), (g(X), \hat{Y}))k((g(X), \hat{Y}), (g(X), \hat{Y}))^{-1}k((g(X), \hat{Y}), (g(x), \hat{y}))$$

and k denotes I/O kernel $k((x_i, \hat{y}_i), (x_i, \hat{y}_i)) = \sigma_{\mathrm{in}}^2 \exp(-\frac{1}{2l_{\mathrm{in}}^2}||x_i - x_j||^2) + \sigma_{\mathrm{out}}^2 \exp(-\frac{1}{2l_{\mathrm{out}}^2}||\hat{y}_i - \hat{y}_j||^2)$ with the hyperparameters $\sigma_{\mathrm{in}}, l_{\mathrm{in}}^2, \sigma_{\mathrm{out}}, l_{\mathrm{out}}^2$.

Forecast phase

Set $\hat{x}_1 = Y[n - 29 : n]$.

for $t = 1$ to T **do**

    Feed $\hat{x}_t$ into the network and extract output of the last LSTM layer as $g(\hat{x}_t)$ and its prediction as $\hat{y}_t$.

    Use Trained Gaussian process regression model to compute $\bar{\hat{e}}$ and $\mathrm{var}(\hat{e})$.

    The predicted GDU is sampled as follow $\tilde{y}_t \sim \mathcal{N}(\hat{y}_t + \bar{\hat{e}}, \mathrm{var}(\hat{e}))$.

    Set $\hat{x}_{t+1} = [\hat{x}_t[2 : n], \tilde{y}_t]$.

end for

5.4.1 Optimization model

Since the main objective is to optimize the planting time of seed population, we cast the scheduling problem as the optimization model using the predicted GDU as the heuristic measurement for harvesting. For case 1, the optimization model tried to schedule the planting of seed population at a minimum number of harvest weeks so that there is consistent harvest quantity among all weeks, and the capacity constraints are met. While at case 2, the optimization model determines the optimal scheduling of seed population's planting time at the lowest possible capacity required. Two sites do not have interaction with each other, and we can optimize them separately. Moreover, we developed one optimization problem for case 1 (when sites have storage capacity) and one optimization for case 2 (when sites do not have storage capacity). In the following, the variables and parameters used in the model are described.

Sets and indices:

$\mathcal{I}$ Set of seed populations, $i \in \mathcal{I} = \{1, ..., I\}$;

$\mathcal{T}$ Set of days in planting horizon, $t \in \mathcal{T} = \{1, ..., T\}$;

$\mathcal{W}$ Set of weeks in planting horizon, $t \in \mathcal{W} = \{1, ..., W\}$;

$\mathcal{S}$ Set of weather scenarios, $s \in \mathcal{S} = \{1, ..., S\}$.

Parameters:

$\mathcal{R}_i$ Accumulated growing degree units needed for harvesting seed population i;

$\mathcal{E}_i$ Earliest date for planting seed population i;

$\mathcal{L}_i$ Latest date for planting seed population i;

$\mathcal{H}_i$ Harvest quantity (number of ears) for seed population i;

$\mathcal{C}$ Capacity of site for problem case 1;

$\mathcal{P}_s$ Probability of weather scenario s;

$\mathcal{G}_{t,s}$ GDUs during day t at weather scenario s;

$\mathcal{M}_{w,t}$ Binary parameter indicating whether the day t belongs to week w ($\mathcal{M}_{w,t} = 1$) or not ($\mathcal{M}_{w,t} = 0$);

$y_{i,t,s,w}$ Harvest quantity of seed population i in week w at weather scenario s when it is planted in day t. $y_{i,t,s,w}$ is computed as $y_{i,t,s,w} = \mathcal{H}_i * \mathcal{M}_{w,t'} \forall i \in \mathcal{I}, t \in \mathcal{T}, \mathcal{E}_i \leq t \leq \mathcal{L}_i, s \in \mathcal{S}, w \in \mathcal{W}$ where $t' \leq T$ so that $\sum_{t''=t}^{t'-1} \mathcal{G}_{t'',s} \geq \mathcal{R}_i$ and $\sum_{t''=t}^{t'-1} \mathcal{G}_{t'',s} - \mathcal{R}_i \leq \mathcal{G}_{t,s}$, otherwise, $y_{i,t,s,w} = 0$.

Decision variables:

$x_{i,t}$ Binary variable indicating whether the seed population i is planted in day t ($x_{i,t} = 1$) or not ($x_{i,t} = 0$);

z Auxiliary variable indicating maximum harvesting amount among all weeks and weather scenarios.

The mathematical programming model for case 1 is formulated as the following optimization model in Equations (5.1-5.5).

$$\min \sum_{s \in \mathcal{S}} \mathcal{P}_s \max_{w \in \mathcal{W}} \{\mathcal{C} - \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} y_{i,t,s,w} x_{i,t}\} \tag{5.1}$$

$$\text{s.t.} \sum_{t=\mathcal{E}_i}^{\mathcal{L}_i} x_{i,t} = 1 \qquad \forall i \in \mathcal{I} \tag{5.2}$$

$$\sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} x_{i,t} = I \tag{5.3}$$

$$\sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} y_{i,t,s,w} x_{i,t} \leq \mathcal{C} \qquad \forall s \in \mathcal{S}, w \in \mathcal{W} \tag{5.4}$$

$$x_{i,t} = \{0, 1\} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{5.5}$$

The objective function in Equation (5.1) is to minimize the expected maximum difference between the capacity and the weekly harvest quantity among all harvest weeks. Constraints (5.2) and (5.3) specify the planting date of each seed population between their earliest and latest planting dates. Constraint (5.4) limits the weekly harvest quantity within existing capacity. Constraint (5.5) is the definition of binary decision variables.

After creating weather scenarios for the next planting and harvesting calendar, we solve the optimization model (5.1)-(5.5). It is better to use Equation (5.6) instead of the Equation (5.1) for objective function because it better reflects the fluctuation in weekly harvest quantity among all harvest weeks. Equation (5.6) computes the difference of all pairs of the weekly harvest quantity among all harvest weeks. But using that makes the model intractable for the large size of the problem. Hence, we use the Equation (5.1) as the objective function to solve the model within a reasonable time and make the model tractable for large problem. Since one of the evaluation criteria is to minimize the total number of harvest weeks, we iteratively shrink the available weeks for harvest so that the model (5.1)-(5.5) cannot result in harvesting in these weeks. Then, we calculate Equation (5.6) just for the harvesting period (from the first harvest week to the last harvest week), and then we select the best period of harvesting time regarding minimizing Equation (5.6).

$$\sum_{s\in\mathcal{S}} \mathcal{P}_s \sum_{w\in\mathcal{W}} \sum_{w'\in\mathcal{W},w<w'} |\sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}} y_{i,t,s,w}x_{i,t} - \sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}} y_{i,t,s,w'}x_{i,t}| \tag{5.6}$$

The optimization model for case 2 (model (5.7)-(5.11)) is formulated as the same as case 1 by modifying objective function and the Constraint (5.4) to accommodate the model for the uncapacitated version. After finding the minimum capacity using model (5.7)-(5.11), the model (5.1)-(5.5) are applied to schedule the planting time of seed population.

$$\min \quad z \tag{5.7}$$

$$\text{s.t.} \sum_{t=\mathcal{E}_i}^{\mathcal{L}_i} x_{i,t} = 1 \qquad \forall i \in \mathcal{I} \tag{5.8}$$

$$\sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}} x_{i,t} = I \tag{5.9}$$

$$\sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}} y_{i,t,s,w}x_{i,t} \leq z \qquad \forall s \in \mathcal{S}, w \in \mathcal{W} \tag{5.10}$$

$$x_{i,t} = \{0,1\} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{5.11}$$

The objective function in Equation (5.7) is to minimize the maximum capacity required among all harvest weeks. Constraints (5.8) and (5.9) assign the planting date of each seed population between their earliest and latest dates of planting. Constraint (5.10) calculates the maximum harvesting quantity among all harvest weeks. Constraint (5.11) is the definition of binary decision variables.

5.4.2 Experiment settings

The proposed time-series model (both LSTM network and fully connected layers) was implemented in python using the TensorFlow package [1]. Parameter tuning of the hyperparameters of LSTM and fully connected layers indicated that the LSTM layer with 20 units and a dense layer with 20 neurons and a rectified linear unit (ReLU) activation function resulted in the most accurate model to capture the nonlinear and temporal aspects of the weather information. To tune the parameter, we used a time-wise five-fold cross-validation, as shown in Figure 5.6. Each fold corresponds to particular six months as test data for prediction, and the rest of the data from 2009 up to test data corresponds to the training set. We applied Adam optimizer [29] with a learning rate of 0.001 and a mini-batch size of 32. Adam optimizer tries to minimize mean absolute error (MAE) instead of mean squared error (MSE) because MAE is more robust in training the model with noisy training data.

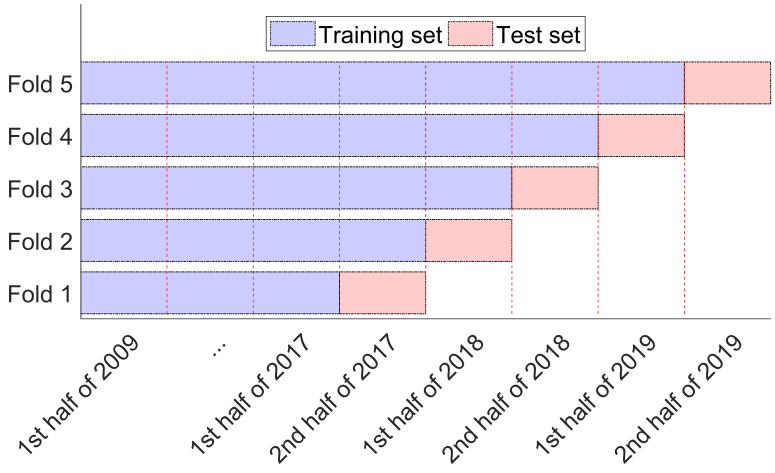

Figure 5.6 Partition of training and test data sets for cross-validation.

To compare with the proposed time-series structure with the state-of-the-art, two different deep learning structures (convolutional neural network and deep fully connected neural network) were deployed. Details of CNN and DNN models are provided as follows.

- **DNN**: DNN with 5 nonlinear layers is implemented in Python by using the TensorFlow package [1]. Each layer has 20 neurons and a ReLU activation function. We used the batch normalization [24] to increase the prediction accuracy.

- **CNN**: CNN with three convolution layers and three max-pooling layers is implemented in Python by using the TensorFlow package [1]. The output of the last max-pooling layer is flattened and fed into two fully connected layers with 100 neurons and a ReLU activation function.

To tune the DNN parameters (numbers of hidden layers and neurons at each layer) and CNN parameters (numbers of convolution layers, fully connected layers and their neurons), we used a time-wise five-fold cross-validation (shown in Figure 5.6) which led to the lowest cross-validation prediction error. Adam optimizer [29] with a learning rate of 0.001, and a mini-batch size of 32 were applied to train the DNN and CNN model with regards to minimizing MAE.

The formulated optimization models were implemented in Python 3. Then, they were solved with the Gurobi optimizer version 9.0 [37].

5.5 Quantitative results

In this section, we report the computational experiments conducted in this research to test the proposed structure's performance in predicting weather into the future and the optimization models' performance in scheduling the planting time of seed population with more consistent harvest quantity among all weeks.

5.5.1 Prediction accuracy comparison with other machine learning models

We compared the performance of the proposed time-series structure with the state-of-the-art in terms of three criteria: RMSE, which indicates the difference between predicted and observed weather, relative RMSE (RRMSE), which represents the normalized difference between predicted and observed weather, and coefficient of determination (R^2), which computes the proportion of the variance in the weather that is explained by independent variables. Table 5.1 summarizes the daily benchmark of GDU prediction performance of the proposed structure and other models over five test years (2015-2019) to illustrate the impact of the proposed model. These results indicate that the proposed time-series model outperformed other machine learning models for all test years for both sites in all evaluation criteria.

Table 5.1 Daily prediction performance of three time-series models for five test years (2015 to 2019) at sites 0 and 1.

criterion	Site	Method	Test Year				
			2015	2016	2017	2018	2019
RMSE	Site 0	DNN	0.475	0.493	0.531	0.509	0.428
		CNN	0.585	0.602	0.632	0.577	0.544
		Proposed	0.429	0.471	0.476	0.447	0.404
	Site 1	DNN	0.718	0.834	0.882	0.747	0.702
		CNN	1.078	1.106	1.112	1.054	1.018
		Proposed	0.689	0.727	0.767	0.710	0.684
RRMSE	Site 0	DNN	5.03%	5.33%	5.95%	5.84%	5.07%
		CNN	6.21%	6.5%	7.08%	6.62%	6.43%
		Proposed	4.55%	5.09%	5.33%	5.12%	4.77%
	Site 1	DNN	6.74%	7.71%	8.42%	7.22%	6.5%
		CNN	10.13%	10.23%	10.61%	10.18%	9.42%
		Proposed	6.47%	6.72%	7.32%	6.85%	6.33%
R^2	Site 0	DNN	0.971	0.972	0.977	0.971	0.983
		CNN	0.956	0.959	0.968	0.963	0.973
		Proposed	0.976	0.975	0.982	0.978	0.985
	Site 1	DNN	0.744	0.736	0.709	0.759	0.714
		CNN	0.423	0.536	0.539	0.522	0.399
		Proposed	0.764	0.799	0.780	0.783	0.728

Figures 5.7 and 5.8 illustrate the consistency of daily prediction of GDU with actual GDU at two test years (2018 and 2019) for sites 0 and 1, respectively. For this prediction, the previous 30 days of GDU are fed into the proposed time-series model to predict GDU one day in the future. The proposed model also shows its ability to capture both the overall trend of GDU over test years and GDU fluctuations from one day to another.

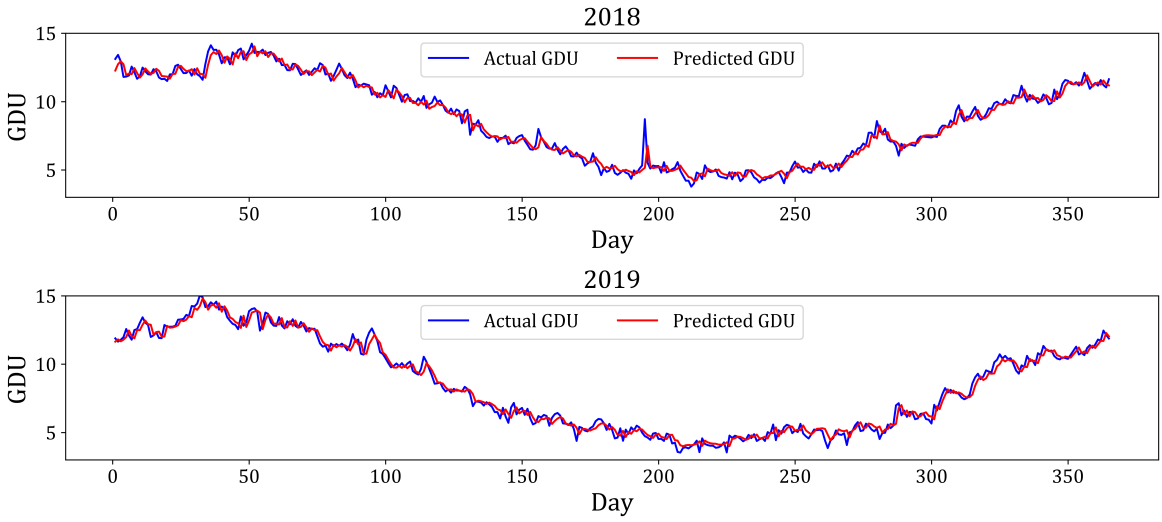

Figure 5.7 Daily GDU predictions of site 0 for test years 2018 and 2019.

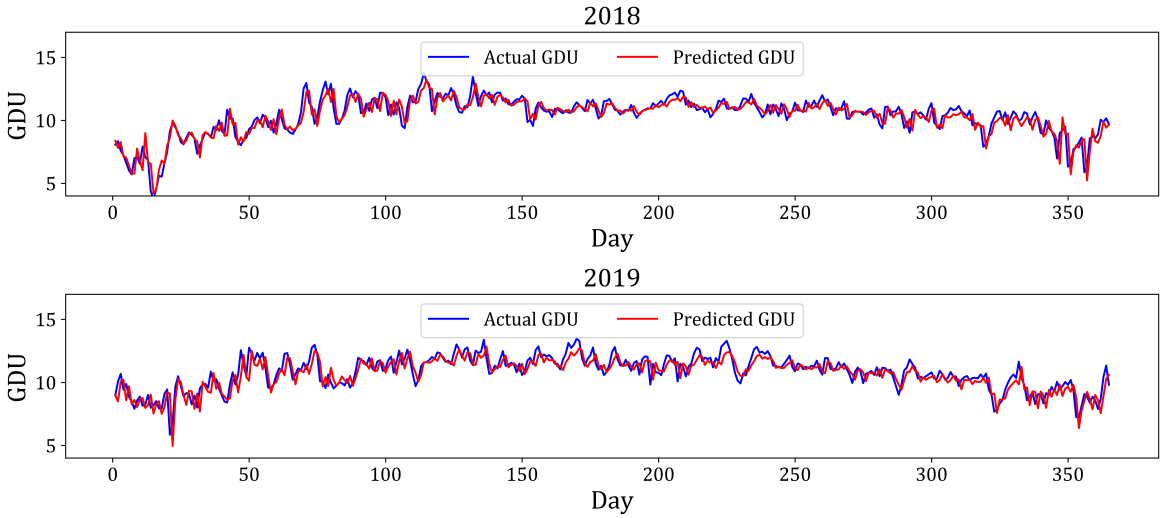

Figure 5.8 Daily GDU predictions of site 1 for test years 2018 and 2019.

The results of using the RIO to estimate uncertainty in the prediction of the GDU into the future (next two years) for sites 0 and 1 were visualized in Figure 5.9 and 5.10, respectively. The proposed time-series model was trained on data up to the end of 2019, and the predictions started on the first of 2020, and it then predicted the GDU 730 days into the future. To model uncertainty of weather for the next planting and harvesting calendar (2020 and 2021), the RIO approach was

used to create 25 weather scenarios. The shadow areas represent the confidence interval of weather prediction, which indicates the range of variability across 25 weather scenarios. We used 25 weather scenarios to formulate the stochastic optimization model under weather uncertainty on the given calendar day of 2020.

The predicted GDU can be compared to the actual GDU during the historical period (2009-2019), and thus the forecasted GDUs into the future follow meaningful trajectories. This result can be attributed to sampling from the Gaussian distribution via Monte Carlo rollouts to estimate weather into the future instead of predicting only by the proposed time-series model and feeding it back into the model to predict the next step.

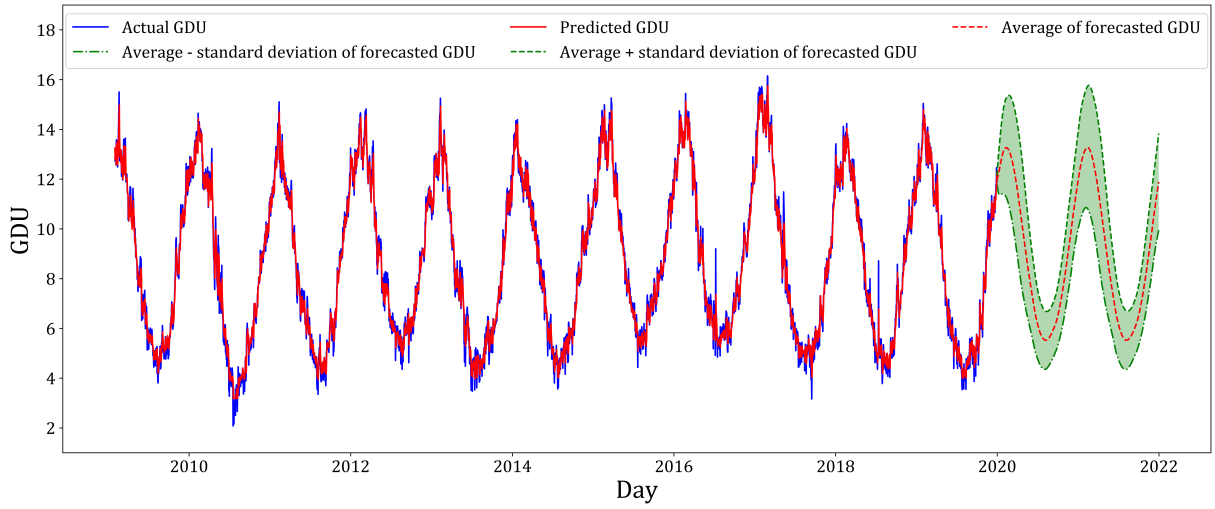

Figure 5.9 Forecasted GDU and its uncertainty at site 0 into the future (years 2020 and 2021) using RIO algorithm.

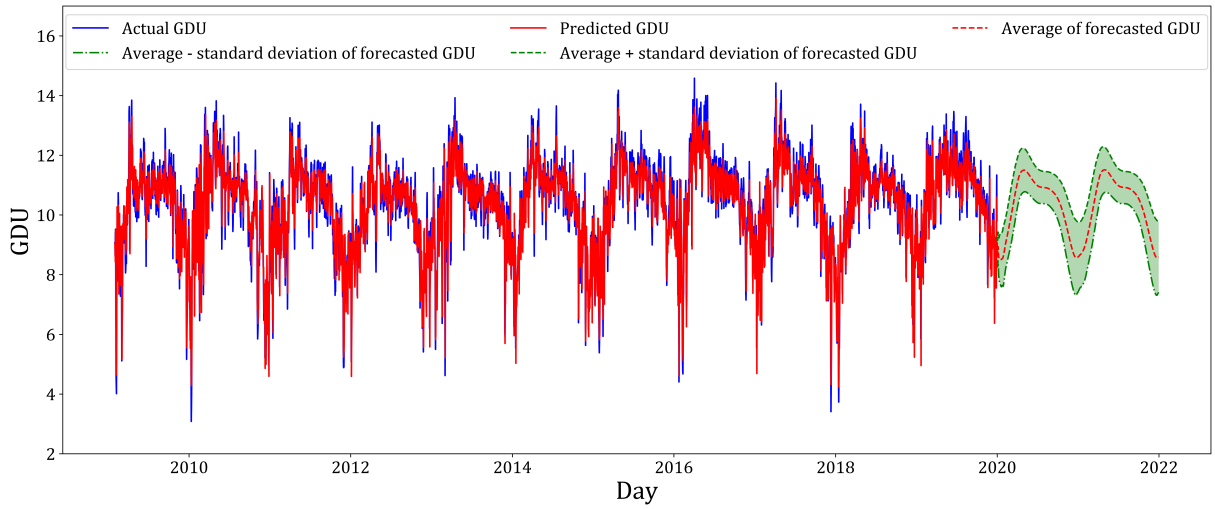

Figure 5.10 Forecasted GDU and its uncertainty at site 1 into the future (years 2020 and 2021) using RIO algorithm.

5.5.2 Optimal schedule for planting time of seed population

The optimization model's goal in case 1 (sites have capacities) is to schedule the seed population's planting time within the fewest number of weeks. Hence, we iteratively limited the first and the last week harvest weeks to the specific shorter periods for two sites, and then the optimization model (5.1)-(5.5) scheduled the best planting time for seed populations so that the harvest must be done in these predefined periods. Figure 5.11 shows the objective values of Equation (5.6) for the various harvesting periods at sites 0 and 1. The results show that the best planting dates with the highest consistent weekly harvest quantity and the fewest harvest weeks are when the allowed harvesting weeks are week 19 to week 67 for site 0 and from week 16 to week 67 for site 1. Figures 5.12 and 5.13 illustrate the original and optimal weekly harvest quantities at sites 0 and 1 in case 1 using the average of forecasted GDU. Table 5.2 reports the maximum required capacity, harvesting period, and value of Equation (5.6). Our proposed model decreased the required capacity by 69% at site 0 and 48% at site 1 compared to the original planting time. Also, the proposed approach reduced the harvesting period by 1 week.

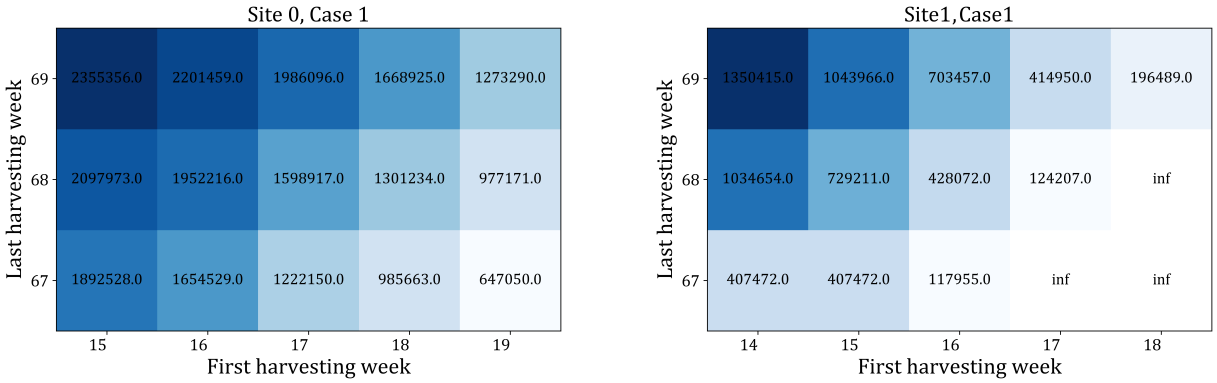

Figure 5.11 The objective values of Equation (5.6) for the different allowed harvesting periods at sites 0 and 1. The inf value refers to the infeasibility of the optimization model (5.1)-(5.5). The numbers in each block refer to the value of Equation (5.6). The darker blocks have higher objective function values, and they are not optimal.

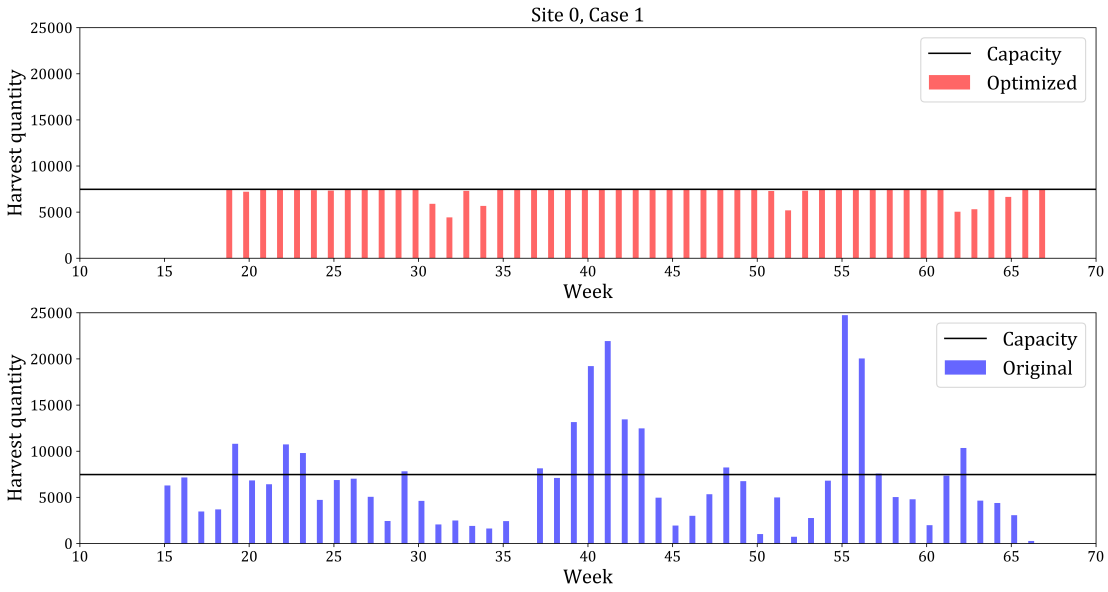

Figure 5.12 The original and optimal weekly harvest quantities at site 0 in case 1 using the average of forecasted GDU.

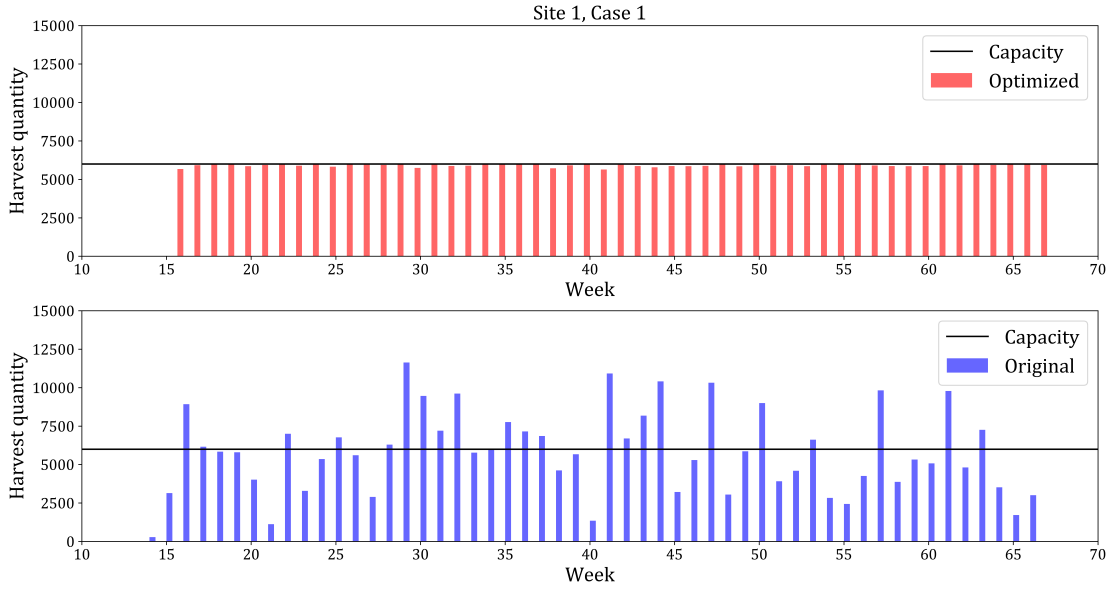

Figure 5.13 The original and optimal weekly harvest quantities at site 1 in case 1 using the average of forecasted GDU.

Table 5.2 Optimal and original planting times for case 1.

Method	Site 0				
	Objective function	First harvesting time	Last harvesting time	Harvesting period	Maximum required capacity
Original	7,410,283	15	66	52	24,736
Optimal	647,050	19	67	49	7,475
Method	Site 1				
	Objective function	First harvesting time	Last harvesting time	Harvesting period	Maximum required capacity
Original	4,263,080	14	66	53	11,632
Optimal	117,955	16	67	52	6,000

The aim of the optimization model in case 2 (sites have no capacities) is to schedule the seed population's planting time at the lowest capacity required for both sites as well as the fewest

number of weeks. Solving the optimization model (5.7)-(5.11) for both sites suggested that the lowest capacity required for sites 0 and 1 are 10,658 and 7,875. Then, the optimization model (5.1)-(5.5) was solved for various harvesting periods with determined capacities. For this purpose, we limited the model to determine the seeds' planting times so that their harvests happened in the limited harvesting periods. The best harvesting periods in terms of minimizing Equation (5.6) are reported in Figure 5.14 for sites 0 and 1. The best harvesting period for site 0 is week 19 to week 66 and for site 1 is week 15 to week 69. The results of solving the optimization model (5.1)-(5.5) with determined capacities from model (5.7)-(5.11) and optimal harvest week for both sites are shown in Figures 5.15 and 5.16. These figures indicate the original and optimal weekly harvest quantities at sites 0 and 1 in case 2 using the average of forecasted GDU. Table 5.3 shows that our proposed model found the lowest required capacities by decreasing the capacity by 69% at site 0 and by 51% at site 1.

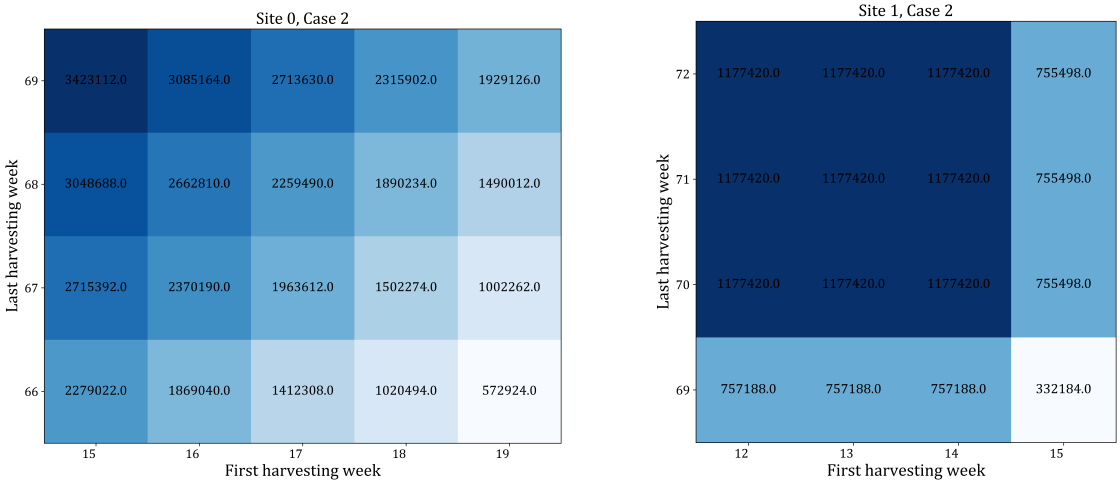

Figure 5.14 The objective values of Equation (5.6) for the different allowed harvesting periods at sites 0 and 1. The numbers in each block refers to value of Equation (5.6). The darker blocks have higher objective function values, and they are not optimal.

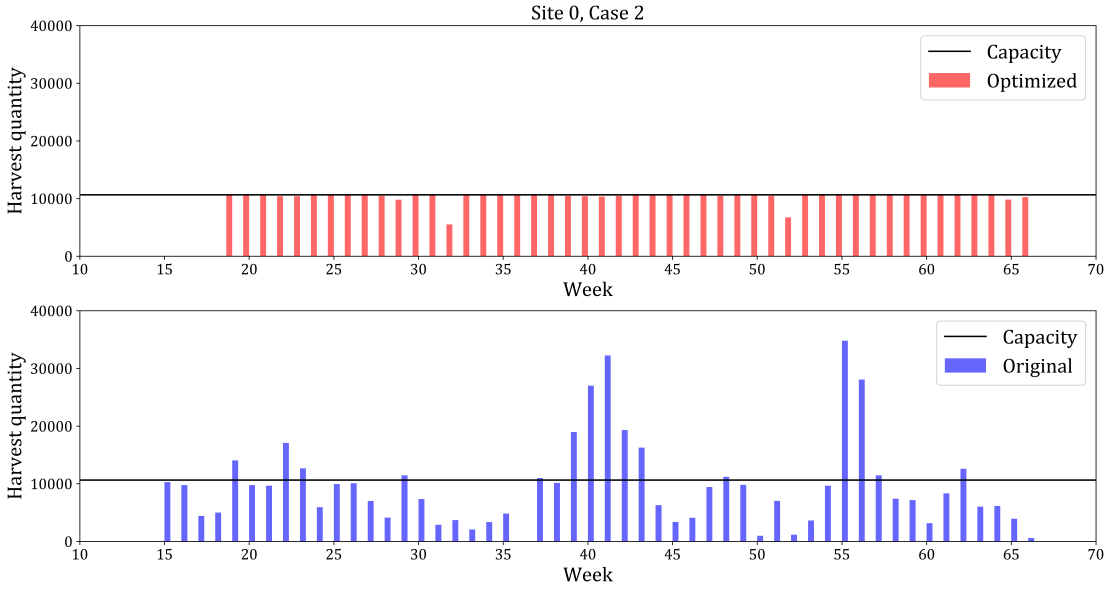

Figure 5.15 The original and optimal weekly harvest quantities at site 0 in case 2 using the average of forecasted GDU.

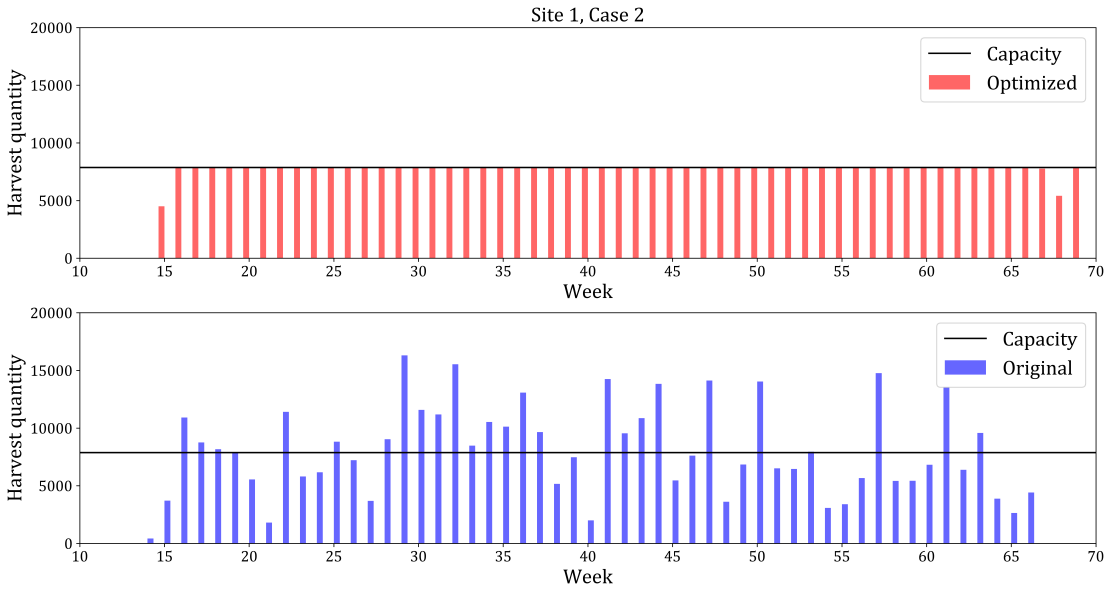

Figure 5.16 The original and optimal weekly harvest quantities at site 1 in case 2 using the average of forecasted GDU.

Table 5.3 Optimal and original planting times for case 2.

Method	Site 0				
	Objective function	First harvesting time	Last harvesting time	Harvesting period	Maximum required capacity
Original	10,362,758	15	66	52	34,799
Optimal	572,924	19	66	48	10,658
Method	Site 1				
	Objective function	First harvesting time	Last harvesting time	Harvesting period	Maximum required capacity
Original	6,210,306	14	66	53	16,299
Optimal	332,184	15	69	55	7,875

5.6 Conclusion

We developed a new framework with the combination of time-series model and optimization models to address the 2021 Syngenta crop challenge by scheduling the planting time of seed populations at the lowest capacity required and the fewest number of harvest weeks. This challenge is to optimize the seed populations' planting times during 2020. Hence, the unseen weather information at the given calendar days was forecasted by the proposed time-series model that consists of the LSTM model and fully connected deep learning. To estimate the uncertainty of the weather forecast into the future, we used the RIO model. The results reported the forecasted weather follows historical trajectories. By having the weather scenarios, we proposed a stochastic optimization model to schedule the farming system. Results from the computational experiment suggested that the optimization model achieved a more consistent weekly harvest quantity in fewer harvesting weeks.

5.7 References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th Symposium on Operating Systems Design and Implementation 16)*, pages 265–283.

[2] Ahumada, O. and Villalobos, J. R. (2009). Application of planning models in the agri-food supply chain: A review. *European Journal of Operational Research*, 196(1):1–20.

[3] Anapalli, S. S., Pettigrew, W. T., Reddy, K. N., Ma, L., Fisher, D. K., and Sui, R. (2016). Climate-optimized planting windows for cotton in the lower Mississippi delta region. *Agronomy*, 6(4):46.

[4] Araya, A., Stroosnijder, L., Habtu, S., Keesstra, S. D., Berhe, M., and Hadgu, K. M. (2012). Risk assessment by sowing date for barley (hordeum vulgare) in northern ethiopia. *Agricultural and Forest Meteorology*, 154:30–37.

[5] Baum, M., Archontoulis, S., and Licht, M. (2019). Planting date, hybrid maturity, and weather effects on maize yield and crop stage. *Agronomy Journal*, 111(1):303–313.

[6] Baum, M. E., Licht, M. A., Huber, I., and Archontoulis, S. V. (2020). Impacts of climate change on the optimum planting date of different maize cultivars in the central US corn belt. *European Journal of Agronomy*, 119:126101.

[7] Boyabatlı, O., Nasiry, J., and Zhou, Y. (2019). Crop planning in sustainable agriculture: Dynamic farmland allocation in the presence of crop rotation benefits. *Management Science*, 65(5):2060–2076.

[8] Brisson, N., Gary, C., Justes, E., Roche, R., Mary, B., Ripoche, D., Zimmer, D., Sierra, J., Bertuzzi, P., Burger, P., et al. (2003). An overview of the crop model STICS. *European Journal of Agronomy*, 18(3-4):309–332.

[9] Cammarano, D., Payero, J., Basso, B., Stefanova, L., and Grace, P. (2013). Adapting wheat sowing dates to projected climate change in the australian subtropics: analysis of crop water use and yield. *Crop and Pasture Science*, 63(10):974–986.

[10] Chang, K.-H., Warland, J. S., Bartlett, P. A., Arain, A. M., and Yuan, F. (2014). A simple crop phenology algorithm in the land surface model cn-class. *Agronomy Journal*, 106(1):297–308.

[11] Cid-Garcia, N. M., Bravo-Lozano, A. G., and Rios-Solis, Y. A. (2014). A crop planning and real-time irrigation method based on site-specific management zones and linear programming. *Computers and Electronics in Agriculture*, 107:20–28.

[12] Dobor, L., Barcza, Z., Hlásny, T., Árendás, T., Spitkó, T., and Fodor, N. (2016). Crop planting date matters: Estimation methods and effect on future yields. *Agricultural and Forest Meteorology*, 223:103–115.

[13] Drewniak, B., Song, J., Prell, J., Kotamarthi, V., and Jacob, R. (2013). Modeling agriculture in the community land model. *Geoscientific Model Development*, 6(2).

[14] Egli, D. and Cornelius, P. (2009). A regional analysis of the response of soybean yield to planting date. *Agronomy Journal*, 101(2):330–335.

[15] Elliott, J., Müller, C., Deryng, D., Chryssanthacopoulos, J., Boote, K., Büchner, M., Foster, I., Glotter, M., Heinke, J., Iizumi, T., et al. (2015). The global gridded crop model intercomparison: data and modeling protocols for phase 1 (v1. 0). *Geoscientific Model Development (Online)*, 8(2).

[16] Fodor, N., Pásztor, L., et al. (2010). The agro-ecological potential of hungary and its prospective development due to climate change. *Applied Ecology and Environmental Research*, 8(3):177–190.

[17] Foley, J. A., Ramankutty, N., Brauman, K. A., Cassidy, E. S., Gerber, J. S., Johnston, M., Mueller, N. D., O'Connell, C., Ray, D. K., West, P. C., et al. (2011). Solutions for a cultivated planet. *Nature*, 478(7369):337–342.

[18] Gadallah, A. M., Mohamed, A. H., and Hefny, H. A. (2014). Fuzzy query approach for crops planting dates optimization based on climate data. In *International Conference on Advanced Machine Learning Technologies and Applications*, pages 436–445. Springer.

[19] Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059.

[20] Godfray, H. C. J., Beddington, J. R., Crute, I. R., Haddad, L., Lawrence, D., Muir, J. F., Pretty, J., Robinson, S., Thomas, S. M., and Toulmin, C. (2010). Food security: the challenge of feeding 9 billion people. *Science*, 327(5967):812–818.

[21] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[22] Holzworth, D. P., Huth, N. I., deVoil, P. G., Zurcher, E. J., Herrmann, N. I., McLean, G., Chenu, K., van Oosterom, E. J., Snow, V., Murphy, C., et al. (2014). Apsim–evolution towards a new generation of agricultural systems simulation. *Environmental Modelling & Software*, 62:327–350.

[23] Indriyanti, A. D., Prehanto, D. R., Permadi, G. S., Mashuri, C., and Vitadiar, T. Z. (2019). Using fuzzy time series (FTS) and linear programming for production planning and planting pattern scheduling red onion. In *E3S Web of Conferences*, volume 125, page 23007. EDP Sciences.

[24] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv preprint ArXiv:1502.03167*.

[25] Jones, J. W., Hoogenboom, G., Porter, C. H., Boote, K. J., Batchelor, W. D., Hunt, L., Wilkens, P. W., Singh, U., Gijsman, A. J., and Ritchie, J. T. (2003). The dssat cropping system model. *European Journal of Agronomy*, 18(3-4):235–265.

[26] Keating, B. A., Carberry, P. S., Hammer, G. L., Probert, M. E., Robertson, M. J., Holzworth, D., Huth, N. I., Hargreaves, J. N., Meinke, H., Hochman, Z., et al. (2003). An overview of APSIM,

a model designed for farming systems simulation. *European Journal of Agronomy*, 18(3-4):267–288.

[27] Kessler, A., Archontoulis, S. V., and Licht, M. A. (2020). Soybean yield and crop stage response to planting date and cultivar maturity in Iowa, USA. *Agronomy Journal*, 112(1):382–394.

[28] Kim, H., Mnih, A., Schwarz, J., Garnelo, M., Eslami, A., Rosenbaum, D., Vinyals, O., and Teh, Y. W. (2019). Attentive neural processes. *ArXiv preprint ArXiv:1901.05761*.

[29] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[30] Li, M., Guo, P., Zhang, L., and Zhang, C. (2016). Uncertain and multi-objective programming models for crop planting structure optimization. *Front. Agric. Sci. Eng*, 3:34–45.

[31] Lowe, T. J. and Preckel, P. V. (2004). Decision technologies for agribusiness problems: A brief review of selected literature and a call for research. *Manufacturing & Service Operations Management*, 6(3):201–208.

[32] May, W. E., Mohr, R. M., Lafond, G. P., Johnston, A. M., and Stevenson, F. C. (2004). Early seeding dates improve oat yield and quality in the eastern prairies. *Canadian Journal of Plant Science*, 84(2):431–442.

[33] Miller, P., Lanier, W., and Brandt, S. (2001). Using growing degree days to predict plant stages. *Ag/Extension Communications Coordinator, Communications Services, Montana State University-Bozeman, Bozeman, MO*, 59717(406):994–2721.

[34] Moore, A. D., Holzworth, D. P., Herrmann, N. I., Brown, H. E., de Voil, P. G., Snow, V. O., Zurcher, E. J., and Huth, N. I. (2014). Modelling the manager: representing rule-based management in farming systems simulation models. *Environmental Modelling & Software*, 62:399–410.

[35] Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

[36] Niu, X., Easterling, W., Hays, C. J., Jacobs, A., and Mearns, L. (2009). Reliability and input-data induced uncertainty of the EPIC model to estimate climate change impact on sorghum yields in the US great plains. *Agriculture, Ecosystems & Environment*, 129(1-3):268–276.

[37] OPTIMIZATION, G. (2014). Inc. gurobi optimizer reference manual, 2015. *URL: http://www. gurobi. com*, page 29.

[38] Pal, B. B., Kumar, M., and Sen, S. (2010). A priority based interval-valued goal programming approach for land utilization planning in agricultural system: A case study. In *2010 Second International Conference on Computing, Communication and Networking Technologies*, pages 1–9. IEEE.

[39] Putri, M., Mardhiyyah, Y., and Rusdiansyah, A. (2019). Development of seeding and planting scheduling algorithms for contract farming of organic vegetable with multi seeding and planting center. In *Journal of Physics: Conference Series*, volume 1376, page 012033. IOP Publishing.

[40] Qian, B., De Jong, R., Warren, R., Chipanshi, A., and Hill, H. (2009). Statistical spring wheat yield forecasting for the canadian prairie provinces. *Agricultural and Forest Meteorology*, 149(6-7):1022–1031.

[41] Qiu, X., Meyerson, E., and Miikkulainen, R. (2019). Quantifying point-prediction uncertainty in neural networks via residual estimation with an i/o kernel. *ArXiv preprint ArXiv:1906.00588*.

[42] Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer.

[43] Sacks, W. J., Deryng, D., Foley, J. A., and Ramankutty, N. (2010). Crop planting dates: an analysis of global patterns. *Global Ecology and Biogeography*, 19(5):607–620.

[44] Sadovski, A. et al. (2019). Method for determination of planting and harvesting dates by fuzzy regression. *Ecological Engineering and Environment Protection*, (1):38–45.

[45] Saiyed, I. M., Bullock, P. R., Sapirstein, H. D., Finlay, G. J., and Jarvis, C. K. (2009). Thermal time models for estimating wheat phenological development and weather-based relationships to wheat quality. *Canadian Journal of Plant Science*, 89(3):429–439.

[46] Shang, J., Liu, J., Poncos, V., Geng, X., Qian, B., Chen, Q., Dong, T., Macdonald, D., Martin, T., Kovacs, J., et al. (2020). Detection of crop seeding and harvest through analysis of time-series sentinel-1 interferometric sar data. *Remote Sensing*, 12(10):1551.

[47] Smith, P. (2013). Delivering food security without increasing pressure on land. *Global Food Security*, 2(1):18–23.

[48] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25:2951–2959.

[49] Soler, C. M. T., Sentelhas, P. C., and Hoogenboom, G. (2007). Application of the csm-ceres-maize model for planting date evaluation and yield forecasting for maize grown off-season in a subtropical environment. *European Journal of Agronomy*, 27(2-4):165–177.

[50] Tilman, D., Balzer, C., Hill, J., and Befort, B. L. (2011). Global food demand and the sustainable intensification of agriculture. *Proceedings of the National Scademy of Sciences*, 108(50):20260–20264.

[51] Tofa, A. I., Chiezey, U. F., Babaji, B. A., Adnan, A. A., Beah, A., Adam, A. M., et al. (2020). Modeling planting-date effects on intermediate-maturing maize in contrasting environments in the Nigerian Savanna: An application of DSSAT model. *Agronomy*, 10(6):871.

[52] Twine, T. E., Kucharik, C. J., and Foley, J. A. (2004). Effects of land cover change on the energy and water balance of the mississippi river basin. *Journal of Hydrometeorology*, 5(4):640–655.

[53] Vitadiar, T. Z., Farikhin, F., and Surarso, B. (2018). Production planning and planting pattern scheduling information system for horticulture. In *E3S Web of Conferences*, volume 31, page 10004. EDP Sciences.

[54] Waha, K., Van Bussel, L., Müller, C., and Bondeau, A. (2012). Climate-driven simulation of global crop sowing dates. *Global Ecology and Biogeography*, 21(2):247–259.

[55] Wang, E. and Engel, T. (1998). Simulation of phenological development of wheat crops. *Agricultural Systems*, 58(1):1–24.

[56] Wang, J., Guo, S., Kang, S., Wang, Y., Du, T., and Tong, L. (2020). Joint optimization of irrigation and planting pattern to guarantee seed quality, maximize yield, and save water in hybrid maize seed production. *European Journal of Agronomy*, 113:125970.

[57] Waongo, M., Laux, P., Traoré, S. B., Sanon, M., and Kunstmann, H. (2014). A crop model and fuzzy rule based approach for optimizing maize planting dates in burkina faso, west africa. *Journal of Applied Meteorology and Climatology*, 53(3):598–613.

[58] Yang, Y., Wilson, L. T., and Wang, J. (2017). A spatially explicit crop planting initiation and progression model for the conterminous united states. *European Journal of Agronomy*, 90:184–197.

CHAPTER 6. FUTURE WORK SUMMARY AND DISCUSSION

6.1 Thesis Contributions

This dissertation is applications of machine learning and optimization algorithms in agriculture by proposing explainable predictive models and decision-making models in four different studies: (i) designing new algorithms for detecting multi-effect and multi-way epistatic interactions, (ii) developing an explainable machine learning model for crop yield prediction, (iii) integrating random forest model and an optimization model for G $\times$ E interaction detection for prediction of crosses in plant breeding, and (iv) formulating scheduling planting time through developing an optimization model and time series analysis of the weather.

Chapter 2 presents three new algorithms to detect multi-effect and multi-way epistases Interactions. The first model was to formulate the problem mathematically as the MIQP model. MIQP model guarantees the global optimality using existing algorithms and solvers, but it is a time-consuming way to solve epistasis detection problems, especially for large-size problems. Therefore, two more heuristic algorithms (local search and advanced local search algorithms) were developed to solve high-order interaction detection problems with many features efficiently. These algorithms can efficiently find local solutions. The heuristic algorithm has three salient features: minimizing both train and validation RMSE, being less prone to overfitting problems, and having a maximally tolerable computation time option. These features specify the tradeoff between speed and quality of the solution and reveal their effectiveness of the proposed approaches, especially the heuristic algorithm on computational results compared with several state-of-the-art methods.

Chapter 3 describes the interaction regression model as our new explainable machine learning model to predict crop yield prediction by combining the power of optimization, machine learning, and agronomic insight. Its iterative algorithm attempts to choose a subset of E and M features for crop yield prediction that are spatially and temporally robust. The proposed model solves the

optimization problem to detect the most revealing interaction between E and M features in yield on top of this subset. This transparent and explainable framework can quantify and break down the yield into contributions from weather, soil, management, and their interactions. This quantifying allows agronomists to assess the favorable and unfavorable yield factors. The computational result of the model on a comprehensive case study of corn and soybean yield prediction in 293 counties of Illinois, Indiana, and Iowa from 2015 to 2018 revealed satisfactorily address both prediction accuracy (it outperformed state-of-the-art machine learning algorithms with respect to prediction accuracy) and explainability (it detected interactions that are insightful agronomically).

Chapter 4 provides a new integrated model to predict the yield performance of inbreds and testers based on historical yield data in multiple years and environments. Two random forest models were combined with an optimization-based interaction-detection model to predict hybrids' yield performance. The first random forest is an effective and powerful machine learning model for prediction by deciphering complex nonlinear relationships between input and output variables. We enhance this prediction by augmenting specific interactions with the most significant contribution to yield and are detected by the interactions model. The second random forest finds more complex nonlinear functions on detected interactions to improve the prediction accuracy. This model won first place in the 2020 Syngenta crop challenge in analytics. Our computational results on the 2020 Syngenta crop challenge dataset illustrated the founded interactions were potentially biologically insightful, and the proposed model outperformed other state-of-the-art models.

Chapter 5 describes the optimization model for scheduling the planting time of population seeds in the year-round breeding process. Because of environmental uncertainty during the year-round breeding process next year, the unknown weather information was predicted by combining a deep LSTM model and a fully connected deep learning model. To model uncertainty in the forecasted weather into the future, we trained a modified Gaussian process regression model over the residual error of predicted weather. We generated weather scenarios by sampling from the Gaussian distribution via Monte Carlo rollouts. We then developed a stochastic optimization model to schedule the seed populations' planting time with consistent weekly harvest quantity at the

lowest harvesting time. The computation result on the 2021 Syngenta crop challenge showed the forecasted weather followed historical trajectories. The stochastic optimization model achieved an effective planting time for seed populations with a more consistent weekly harvest quantity in fewer harvesting weeks.

6.2 Future Research

The proposed optimization models in chapters 2 and 3 can be applied to detect interactions between discrete and continuous variables in other case studies, such as disease detection and case-control studies. Optimization algorithms such as Bayesian optimization can be combined with the proposed heuristic interaction detection algorithm to improve the algorithm's efficiency.

For yield prediction problems in chapters 3 and 4, a future direction should focus on the possibility of using additional data (genotype data, plant traits, detailed management strategies, and satellite images) and analyzing weather, soil, and management variables with them and their interactions. Deep learning models can utilize satellite images to predict yield and help the plants' growing stages estimations. They provide additional data for feeding the proposed machine learning models to predict yield more accurately. Hyperparameter optimization methods such as grid search and Bayesian optimization can be applied to find optimal models' parameters in chapters 3 and 4.

Future research can improve the quality of the solution using reinforcement learning algorithms such as Deep Q-learning to solve dynamic decision-making problems for scheduling planting time of seed populations in chapter 5. Casting the scheduling problem as a multi-stage stochastic problem or robust optimization with a high number of scenarios can help the decision-maker find optimal actions and take appropriate actions to modify the initial actions in the future. Considering biological and agronomical discoveries can be used further to improve the prediction accuracy of the growing stage of plants and make more robust scheduling. Moreover, the integration of crop modeling and machine learning models can improve yield predictions and planting time scheduling.