FORECASTING BIOMASS YIELDS WITH MACHINE LEARNING

AND DOMAIN ADAPTATION

by

JONATHAN MITCHELL VANCE

(Under the Direction of Khaled M. Rasheed)

ABSTRACT

Recognizing the crucial role alfalfa plays in global food security, and noting recent successes in an increasing number of domains due to advances in artificial intelligence and machine learning (ML), the current work endeavors to contribute to increasing the efficiency of alfalfa farming through applied ML. We propose a novel ML-based time series forecasting technique that outperforms traditional statistical methods, and we used this as the backbone for a proposed application we call Predict Your CropS (PYCS). This what-if and crop forecasting tool and its underlying techniques are intended to help farmers develop improved contingency plans for possible shortages and surpluses of alfalfa, and they forecast future crop yields based on historical weather data and historical crop yields. Under ideal alfalfa farm management conditions, experiments have shown our forecaster to be surprisingly accurate, producing symmetric mean absolute percent error (sMAPE) scores as low as 9.81%, beating the performance of traditional, non-ML-based techniques like ARIMA and SARIMAX, which is especially encouraging considering that this is a difficult domain. This work also explores estimating historical and present-day tabular data using data synthesis. In that phase of the project, we proposed a novel tabular data synthesizer we call Scale Invariant Tabular Synthesis (SITS), which helps boost the

performance of our ML models by increasing training dataset sizes. We show that our synthesis algorithm leads to R scores over 100% higher than the established synthesizer in this domain. Training with data from one location to estimate historical and present-day data in another location provides insight into which regions can be effectively used to train models to estimate other target regions, especially when the target region's dataset is too small to train its own model. We call this non-local training, and when we include synthesis in the pipeline, we call this approach Synthetic Non-Local Training (SNLT), and it is essentially a form of domain adaptation (DA). Three primary contributions of the work are (1) our novel ML-based forecasting technique, (2) our novel DA technique combining the SITS data synthesizer with pre-training, and (3) combing our DA and forecasting techniques into one enhanced forecaster.

INDEX WORDS: time series, data synthesis, machine learning, crop yield forecasting, precision agriculture, alfalfa, biomass

# FORECASTING ALFALFA YIELDS WITH MACHINE LEARNING

by

JONATHAN MITCHELL VANCE

B.S., The University of Georgia, 2009

A Dissertation Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2023

FORECASTING ALFALFA YIELDS WITH MACHINE LEARNING

by

JONATHAN MITCHELL VANCE

| | |
|---|---|
| Major Professor: | Khaled M. Rasheed |
| Committee: | Hamid R. Arabnia |
| | John A. Miller |
| | Ali Missaoui |

Electronic Version Approved:

Ron Walcott
Vice Provost for Graduate Education and Dean of the Graduate School
The University of Georgia
August 2023

# DEDICATION

*This is dedicated to my wife, son, and parents, without whose support over these years I could not have done it.*

# ACKNOWLEDGEMENTS

I would like to thank my advisor and major professor, Dr. Rasheed, who patiently guided me every step of the way toward completing my dissertation.

I would also like to thank my committee: Dr. Arabnia, Dr. Miller, and Dr. Missaoui. Dr. Arabnia helped spark my initial idea to pursue a Ph.D. Dr. Miller helped fill crucial gaps in my research. Dr. Missaoui provided an essential alternative perspective to this group of computer scientists and was especially helpful during the home stretch of my research.

I would also like to thank Dr. Maier, who was always there to help and offer insight along the way.

Thank you all sincerely.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS ..... v

LIST OF TABLES ..... ix

LIST OF FIGURES ..... x

CHAPTER

1 INTRODUCTION ..... 1

Comparing ML models for estimating past and current yields ..... 3

Synthesizing data for yield classification ..... 3

Forecasting yields with time series data ..... 4

Primary Contributions ..... 6

Dissertation Organization ..... 6

2 LITERATURE REVIEW OF PREDICTING CROP YIELDS WITH ML ..... 8

3 COMPARING MACHINE LEARNING TECHNIQUES FOR ALFALFA BIOMASS YIELD PREDICTION ..... 19

Abstract ..... 20

Introduction ..... 21

Related Work ..... 24

Approach ..... 26

Results ..... 30

Conclusion ..... 31

Future Work ..... 32

4 DATA SYNTHESIS FOR ALFALFA BIOMASS YIELD ESTIMATION ..... 33

Abstract ..... 34

Introduction ..... 34

Related Work ..... 40

Materials and Methods ..... 43

Results ..... 50

Discussion ..... 54

Conclusions ..... 57

5 UTILITY OF DOMAIN ADAPTATION FOR BIOMASS YIELD FORECASTING 59

Abstract ..... 60

Introduction ..... 61

Related Work ..... 64

Materials and Methods ..... 69

Results ..... 76

Discussion ..... 93

Conclusions ..... 95

6 PYCS: PREDICT YOUR CROPS WHAT-IF FORECASTING TOOL ..... 97

Introduction ..... 97

Application ..... 98

Conclusion ..... 110

7 CONCLUSION ..... 112

REFERENCES ..... 116

APPENDICES

A CODE AND DATA ACCESSIBILITY ..... 127

B HYPERPARAMETER GRID VALUES REGRESSION ..... 129

C HYPERPARAMETER GRID VALUES CLASSIFICATION ..... 129

D ADDITIONAL TIME PLOTS ..... 130

E FORECASTING RESULTS WITH SYNTHESIS ONLY ..... 160

# LIST OF TABLES

Page

Table 3.1: Estimator Performance: KY & GA....28

Table 3.2: Estimator Performance: KY, GA, WI, & MS....30

Table 3.3: Datasets per state with number of datapoints....30

Table 4.1: KY & GA Data Example....38

Table 4.2: SD & OH Data Example....39

Table 4.3: Class labels by standard deviation....46

Table 4.4: Train KY, test GA....47

Table 4.5: Train SD, test OH with 8 samples....47

Table 4.6: Train SD, test OH with 10 samples....48

Table 4.7: Train SD, test Highmore, SD w/10 samples....48

Table 4.8: Train SD, test Highmore, SD w/8 samples....49

Table 5.1: SITS vs TVAE, CTGAN....77

Table 5.2: SITS vs TVAE, CTGAN, TDA....78

Table 5.3: DA with Synthesis, averages for OH with LOOCV....79

Table 5.4: DA with Synthesis....80

Table 5.5: Univariate time series....81

Table 5.6: Multivariate time series....82

Table 5.7: Univariate sliding window validation results....83

Table 5.8: Multivariate sliding window validation results....85

Table 5.9: ForDA with SITS vs CTGAN, TVAE w/ XGBoost....86

Table 5.10: ForDA with SITS vs CTGAN, TVAE w/ XGBoost ....86

Table 5.11: ForDA with SITS vs CTGAN, TVAE w/ XGBoost averages ....86

Table 5.12: ForDA results using SITS....88

Table 5.13: Target Athens, GA....91

Table 5.14: LOOCV with MI, OH, SD, and KYs....92

Table 5.15: Source: MI, Target : OH ; and vice-versa....92

Table 6.1: Hypothetical higher radiation ....102

Table 6.2: Hypothetical lower radiation ....102

Table 6.3: Hypothetical very low radiation ....103

Table 6.4: Hypothetical very high radiation ....103

Table 6.5: Hypothetical +20mm high precipitation....104

Table 6.6: Hypothetical +50mm high precipitation....104

Table 6.7: Hypothetical +100mm high precipitation....104

Table 6.8: Hypothetical +200mm high precipitation....105

Table 6.9: Hypothetical -20mm low precipitation....105

Table 6.10: Hypothetical -5 C° low average temperatures....106

Table 6.11: Hypothetical -2 C° low average temperatures....106

Table 6.12: Hypothetical +5 C° high average temperatures....106

Table 6.13: Hypothetical +2 C° high average temperatures....107

Table 6.14: +50mm precipitation, -1 C° average temperatures....108

Table 6.15: -95mm precipitation, +2 C° average temperatures....108

Table 6.16: +50mm precipitation, -10,000 W/$m^2$ solar radiation....109

Table 6.17: -95mm precipitation, +15,000 W/m$^2$ solar radiation.....109

Table 6.18: -95mm precip., +15,000 W/m$^2$ rad., +2 C° avg. temp.....110

# LIST OF FIGURES

Page

Figure 4.1: The SNLT Pipeline....50

Figure 4.2: Confusion matrix for Table 4.4....52

Figure 4.3: Confusion matrix for Table 4.5....53

Figure 4.4: Confusion matrix for Table 4.7....54

Figure 5.1: SITS pseudocode....74

Figure 5.2: Scatterplots from Table 5.2's best....80

Figure 5.3: Time plot, Table 5.7 SVR....84

Figure 5.4: Beresford, SD sliding window RF time plot....85

Figure 5.5: Best results from Table 5.10....87

Figure 5.6: Best results from Table 5.11....87

Figure 5.7: 2nd best run in OH round-robin tests....89

Figure 5.8: Our best run in OH round-robin tests....89

Figure 5.9: Source: KY, Target: Athens, GA....91

Figure 5.10: Source: MI, SD, KY, Target: OH....92

Figure 5.11: Source: OH, Target: East Lansing, MI....93

Figure 6.1: Top section of PYCS GUI....99

Figure 6.2: Bottom section of PYCS GUI....99

# CHAPTER 1

## INTRODUCTION

Alfalfa is a key crop globally, as it is a major livestock feed, and this research focuses on forecasting alfalfa biomass yields. The Earth's climate continues to change, continuously challenging ecosystems, agriculture, and societies as weather and climates wane in predictability. Therefore, among the most significant problems humanity faces is how we can continue to eat, which necessitates efficient farming and precision agriculture. Working toward a solution to the problem of forecasting crop yields in a changing climate is the primary goal of this research. Alfalfa is sometimes referred to as the "Queen of the forage crops", and global food security hinges on it due to its monetary value, its sustainability, and its protein-richness. Alfalfa is an important livestock feed, and humans consume also consume it, usually in the form of sprouts (El-Ramady). However, climate change currently exacerbates challenges with alfalfa's efficient and sustainable cultivation (Kulkarni). Speaking to alfalfa's significance, over twenty U.S. land-grant universities conduct systematic alfalfa research, and they commonly publish variety trials that detail their findings. The current work aggregates those variety trials with corresponding weather data from the growing periods, and the resulting aggregated datasets provide the basis for our experiments. Furthermore, in 2015, the United Nations (U.N.) released to the public their "Sustainable Development Goals". Those 17 goals "are the blueprint to achieve a better and more sustainable future for all. They address the global challenges we face, including poverty, inequality, climate change, environmental degradation, peace and justice." Those goals include "Goal 2: Zero Hunger" and "Goal 13: Climate Action," and our research seeks to further progress toward them.

In 2020, the U.N. celebrated the 5th anniversary of the adoption of the goals in 2020, renewing their commitment to them (United Nations).

The project described in the current dissertation focuses on alfalfa yields and weather data as time series, and we propose a novel technique to forecast crop yields using models trained on historical yields and weather data. We hope that the end application powered by this technique will be useful to farmers, especially as climate change and global warming make the Earth's climate and weather less predictable (Sheshadri). Related literature on crop yield prediction, including traditional low-tech methods and high-tech approaches, has shown that accurate forecasting is very difficult in this domain (Gopal) (Sharma). One reason for this is that yield data are abundant nationally, but scarce locally, at least in the context of ML. Data from disparate locations, even geographically close ones with similar climates, cannot always be combined to create larger, more viable training datasets.

This work also recognizes a growing concern regarding the potential overuse of chemicals in precision agriculture, especially Monsanto's ubiquitous RoundUp herbicide, which the World Health Organization has labeled as a probable carcinogen (Duke and Powles). Therefore, this work attempts to exclude Roundup-ready (RR) crops from its analysis, aiming instead to encourage improving practices that do not involve herbicides. Likewise, this research focuses on rainfed alfalfa crops rather than artificially irrigated growing sites, as it aims to help farmers manage their crops in their actual changing climates. While working on these problems, the author has developed novel techniques involving artificial intelligence (AI) and ML. This work culminates in a software application, or what-if tool, called Predict Your CropS (PYCS), which could be useful to scientists for research, and to farmers in preparing for various weather scenarios. For example, consider a hypothetical agricultural supply chain where a livestock herd requires 10 tons of alfalfa

feed per season. If a farmer can fairly guess that between 2400mm and 2800mm of rain will fall this season, then they could use the PYCS tool to help determine how much alfalfa will grow in either instance, or anywhere in between. Then, the farmer could develop contingency plans to handle the potential resulting shortages and surpluses.

This manuscript-style dissertation is organized into six chapters, the middle three of which (3, 4, and 5) are published articles authored by the current author et al., and reprinted with permission from the publishers. These middle chapters represent three distinct phases in which the project has progressed.

Comparing ML models for estimating past and current yields

In the first phase of this project, Whitmire et al. showed that ML combined with feature selection techniques, in the task of estimating alfalfa biomass yields (C. V. Whitmire). Vance et al. extend this by focusing on comparing ML models, expanding the training datasets, and exploring trivial domain adaptation, or non-local training, where models trained on data from one location are used to estimate yields in another. They also show that expanding training datasets by combining multiple locations often leads to better results when training and testing on historical and present tabular data. That work reports $R^2$ scores as high as 0.982 on such tasks (J. R. Vance).

Synthesizing data for yield classification

Motivated by recent progress in transfer learning (TL) and domain adaptation (DA), the second phase of this project focused on improving non-local training as a preliminary step toward true DA. We reframed our regression problem as a classification problem, and we labeled our yields as high, medium, or low, based on how many standard deviations they were from the average in

tons per acre. The number of standard deviations was variable, and they are detailed in Chapter 4. We also considered very small target datasets for prediction, as real-world farmers would not likely be able to produce target datasets as rich as ours. We explored using tabular data synthesis to expand our training datasets, and we found that this led to better accuracy than plain non-local training, and it helped us take advantage of deep learning (DL) models like extreme gradient boosting (XGBoost). Two robust tabular synthesizers we applied were called conditional tabular generative adversarial network (CTGAN), and tabular variational autoencoder (TVAE), and Chapter 3 details how they work (Xu). Together, the tabular data synthesis and ML estimation comprise a pipeline we call synthesized non-local training (SNLT, pronounced like sunlight). We also began the development of the PYCS application and user interface in this phase (J. Vance, PYCS).

Forecasting yields with time series data

Phase three of this project represents the culmination of the current author's research, and it combines work in ML, DL, data synthesis, DA with pre-training, and time series forecasting. Most importantly, we propose an ML-based technique that is novel for forecasting crop yield time series data. We propose univariate and multivariate versions of our technique and compare them with traditional forecasting models like auto-regressive integrated with moving average (ARIMA) and seasonal ARIMA with exogenous variables (SARIMAX). We show that our technique produces better results, with sMAPE scores as low as 9.81% and MAEs as low as 0.14 tons per acre, for example. We also propose a novel data synthesizer called Scale Invariant Tabular Synthesizer (SITS) which outperforms the DL models we applied in phase two. Chapter 5 details our novel

forecasting technique, explains how SITS works, and presents our results compared to those baselines.

Alfalfa is typically harvested in the warmer months, rarely earlier than May or later than October, and most commonly in June, July, August, and September; therefore, alfalfa is seasonal, and its resulting time series data feature inherent seasonality. Stationarity is a term which, in the context of time series, means that any two spans of equal time observed in the time series will have the same data distribution if they are stationary, roughly speaking. Seasonal data is inherently non-stationary, as its peaks and valleys over time mean two arbitrary spans of equal width will likely not have the same distribution (Zolghadr-Asli). Since alfalfa data is seasonal, it is not stationary. While training ML models on past years' data and using them to predict a future target year works fairly well in this authors' experience, we found that enforcing stationarity on our time series data improved performance significantly. Therefore, the forecasting technique we propose enforces stationarity in alfalfa time series data for training ML models, and we have not seen this approach elsewhere in the precision agriculture literature. However, enforcing stationarity on this seasonal data was inspired by the success of the seasonal ARIMA (SARIMA) model, which usually also enforces stationarity (ArunKumar). As the alfalfa growing season predominately occurs from June to September, this work focuses on those warmer months, and thus trains and tests on four-month and sometimes three-month years, omitting the other zero-yield months from the time series.

The SITS synthesizer is also explained in detail in Chapter 5, and it is a simple algorithm based on standard deviations. Briefly, this algorithm accepts labeled table data as input, along with parameters for how many output records the user wants to generate plus the maximum number of standard deviations from the average to be generated. The algorithm then, for each label, for each feature, calculates the average and standard deviation, then generates random values within the

specified number of standard deviations for that feature. The scale invariant aspect is that however large the desired output dataset is, it will maintain the same proportions of class labels as the original input dataset.

## Primary Contributions

This dissertation represents a significant contribution to the field of computer science in the domain of ML in precision agriculture. There are at least four main contributions. First, this author knows of no ML-based crop forecasting technique that produces better results than our own, and no other work that presents time plots of true versus forecast yields over several time points; furthermore, our technique beats firmly established and popular methods, so it is a significant contribution. Second, the SITS data synthesizer is useful in training more accurate ML estimators of historical and present crop yield data than established synthesizers, making it useful when training models on data from one locality to predict yields in another. This can be useful in helping identify data-rich regions that might train strong models for predicting data-scarce target regions, and doing this is integral to our third contribution, the SNLT pipeline. SITS with pre-training is a form of DA, and it also improves the performance of ML-based forecasting, leading to forecasting with DA (ForDA). Fourth, we propose the PYCS application and describe potential use cases.

## Dissertation Organization

This manuscript-style dissertation is organized into 6 chapters as follows: Chapter 1 introduces the current project from a high altitude, states the problem and major contributions, and outlines the publications that comprise this dissertation and how they tie together to make a cohesive body of research; Chapter 2 is a literature review of related work in the field of predicting crop yields using

ML and related time series research; Chapter 3 compares traditional ML models trained on past and present weather at the task of estimating past and current alfalfa yields as continuous values; Chapter 4 describes how we train models on one location to classify yields in another when data is scarce, and it explains how we used generative ML models to synthesize data to train more accurate estimators; Chapter 5 details the culmination and focus of this research, which is forecasting future yields based on historical weather and yield data organized as time series, and it also describes our own novel data synthesis technique that beats established methods in estimating tabular yields not aggregated as time series; Chapter 6 proposes a preliminary graphical user interface (GUI) for the PYCS application and demonstrates some of its use cases and results; finally, Chapter 7 concludes the dissertation, summarizes the key points and contributions, and reiterates how these publications connect to create a strong, cohesive dissertation.

# CHAPTER 2

## LITERATURE REVIEW OF PREDICTING CROP YIELDS WITH ML

The research most closely related to that in the current dissertation is C. Whitmire's master's thesis, written under the advisement of K. Rasheed et al. at the University of Georgia (C. D. Whitmire). That work collected 771 records of alfalfa biomass yield data from the U.S. states of Georgia (GA) and Kentucky (KY) and aggregated it with weather data from those time periods when the crops were grown and harvested. The weather data for this previous work came from weather monitoring stations run by the National Oceanic and Atmospheric Association (NOAA). That work's experiments with feature selection techniques showed the most salient features to be Julian day of harvest, temperature, number of days since the crop was sown, and cumulative rainfall and solar radiation since last harvest (C. V. Whitmire). They combined the KY and GA data into one dataset, which they split into train, test, and validation sets to train a neural network (NN), random forest (RF), decision tree regressor (DT), Bayesian Ridge Regression (BRR), linear regression (LR), k-nearest neighbors (KNN), and support vector machine (SVM) models to predict yields in the test set. They applied 10-fold cross-validation and parameter grid searches to choose the optimum models. Overall, those results are quite promising, and they indicate that ML models do a great job of estimating historical and current yield data based on historical and current weather and climate data, with R and $R^2$ scores near or above 0.90.

Vance et al. expand this work by using the selected features from previous work by Whitmire et al. to focus on comparing the models themselves, and by expanding the alfalfa and

weather data to more U.S. states. Again, the resulting metrics were very promising and demonstrated that increasing the size of the dataset by adding more locations will drive up the performance of the models. Somewhat similar crop prediction research by You et al. at Stanford University (You J) also produced very good results with mean average percent errors (MAPEs) below 3%; however, those authors collected their own hyperspectral image data via sophisticated methods involving remote sensors, so they were able to take advantage of deep neural networks (DNNs) and convolutional neural networks (CNNs).

Overall, related literature demonstrates that using ML models trained on weather and yield data to estimate yield data in the same location has been attempted with limited success on which we can improve. This led the current author to wonder whether using data from one location to estimate another, where data are scarce, could also work. This was initially envisioned as a form of trivial domain adaptation (TDA), but as TDA lacks any model pretraining or data augmentation, the current author simply calls this non-local training, which is inspired by domain adaptation (DA) and provides a steppingstone toward it.

DA is a form of transfer learning (TL), and both DA and TL typically involve pre-training models for a task where the training data, called the source data, differs from the test data, called the target data. In TL, the feature space of the source data is different from that of the target (Torrey). In DA, the feature spaces may be the same, but the data distributions are different. However, there exists much in the literature that claims to be DA, though it can be very difficult to identify what commonalities they all share that make them proper DA (Jiang). Frequently in the literature, DA involves adding new knowledge from the target domain to models already trained on the source domain, and this is closely related to pre-training (Mishra) (Poerner) (Shen). The source task may be different from the target task. Also, the source data or the weights in the model

may be mathematically augmented according to some specific algorithm to make it a strong predictor for the target data (Jiang). In the current work, disparate geographic regions may differ significantly in their feature distributions. For example, normal temperature ranges and levels of rainfall vary. However, data from those disparate regions may benefit model pre-training. One general area of research where DA has shown notable promise is in sequence labeling (Daumé III). Often in the literature, DA and image classification are combined to predict crop yields. However, the current work avoids the expense of cameras and related equipment by instead collectively taking advantage of decades of isolated alfalfa research projects (Bellocchio). We make use of nationally abundant yield data to help solve the problem of locally scarce yield data by integrating DA without relying on image collection and analysis techniques.

As data scarcity is one of the core problems with successfully predicting general crop yields using ML, much of the related literature focuses on techniques designed to curate larger datasets. For example, You, Xue and Su at Northwest A&F University in China demonstrate the benefits of remote sensor (RS) technology using image data they collected from cameras aboard unmanned aerial vehicles (UAVs) (You J). Xue's surveys more than 100 vegetation indices (VIs) in his exhaustive literature review. VI's are collections of algorithms that do not share a unifying mathematical foundation but are adapted to measure significant characteristics involved in predicting crop yields (Xue).

The current work shares much in common with research by Cunha et al. at IBM Research in forecasting soybean and maize yields with a focus on deep learning (DL) (Cunha). That work employs a simple approach, like ours, where they manually aggregate yield data instead of developing VIs or complex RS and UAV pipelines. That work also focuses on forecasting future yields, and it takes advantage of large datasets in excess of 50,000 samples, facilitating the use of

deep neural networks (DNNs). Despite this, results reported in Chapter 3 show better $R^2$ scores in the 0.8 to 0.9+ range, while Cunha et al. report lower $R^2$ scores in the .55 to .75 range (Cunha).

Overall, the literature details many sophisticated and high-tech methods involving remote sensors, UAVs, computer vision, and image processing to develop vegetation indices to measure crop health (C. V. Whitmire) (J. R. Vance). The current research diverges from this trend in that our data come from publicly available reports, and we use simple daily weather information. Baral et al. at Kansas State University recently published a paper focused on rain gaps and their effect on alfalfa yields in multiple disparate regions in the U.S. They aggregated alfalfa yield data with historical weather reports for 10 states, and like the current work, they relied on university variety trial reports for their yield data (Baral). Like the current research, Baral et al. work only with data from rainfed alfalfa, which includes data from South Dakota (SD) State University and Ohio (OH) State University, so that work highlights a helpful resource for rainfed alfalfa growing locations. Another notable aspect of that research is that they work with virtually all purely rainfed alfalfa data available from land-grant universities in the U.S. They also measure alfalfa yield over growing degree days (GDD), meaning they exclude days when a temperature threshold alfalfa requires to grow was not reached (Baral). The overall goal of that work was to analyze the cost in alfalfa of dry periods (rain gaps) using frontier function analysis and boundary function analysis. That work applied conditional inference trees (CITs) to select weather features that affect crop yields most. However, they do not use ML to forecast or estimate alfalfa yields (Baral).

The current work explores generative adversarial networks (GANs), which are DL models that have grown in popularity preceding this research. GAN training involves simulating a minimax game to arrive at the Nash equilibrium, and this concept led to a breakthrough in the field of economics during the 20th century (Myerson). The GAN leverages the same game theory to DL

training, creating a competition between two entities where each motivates the other to improve. One of these, the generator, simulates data that resembles the actual training data, while the other, the discriminator, attempts to determine which is the actual versus the simulated data. This results in better guesses from the discriminator, which leads to better counterfeiting by the generator, motivating the discriminator's further improvement with guessing. This process closely resembles a minimax game, where two players maximize their odds of winning by minimizing the opponent's odds (Goodfellow). GANs have especially shown promise in computer vision, but they also have uses related to data privacy. For example, differentially private GAN (DPGAN) augments data to break connections between private data and real-life individuals (Xie). A GAN is employed to generate private health record information in medical GAN (MedGAN) (Choi). Another model called TableGAN (Park) synthesizes sanitized versions of private tabular data. Those three works demonstrate a GAN's ability to generate data that is statistically similar to the training data while introducing enough noise that identifying connections between actual individuals and the data are broken, protecting individuals' privacy (Xie). MedGAN combines a GAN with an autoencoder to bring differential privacy to health records, where anonymity is often desired, and where patients' willingness to share potentially crucially helpful data may be inhibited by privacy concerns (Choi).

CTGAN applies a conditional generator combined with training-by-sampling to mitigate the class imbalance within discrete columns. CTGAN normalizes each column in a dataset with "mode-specific normalization" and fully-connected networks (Xu).

Variational autoencoders (VAEs) used to synthesize tabular data are another approach we explore. VAEs, also generative neural networks, have been featured prominently in the literature and shown promising results in recent years (Xu). The current work explores the tabular variational

autoencoder (TVAE) proposed by Xu et al. in (Xu), where they adapt a vanilla VAE to mitigate issues specific to tabular data synthesis.

Another model that is integral to the current research is extreme gradient boosting (XGBoost or XGB). Boosting typically creates a strong learner by combining an ensemble of weak learners. Adaboost is one popular algorithm that typically boosts random forest (RF) or decision tree (DT) models trained and improved upon iteratively, and incorrect predictions are assigned greater weight in further iterations up to a predefined depth or maximum accuracy (Freund). In XGBoost, DTs iteratively minimize prediction errors using gradient descent, resulting in one optimized strong learner. This leads to extremely fast training times, so XGBoost can train on very large datasets very quickly (Chen).

Jin et al. introduce a domain adaptation forecaster (DAF) that claims promising results, but they report a metric that is non-standard, normalized deviation (ND), so comparisons between DAF and other forecasters are hard to draw. More common in time series literature are metrics like sMAPE , R, and MAE such as are found in the current work. DAF employs data synthesis and ML with attention sharing. That work makes predictions in four benchmark domains: household electricity consumption, traffic patterns, grocery sales, and visits to Wikipedia pages. Jin et. synthesize data, use that to train models, then they synthesis and non-synthesis based results, like the current work. The DAF attention module “captures domain-dependent patterns by context matching using domain-invariant queries and keys,” meaning values in the source domain are adapted to the target by applying weights to keys instead of values. (Jin).

Bashar et al. report promising classification accuracy with identifying hate speech on social media, but they classify only past and present data rather than forecasting the future as the current work and Jin’s, do. Like us, they propose a DA technique involving pretraining of models

including XGBoost. That research seeks to identify hate speech on social media directed at the East Asian community, and they claim this has become more difficult to detect since COVID-19 altered sentiments and increased bias against that community. They seek to use DA to adapt social media communications from pre-pandemic years, making such data more useful in classifying current-day language. They introduce a progressive transfer learning (TL) technique that reports success with integrating knowledge from multiple datasets, then using those models to classify veiled or hard-to-detect hate speech (Bashar).

DA techniques based on data synthesis are common and promising in much of the related literature. However, DA and data synthesis in image classification and segmentation tasks dominate such literature, and those tasks diverge substantially from those in our research. Ultimately, we use DA for regression or predicting continuous values, rather than classification. Peng et al. offer the Visual Domain Adaptation (VisDA) dataset and challenge in their 2017 paper. VisDA offers a standard by which researchers can compare DA techniques. It consists of synthetic images for the intended use as source data for DA models, and it also has corresponding non-synthetic photographs as target data. The images include everyday objects like trains, airplanes, cars, horses, and others. Any researcher interested in comparing their technique to related research may refer to the baseline results provided by Peng et al. for classification and segmentation using VisDA data. That work helps motivate the current author to use DA for alfalfa biomass yield forecasting (Peng, Visda: The visual domain adaptation challenge). Similarly, Bak et al. offer a dataset of synthetic images they call the Synthetic person Re-Identification (SyRI) dataset. Intended as source data for experiments with DA, SyRI addresses the task of re-identifying humans captured in images as the same humans in images from other sources. This work notes that variations among cameras, lenses, lighting conditions, angles, etc. present challenges that

complicate facial recognition and identification (Bak). One field that applies DA is known as disentanglement analysis. That applies ML to identify features present in one domain from features common to both domains, the source and the target. Cao et al. use DA and data synthesis to classify English-language characters in common benchmark datasets. They use those results to further generate labelled target data and improve DA results (Cao). Another paper that uses DA in computer vision for classification and segmentation tasks is Choudhary et al.'s 2020 review of state-of-the-art (SOTA) research into DA's utility in medical imaging. That work features a chapter focused on reviewing work that applies data synthesis for DA with medical imaging (Choudhary). Partially motivated by Cao's work (Cao), Li et al. explore DA with data synthesis in the task of identifying animal poses. They claim high accuracy in classifying animal body parts such as chins, legs, knees, hooves, and others for several commonly identifiable animals like sheep, horses, and tigers. Although they claim significant improvement over other work in this domain, their mean accuracy was 57.23% when classifying unseen Zebra body parts, and that may be too low be practical (Li). This brief literature survey of related work using DA with synthesis confirms that this is a promising approach, explored by many, though usually applied to computer vision techniques and classifying images. However, DA with synthesis as applied in the current work, for the task of crop yield forecasting based on tabular weather data, leaves many open research opportunities.

In their 2022 paper exploring classification of alfalfa biomass yields, Vance et al. (Vance, Rasheed and Missaoui) synthesized training datasets using two established models, the conditional tabular generative adversarial network (CTGAN) and tabular variational autoencoder (TVAE) introduced by Park et al. (Park). Those synthesizers produced promising results that motivated us

to continue exploring such techniques as we segued into forecasting, and they eventually inspired us to create our own data synthesis algorithm, which materialized as the SITS synthesizer.

Related work by Kastens et al. presents promising results in their work with crop yield time series forecasting using computer vision (CV) and masking. However, they limit their experiments to linear regression models, and they stop short of comparing their results to the ARIMA family and other baseline forecasting models. They report their results in mean absolute errors (MAEs) of metric tons per acre for corn, wheat, and soybean. As the current work reports tons per acre of alfalfa, their results are in grain instead of biomass, so theirs is hard to compare to ours. Kasten et al. also need at least 11 years of training data to produce a strong forecasting model, while the current work requires only six years of data to train models with promising sMAPE scores. Furthermore, that research serves as further evidence that CV dominates the crop yield forecasting research, highlighting the novelty in the simplicity of the current work (Kastens). Choudhury and Jones show promising results with applying Auto Regressive (AR) models to predict maize yields in Ghana. However, the main table of interest in their paper seems to not be referred to or explained in the text of the paper, so it is difficult to understand. For example, they appear to claim mean squared errors (MSEs) as low as .03 metric tons per acre when comparing statistical forecasting techniques, but the details of what that means are unclear. That paper's discussion section also emphasizes coefficient of determination ($R^2$) as a metric, but related forecasting literature and the current work reveal that coefficients are less interesting when predicting multiple future time points at once. Choudhury and Jones do not explore models beyond AR and variations of exponential smoothing, and they investigate only univariate models, unlike the current research (Choudhury). In their 2016 paper on forecasting winter-wheat crop yields, Bose et al. introduce a forecasting model they call spiking neural network (SNN). In their words, the SNN "encodes

temporal information by transforming input data into trains of spikes that represent time-sensitive events." These spikes are represented as binary values. While they do not offer a univariate version, and they do not compare their SNN to the ARIMA family, they claim high accuracy when compared with linear regression (LR), k-nearest neighbors (KNN), and support vector regression, (SVR). Furthermore, while the current work typically forecasts yields nine months into the future and includes 3 to 4 yields, the SNN predicts one yield per year, six weeks into the future. They report Rs that are calculated over 14 years of predictions, as opposed to R scores in the current work, which are calculated using the three or four predictions in a year and their corresponding truth (Bose). In his 2019 paper, Pavlyshenko reported promising results in predicting future sales, focusing on Rossmann stores in Europe. They introduced a stacking technique that scores higher than ARIMA and other ML models, but they do not explore forecasting in other domains like biomass yields (Pavlyshenko).

In the current work, we compare our novel ML-based forecaster to several traditional statistical forecasting models in the family of autoregressive integrated moving average (ARIMA). Specifically, we use the following baselines: ARIMA, seasonal ARIMA (SARIMA), and SARIMA with exogenous variables (SARIMAX). Vishwas and Patel offer definitions of the widely known ARIMA family of models in their book on time series. Very simply put, ARIMA considers values at previous time points, called lags, and uses those to forecast future values. That is the autoregressive (AR) step in a nutshell. MA stands for the moving average step. MA considers only error lags, the differences between predicted and true values, to predict future errors. In the integration (I) step, differences between related time points are calculated. This mitigates the effects of seasonality and makes the time series stationary. AR and MA typically perform better when the time series is stationary. Roughly put, a time series is stationary if any two equal size

windows of time in the series share the same data distribution. Seasonal data such as crop yields are typically not considered stationary. ARIMA can enforce stationarity by integrating the differences between values at corresponding steps in a time series rather than the actual values, which is the I part. ARIMA accepts three input values, as in ARIMA(1, 0, 0). The first 1 is the *p* parameter, or the number of lags considered for AR. Next, the, 0 is the *d* parameter, or the number of times differences are taken. The last 0 is the *q* parameter, or the number of error lags considered for MA. Seasonal ARIMA (SARIMA) takes seasonality into account by accepting four more seasonal parameters, such as SARIMA(p, d, q)(P, D, Q, m) or SARIMA(1,0,0)(5,1,10,4). In SARIMA, the *P* parameter represents the number of seasonal AR lags, *D* represents the number of times seasonal differences are taken, the *Q* parameter represents the number of seasonal MA lags, and the *m* parameter represents the number of time points that are contained in one season. SARIMA with exogenous variables (SARIMAX) accepts an additional parameter, which is a vector of exogenous variables. SARIMAX identifies cross-correlations between the target and the exogenous variables, and they are used to increase forecast accuracy. Vishwas and Patel offer a precise mathematical definition of the ARIMA family in their book on time series with the Python programming language (B V Vishwas).

# CHAPTER 3

# COMPARING MACHINE LEARNING TECHNIQUES FOR ALFALFA BIOMASS YIELD PREDICTION [1]

[1] Vance, J., Rasheed, K., Missaoui, A., Maier, F., Adkins, C., & Whitmire, C. 2021. Proceedings from CSCE, arXiv preprint arXiv:2210.11226.

Abstract

The alfalfa crop is globally important as livestock feed, so highly efficient planting and harvesting could benefit many industries, especially as the global climate changes and traditional methods become less accurate. Recent work using machine learning (ML) to predict yields for alfalfa and other crops has shown promise. Previous efforts used remote sensing, weather, planting, and soil data to train machine learning models for yield prediction. However, while remote sensing works well, the models require large amounts of data and cannot make predictions until the harvesting season begins. Using weather and planting data from alfalfa variety trials in Kentucky and Georgia, our previous work compared feature selection techniques to find the best technique and best feature set. In this work, we trained a variety of machine learning models, using cross-validation for hyperparameter optimization, to predict biomass yields, and we showed better accuracy than similar work that employed more complex techniques. Our best individual model was a random forest with a mean absolute error of 0.081 tons/acre and $R^2$ of 0.941. Next, we expanded this dataset to include Wisconsin and Mississippi, and we repeated our experiments, obtaining a higher best $R^2$ of 0.982 with a regression tree. We then isolated our testing datasets by state to explore this problem's eligibility for domain adaptation (DA), as we trained on multiple source states and tested on one target state. This Trivial DA (TDA) approach leaves plenty of room for improvement through exploring more complex DA techniques in forthcoming work.

Introduction

With the intent of directing world leaders toward solving some of the world's biggest problems, the United Nations recently developed a collection of 17 goals and 169 targets. The hope is that the world will reach these goals by the year 2030 (United Nations). However, it is the opinion of the Copenhagen Consensus Center (CCC), a think tank, that appropriately prioritizing these goals will increase the likelihood that the world will reach them ( Copenhagen Consensus Center). The CCC has performed a cost-benefit analysis on all these targets and ranked them accordingly. One of their findings was that increasing research and development into maximizing global crop yields would be one of the most cost-effective ways of achieving the UN's goals (Mark W Rosegrant). Specifically, every $1 spent on this kind of R&D would result in $34 worth of benefit worldwide (Lomborg).

One possible way to increase yields is to improve agricultural planning, which would help ensure that there are sufficient yields of key crops. At the start of every season, agricultural planners need to estimate the yields of different agricultural plans (Frausto-Solis J). Often, farmers rely on their own personal experiences of history to predict what their yields will be, but this can result in limited accuracy (G.). Given that crop yields vary spatially and temporally, and are sensitive to varying conditions like weather, other prediction methods should be investigated.

The USDA, with its National Agricultural Statistics Service branch, makes monthly forecasts of crop yields in the United States. It does this by conducting two surveys, a farm operator survey and an objective survey. The farm operator survey is done by calling farmers at random and asking them what they think their predicted yield for the next month will be. The objective survey involves an investigator going out and surveying random fields and recording data on the output of those fields. The findings of these surveys are compared to previous historical data to

confirm that the findings are consistent with previous harvests with similar conditions. The final predicted yields then come from the results of these surveys (National Agricultural Statistics Center. Crop production) (DM.). The findings of this methodology, when compared to the ground truth, have had very low errors (You J) (National Agricultural Statistics Center. Crop production). However, this process is very resource intensive. The farm operator survey is done primarily over the phone, and the objective survey requires measurements to be taken in person at hundreds of farms every month (National Agricultural Statistics Center. Crop production) (DM.).

An alternative approach is to use remote sensing (RS) data. RS techniques use images achieved primarily from aircraft of satellites, and these images record spectral, spatial, and temporal information (Chlingaryan). Mathematical operations can be performed on these images to form vegetation indices (VIs), which can be used as inputs into machine learning algorithms (Xue J). Recent work has been done to use VIs to predict crop yield. You et al., had great success at predicting county-level soybean yield in the United States using remote sensing data as input for a convolutional neural network (CNN) and a long short-term memory (LSTM) model, both with a Gaussian Process component (You J). Panda, Ames, & Panigrahi used several different VIs as an input to a neural network (NN) to predict corn yield (Panda). Johnson did something similar but used regression trees to predict both corn and soybean yield (DM.). However, despite these successes, there are difficulties with building machine learning models based on remote sensing data. This is because using remote sensing data depends on the processing of large amounts of data across different platforms (Chlingaryan). These models also cannot make a prediction unless there are images available for input, which means that this model cannot begin making predictions until the growing season has started (Cunha). Xue and Su also compared over one hundred different vegetation indices and found that no VI is universally better than the others. Each is more

suitable to certain situations, and each has their own limitations (Xue). This means that it may be difficult to know the optimal VI for each case.

Weather, spatial, and soil features have also been used to train machine learning models to predict crop yield (Gonzalez-Sanchez) (Ayoubi) (J. R. Jeong). These kinds of data also require less processing than remote sensing data and can be used to make predictions before the season starts. They also have the potential to use weather forecasting results to make predictions before the season begins, making it more convenient for planning purposes than using remote sensing data. Similarly, the current paper uses weather and planting data to develop a variety of machine learning models and compares the results.

The current work builds further on the previous work by Whitmire et al. by comparing biomass yield prediction experiment results among various ML models, as we explore elements of domain adaptation (DA) to further improve our application pipeline (C. V. Whitmire). We used data from different source regions to train models which then predict crop yields in a specific target region. To this end, we achieve three major contributions:

- Provide results from ML experiments comparing various models, showing our weather-based ML approach to be promising
- Broadly expand the alfalfa ML training dataset, adding curated crop yield and weather data from three additional states
- Discover positive results for seemingly disparate sources and targets, such as source: Wisconsin, target: Georgia

In Section II, we discuss related research and topics relevant to this project. Section III provides details about our methods as well as the collection and characteristics of the data.

Experiment setups and results are discussed in Section IV, Section V contains conclusions, and Section VI outlines directions of future work.

## Related Work

This project began as C. Whitmire's master's thesis, with help from H. Rasheed, under the direction of K. Rasheed et al. at the University of Georgia (C. D. Whitmire). They aggregated 771 records of alfalfa biomass yield data from Georgia and Kentucky, and they correlated this with weather data from the same time periods; specific weather features include the same ones in the current work, harvested from the NOAA as well as other sources. By experimenting with feature selection techniques, they found that the most salient features were Julian day of harvest, temperature, number of days since the crop was sown, and cumulative rainfall and solar radiation since last harvest (C. V. Whitmire). They split the data into train, test, and validation sets to train a neural network (NN), random forest (RF), regression tree (RT), Bayesian Ridge Regression, linear regression, k-nearest neighbors (KNN), and support vector machine (SVM) models to predict yields in the test set. Using k-fold cross-validation, they performed grid searches to determine which model configurations achieved the highest accuracies (C. V. Whitmire). Our results approach the accuracy of similar work by You et al. at Stanford University (You J), even though [11]'s datasets are much larger and collected by more sophisticated methods involving remote sensors, which enables the application of much deeper and more complex neural networks. Overall, our results were promising enough to motivate us to explore domain adaptation (DA), where models are trained using one mathematically augmented source dataset to predict a distinct target dataset. We explore DA by attempting a trivial DA (TDA) approach, where we train on

source data from our original dataset plus three new U.S. States, and we set the target dataset as GA.

As the core problem with successfully predicting crop yields using machine learning is scarcity of data, much of the related works focus on techniques designed to accumulate larger datasets. Like You, Xue and Su at Northwest A&F University in China highlight the benefits of remote sensor (RS) technology, such as image data collected by unmanned aerial vehicles (AEVs) (Xue J). Xue's work reviews over 100 vegetation indices (VIs), which are assorted algorithms with no mathematical unifying basis that are customized to measure important characteristics used in crop prediction and related applications.

Our work most closely resembles research applying deep learning (DL) to yield forecasting soybean and maize by Oliveira et al. at IBM Research (Cunha). Like ours, that work applies a very simple method of manual yield data collection instead of developing VIs or using RS technology. On the other hand, their work forecasts future yields, while the current work only predicts known yields. [15]'s abundance of over 50,000 datapoints opens that work to the use of a deep neural network (DNN), which we would like to try in future work. However, the current work achieves better $R^2$ scores that fall comfortably in the 0.8 to 0.9+ range, while [15] reports $R^2$ scores ranging from .55 to .75 (Cunha) (C. D. Whitmire).

*Transfer Learning and Domain Adaptation*

Transfer learning is an area of research focused on improving learning in a new task through the transfer of knowledge from a related task that has already been learned. This is in contrast with most machine learning algorithms which are designed to address single tasks (Torrey). While useful in many research problems, transfer learning as a whole does not directly apply to this work.

The task of predicting alfalfa yield remains the same; what changes is the region that data comes from. These different regions may have significant differences in the underlying distributions of feature values, with varying temperature ranges and standard levels of rainfall for example.

Thus, a subcategory of transfer learning known as domain adaptation (DA) is more potentially applicable to our research. DA uses data from a source with one distribution to predict target values in a test domain with a different distribution (Jiang). A large amount of research has utilized DA in the area of sequence labeling (Daumé III). In the context of agriculture, image classification is a popular technique for studying aspects such as crop yield (Bellocchio). But such methods are very expensive and time consuming, are not feasible in all locations, and do not benefit from existing data collection that has been carried out over the years. Our work aims to incorporate DA in order to coalesce such widespread numerical data and achieve meaningful prediction accuracy without relying on expensive image collection and analysis techniques.

## Approach

We used the Python programming language throughout this research (Foundation). Specifically, Python as provided within the Anaconda environment was used (Distribution). The following packages were used: Pandas for data cleaning and preparation (McKinney), matplotlib (Hunter) and seaborn (Waskom) for visualizations, sci-kit learn to make and evaluate the machine learning models (Pedregosa), and finally, numpy for general mathematical operations (Oliphant) (Van Der Walt).

The features used in training our machine learning models were the Julian day of the harvest, the number of days between the harvest and the sown date of the crop, the cumulative

solar radiation since the previous harvest, temperature, and the cumulative rainfall since the last harvest. The cumulative solar radiation and rainfall values were found by summing daily values.

All the data sources for the previous and current work are presented in Appendix A. Alfalfa harvest data was obtained from variety trials conducted by the University of Georgia (UGA) and University of Kentucky (UKY) for our first round of experiments. We added data from Pennsylvania (PA), Wisconsin (WI), and Mississippi (MS) for the second round. These reports detail the yield in tons per acre of multiple varieties of alfalfa. UGA's data came from Athens and Tifton, GA from the years 2008 to 2010 where harvests occur from April to December. UKY's trials occur in Lexington, KY ranging from 2013 to 2018 and contain data from the months of May through September. These variety trials reported multiple cut dates per year.

We carefully curated daily weather data for each location. Data for Tifton and Watkinsville, which is about 13 miles from Athens, GA, was retrieved from the Georgia Automated environmental network. Similar data was found for Versailles, which is nearby Lexington, KY, from the National Oceanic and Atmospheric Administration (NOAA).

Also, all the data points that had harvest dates with the same year as the sown date were filtered out. Similarly, the first harvest of every season was removed because the amount of time since the previous harvest would be much larger for this harvest relative to subsequent harvests. After this cleaning process, we ended up with 770 datapoints total.

Before training the models, all of the features were normalized according to the formula

$$x_{new} = x^{\underline{old}} - x_{SDev}{}^{x}{}_{mean}$$

where $x_{old}$ was the original value of the feature, $x_{mean}$ is the average value of the features, and $x_{SDev}$ is the standard deviation of the values for that feature.

Before training the models in our previous work, the data was shuffled and split into ten folds to be used for 10-fold cross validation. For each fold, a machine learning model was initialized. This means that ten models were made for each method, one model for each fold. Then, within this outer fold, a grid search (Appendix B) with 5-fold cross validation was done to find the hyperparameters for the model that most minimized the mean absolute error. Once the hyperparameters were found, the machine learning model was trained on the training set and was evaluated against the testing set. We calculated the mean absolute error (MAE), R value, and $R^2$ value. The average errors, percent error, R, and $R^2$ value over the ten iterations was found and recorded, and the results of the best model were also recorded along with their standard deviations (C. D. Whitmire).

This process was done to train and evaluate the following methods: regression tree, random forest regression, k-nearest neighbors, support vector machines, neural networks, Bayesian Ridge regression, and linear regression. Once the results for each method were obtained, an unpaired two-tailed t test was used to find the p-value between the average $R^2$ values of each method (C. D. Whitmire). As shown in Table 3.1, we achieved our highest $R^2$ score of 0.941 with our Random Forest model.

**Table 3.1:** Estimator Performance: KY & GA. Our best $R^2$ score overall, training on KY and GA data combined, was 0.941 with RF; RT, KNN, and SVM showed similar accuracies.

| **Model** | **$R^2$** | **MAE (tons/acre)** |
|---|---|---|
| DT | 0.928 | 0.091 |
| RF | **0.941** | 0.081 |
| KNN | 0.936 | 0.091 |
| SVM | 0.917 | 0.094 |
| BRR | 0.777 | 0.147 |
| LR | 0.723 | 0.160 |

We collected data from Mississippi State University (MSU), Penn-State University, and the University of Wisconsin-Madison (UW-Madison). Our MSU data come from Starkville, Poplarville, Holly Springs, and Newton, MS between March and December from 2012 to 2016. Our Penn-State data come from Rock Springs and Landisville, PA annually from 2008 to 2019. Our UW-Madison data come from Lancaster, Marshfield, and Arlington, WS from 2011 to 2016. Some variety trials report annual biomass yields, while others report multiple cuts per year, and all report sown dates. Therefore, we decided to annualize all our data for the current approach, since the reverse is not feasible. We also referred to the NOAA for our PA, WI, and MS weather data, matching to the exact city where possible, and sometimes to nearby localities with more complete data, as in the previous approach. While we made every effort to compile the most accurate data possible for this work, there were cases where we had to make assumptions about some missing weather data from NOAA to compile our dataset.

We selected the following ML models implemented in SciKit-Learn: random forest (RF), k-nearest neighbors (KNN), Bayesian ridge regression (BRR), logistic regression (LR), support vector machine (SVM), and decision tree (DT). During the training phase, we run our models through a function to select the best hyperparameters using k-fold cross-validation, tuning each model based on its training data. We used these optimized models as our predictors for each experiment with each source dataset. As shown in Table 3.2, our DT model achieved the highest $R^2$ score of 0.982, and RF achieved an $R^2$ of 0.981 after we added WI and MS data. KNN was a close third place with $R^2 = 0.979$. We detail the number of datapoints in each state for reference in Table 3.3.

**Table 3.2:** Estimator Performance: KY, GA, WI, & MS. Our best R2 score overall after adding WI and MS increased to 0.982.

| **Model** | **$R^2$** | **MAE (tons/acre)** |
|---|---|---|
| DT | 0.982 | 0.240 |
| RF | **0.981** | 0.247 |
| KNN | 0.979 | 0.203 |
| SVM | 0.900 | 0.392 |
| BRR | 0.391 | 1.253 |
| LR | 0.416 | 1.248 |

**Table 3.3:** Datasets per state with number of datapoints

| **State** | **No. of Datapoints** |
|---|---|
| PA | 577 |
| WI | 338 |
| KY | 38 |
| MS | 184 |
| GA | 84 |

Results

In our first round of experiments, taking the traditional (non-DA) ML approach and training and testing on a combination of KY and GA data, we obtained our highest $R^2$ of 0.941 from the RF model, while our KNN and DT scores followed closely behind. In the second round, we repeated these experiments adding PA, WI, and MS data. While these results were also positive, we obtained better results when we removed PA, so we present the results from only KY, GA, MS, and WI in Table 3.2, which shows our highest $R^2$ increased to 0.982 training on the expanded dataset, using the DT model. One interesting surprise was that we saw relatively high $R^2$ scores training on combinations that include WI, whose weather shares less in common with GA weather than MS or KY; WI experiences much lower winter temperatures and much lower soil moisture, among other differences.

Next, we ran our DA-inspired ML experiments, where we use a trivial form of DA (TDA) in which we train on data from selected source states and test on data from one specific target state. As opposed to a true DA technique that mathematically augments the source data to make its distribution similar to the target data, TDA simply uses the data as-is. Overall, our TDA technique performed quite poorly, resulting in much lower accuracies than when we combine the states into one table, and both training and testing sets include multiple states. Though unsuccessful, TDA leaves plenty of room for improvement by using a true DA technique that augments the training data, and it shows us that simply training on specific source states and testing on one target state is not likely to yield good results in this domain.

## Conclusion

We demonstrated that the problem of training ML models on scarce crop yield datasets for one geographical region can be partially remedied by training those models on data aggregated from other regions, even when they have disparate climates. We were able to improve on $R^2$ scores obtained using traditional ML without DA, which is an important first step toward creating more sophisticated transfer learning and domain adaptation based pipelines, suggesting that further research into using these techniques is warranted.

In the interest of improving machine learning's utility in the crucial field of agriculture, we expanded on previous work that compared feature selection techniques to pick the best features for predicting alfalfa crop yields. Toward the goal of creating a universal dataset that we can share broadly, we added virtually exhaustive data for three new U.S. states, for the timeframes available in variety trial reports. With more datapoints, we were able to interpret data on annual basis, instead of per harvest, which may be more meaningful.

Such data is widely collected in variety trials run by various universities but not in any uniform, organized format and not combined with any relevant weather data. This leads to artificial scarcity of data as different local regions' data prove challenging to combine. To deal with this scarcity, some previous works have implemented expensive, highly-specialized techniques for collecting data, such as using advanced remote sensors at specific locations. Such methods are typically not as widely adopted and do not cover comparable time spans to standard variety trials. Inspired by the ability of domain adaptation to reduce sparsity of data, we used our curated sources to learn models in order to make predictions on the target region (the original GA dataset). This technique yielded poor results, indicating that some true DA technique that augments data distributions should be explored and would probably yield better results.

Future Work

We are currently planning three main directions for our next projects with predicting alfalfa biomass yields. First, we plan to implement a variety of more sophisticated DA techniques than the one we present here and compare those results. Second, we plan to develop a time series strategy that helps us forecast future yields based preceding ones, adjusting its predictions along the way. Third, we are interested in analyzing how the different varieties perform in different geographical locations. We are also interested in streamlining our data curating pipeline to reduce the manual labor, and intend to introduce stabilization metrics into this future work.

# CHAPTER 4

# DATA SYNTHESIS FOR ALFALFA BIOMASS YIELD ESTIMATION [2]

[2] Vance, J., Rasheed, K., Missaoui, A., & Maier, F. W. 2022. *AI*, 4(1), 1-15.
Reprinted here with permission of the publisher.

Abstract

Alfalfa is critical to global food security, and its data is abundant in the U.S. nationally, but often scarce locally, limiting the potential performance of machine learning (ML) models in predicting alfalfa biomass yields. Training ML models on local-only data results in very low estimation accuracy when the datasets are very small. Therefore, we explore synthesizing non-local data to estimate biomass yields labeled as high, medium, or low. One option to remedy scarce local data is to train models using non-local data; however, this only works about as well as using local data. Therefore, we propose a novel pipeline that trains models using data synthesized from non-local data to estimate local crop yields. Our pipeline, synthesized non-local training (SNLT pronounced like sunlight), achieves a gain of 42.9% accuracy over the best results from regular non-local and local training on our very small target dataset. This pipeline produced the highest accuracy of 85.7% with a decision tree classifier. From these results, we conclude that SNLT can be a useful tool in helping to estimate crop yields with ML. Furthermore, we propose a software application called Predict Your CropS (PYCS pronounced like Pisces) designed to help farmers and researchers estimate and predict crop yields based on pretrained models.

## Introduction

The alfalfa crop is an important livestock feed and is crucial to global food security. In previous work, we used climate data to estimate alfalfa biomass yields. We compared the accuracies of feature selection techniques and machine learning (ML) models for this task. We obtained promising results using local training data with $R^2$ values over 0.90, as we had access to rich

curated datasets from state university variety trials (C. V. Whitmire). However, since our team is developing a software application to aid real-world farmers, whose datasets may be much smaller, the current work addresses the problem of estimating yields for very small target datasets. We find that local training on very small target datasets results in very low accuracy, while non-local training on much larger datasets performs only about as well as local training. Our solution combines ideas inspired by (Hendrycks), which shows success using pretrained models and sparse datasets, with ideas inspired by (Makhzani) and (Goodfellow), which show the promise of deep learning generative models like the adversarial autoencoder (AAE) (Makhzani) and generative adversarial networks (GANs) (Goodfellow). We propose a novel pipeline where models are trained with data generated or synthesized by other deep learning (DL) models. In this pipeline, the synthesized training data are synthesized from non-local sources, and the resulting classifiers estimate local targets. We call this synthesized non-local training (SNLT pronounced like sunlight), and we show it consistently achieves better accuracy than both local and non-local training. We extend the work of Xu et al. (Xu) by using their conditional tabular GAN (CTGAN) and tabular variational autoencoder (TVAE) synthesizers in our pipeline. The highest accuracy we obtained from SNLT was 85.7% using a decision tree classifier (DT) with CTGAN. We also obtained good results using data synthesized by TVAE, especially when training extreme gradient boosting models (XGBoost), scoring as high as 75.0% and 70.0% classification accuracy. One long-term goal of this work is to develop a software application called Predict Your CropS (PYCS, pronounced like Pisces) (J. Vance, PYCS). PYCS is a research software application and what-if tool that farmers and others can use to build, train, and run ML models to potentially predict future crop yields based on climate or weather data. Though the current iteration of PYCS is not based on time series, and therefore does not forecast future yields or trends, that is the ultimate goal for

the application. However, the current iteration is a potentially useful what-if tool. For example, a farmer could design worst-case and best-case scenarios and input these hypothetical features into PYCS to create what-if predictions. This could help the farmer understand how large or small their yields might be, so they could prepare in advance to handle good, fair, and bad years appropriately and manage their resources wisely.

Alfalfa is so crucial to food security that many U.S. universities support initiatives to research this crop and publish data on the experimental growth, cultivation and harvesting of available varieties of alfalfa. These variety trial reports often include crop yield data for multiple cuts each year, contributing to the richness of the nationally available yield data. Very recent work by (Baral) highlights the relevance of alfalfa re-search, as their data come from the same variety trial reports as ours. They focus on the effects of rain on yield gaps and, like us, primarily look at rainfed rather than irrigated sites (Baral). The importance of alfalfa and this abundance of alfalfa data, along with recent strides forward in ML, motivate our team to apply ML to the problem of estimating and predicting crop yields, with a focus on the alfalfa crop. As in our team's previous work (C. V. Whitmire), we use aggregated climate and yield data detailed in Tables 4.1 and 4.2.

Mitigating the effects of climate change is another motivator behind this work. Evidence suggests that traditional approaches to forecasting crop yields are becoming less reliable as the Earth's climate becomes less predictable (Matouq) (Feleke) (Scher) (Yahya) (Schlenker) (Jeong, Kim and Lee) (Dhore, Byakude and Sonar) (Easterbrook). As the global effort to combat climate change continues, led in part by the United Nations (United Nations), we hope this work is a positive force toward these efforts.

This work is part of an ongoing project that explores the benefits of using ML to predict biomass yields for crops. We focus on the alfalfa crop, but this research could be expanded to other

crops. Our first paper concentrates on feature selection, as it presents results from a battery of tests of the following feature selection techniques: correlation-based feature selection (CFS), ReliefF method, and a wrapper method. The data for that paper is a curated mixture of alfalfa biomass yields from variety trials in Georgia (GA) and Kentucky (KY) combined with weather and soil moisture sourced from the National Oceanic and Atmospheric Association (NOAA) (C. V. Whitmire).

Our second paper focuses on comparing the ML models themselves, presenting R, $R^2$, and MAE scores for most of the same models in the current and previous works. That paper expands the dataset to include the U.S. states of Pennsylvania (PA), Wisconsin (WI), and Mississippi (MS). Those results show that larger datasets lead to higher accuracies when the states are all combined into one large dataset. That work also revealed that while data from multiple sites can be combined to train one model with good $R^2$ scores above 0.90, training a model with strictly non-local data to estimate yields in a separate locality did not produce usable results with regression models (J. R. Vance). This motivated us to reframe this as a classification problem in the current work, and the results are much better.

We reframed our regression problem from (J. R. Vance) as a classification problem with three tiers for yields: high, medium, and low. We swapped the regression models in our original application programmer interface (API) from (C. V. Whitmire) and (J. R. Vance) for analogous classification models from ScikitLearn (Pedregosa). For this work, as we are interested in estimating very small target datasets, we reduce the original target GA dataset in (C. V. Whitmire) to one alfalfa variety per year, and we annualize the cuts as detailed in Section 3, resulting in only seven records detailed in Table 4.1. We repeat this procedure for Ohio (OH) and South Dakota (SD), producing very small OH target datasets of 8 and 10 records and producing very small

datasets for one SD town, also of 8 and 10 records. Table 4.2 shows our annualized, one-variety per year target data from OH with its four features. OH's features are slightly different from GA's and KY's in that OH lacks soil moisture data. OH and SD follow the same schema. In Tables 4.1 and 4.2, class labels 0, 1, and 2 correspond to low, medium, and high yields, respectively.

Table 4.1. KY & GA Data Example. Shows the five features used to train all KY and GA models plus year harvested and class, assigned according to standard deviation and yield; these are the 7 records in the very small GA target dataset.

| **Year** | **Yield (tons/acre)** | **Class** | **Solar Radiation ($MJ/m^2$)** | **Total Rainfall (mm)** | **Avg Min Temp (°C)** | **Avg Max Temp (°C)** | **Avg Soil Moisture (%)** |
|---|---|---|---|---|---|---|---|
| 2009 | 2.40 | 0 | 3400.10 | 740.67 | 16.10 | 27.88 | 0.19 |
| 2008 | 3.33 | 1 | 3810.06 | 664.88 | 18.44 | 29.26 | 0.12 |
| 2008 | 3.35 | 1 | 3545.32 | 413.28 | 12.54 | 24.64 | 0.13 |
| 2008 | 5.20 | 1 | 4463.92 | 599.97 | 15.77 | 28.37 | 0.19 |
| 2009 | 5.92 | 1 | 5320.75 | 1323.90 | 15.69 | 26.19 | 0.14 |
| 2009 | 6.26 | 2 | 3915.63 | 925.56 | 16.31 | 27.22 | 0.13 |
| 2010 | 6.50 | 2 | 4092.75 | 847.82 | 16.28 | 28.33 | 0.13 |

Table 4.2. SD & OH Data Example. Shows the four features used to train all SD and OH models plus year harvested and class, assigned according to standard deviation and yield; these are the 10 records in one very small OH target dataset.

| Year | Annual Avg Yield (tons/acre) | Annual Avg Min Temp (°C) | Annual Avg Max Temp (°C) | Total Accumulated Rain (mm) | Total Accumulated Radiation ($W/m^2$) | Class | Year |
|---|---|---|---|---|---|---|---|
| 2019 | 1.36 | 7.56 | 17.82 | 7462.16 | 552,381.90 | 1 | 2019 |
| 2019 | 1.54 | 6.32 | 16.84 | 13,854.04 | 1,009,929.62 | 1 | 2019 |
| 2010 | 1.80 | 6.76 | 17.44 | 3764.83 | 561,106.56 | 2 | 2010 |
| 2010 | 1.24 | 5.88 | 16.98 | 9814.53 | 1,054,700.52 | 0 | 2010 |
| 2011 | 1.28 | 6.16 | 17.26 | 6155.62 | 574,282.91 | 0 | 2011 |
| 2011 | 1.89 | 6.19 | 16.88 | 7295.27 | 1,033,772.80 | 2 | 2011 |
| 2011 | 0.88 | 5.74 | 16.81 | 15,084.74 | 1,518,744.53 | 0 | 2011 |
| 2012 | 1.30 | 6.31 | 17.48 | 10,412.91 | 1,014,862.70 | 0 | 2012 |
| 2012 | 1.57 | 6.37 | 17.09 | 11,656.76 | 1,459,165.91 | 1 | 2012 |

Though CTGAN consistently outperforms TVAE in (Xu), TVAE was competitive in the current work, especially when combined with XGBoost in the SNLT pipeline. The synthetic datasets generated by CTGAN train our most accurate predictor when training with KY and classifying GA, but those generated by TVAE beat CTGAN in our SD and OH experiments. SNLT trained with TVAE data produced highest accuracies of 75.0% and 70.0% using random forest (RF) and XGBoost models. SNLT trained with CTGAN data, however, produces the highest accuracy of 87.5%. Other accuracies using synthetic datasets achieved modest accuracies of 62.5% and 60.0%, but this is a significant improvement over local-only and non-local training. We determined through experimentation that accuracies increased as sample sizes increased up to 1000 or 2000, but we noticed diminishing returns after that on all models except XGBoost, so we chose 1000 or 2000 samples for all other models. For XGBoost, which trains far faster than all the other models, we experimented with synthetic sample sizes up to 200,000, though 5000 usually delivered similar accuracies faster; therefore, all but one of our XGBoost models generate 5000

samples, and the other generates 200,000, as summarized in Section 4′s results tables. Overall, non-local and local training result in low classification accuracies, with no clear winner between the two; however, our SNLT accuracies are significantly higher than non-local training or local training.

## Related Work

While results in this team's previous papers are very promising and show very high accuracies using rich datasets collected by scientists and researchers, the current work considers the problem of very small datasets potentially collected by farmers at real-world farms. In previous experiments, our high accuracies are aided by the coexistence of many varieties of alfalfa growing in the same location, providing plenty of data to train and predict locally or on datasets of combined locations. On the other hand, the current work's end goal is to create a practical application that is useful to actual farmers, and we contend that farms in the real world will typically have much smaller datasets. This presents a problem where strong, accurate models are difficult or impossible to train. As a solution, PYCS offers the option to use already-trained models to estimate users' yields, even if their datasets are very small.

Our previous work has detailed some of the related work in the domain of ML and precision agriculture. Overall, the literature shows many complex techniques involving remote sensors, unmanned aerial vehicles (UAVs) or drones, computer vision, and image processing to develop vegetation indices to measure crop health (C. V. Whitmire) (J. R. Vance). Our work diverges from this trend in that our data come from public sources and simple weather data. More recently, Baral et al. at Kansas State University published a study of the effects of rain gaps on alfalfa yields across the United States. They aggregated alfalfa yield and weather data for 10 states, and like the current

work, they obtained their yield data from university variety trials (Baral). This helped inform the current work's use of data from South Dakota (SD) State University and Ohio (OH) State University, and as (Baral) is only interested in rainfed alfalfa, it provides a helpful guide of rainfed locations for our team to use in future experiments. The exhaustiveness of (Baral) inspires us to explore as many states as possible. Furthermore, they measure alfalfa yield over growing degree days (GDD), which means they do not consider days where the temperature did not meet some threshold alfalfa requires to grow, so that concept may help inform our future work (Baral). Otherwise, the current work bares little similarity to (Baral). That work essentially analyzed the cost in alfalfa of dry periods (rain gaps) using frontier function analysis and boundary function analysis. They also used conditional inference trees (CITs) to determine which weather features most affect crop yields. They did not compare ML models' ability to predict or estimate alfalfa yields, and they do not mention a goal of forecasting future yields using time series and ML, as the current work does.

The current work takes advantage of one DL model that has shown great promise in recent years—the generative adversarial network (GAN). Training a GAN requires finding the Nash equilibrium in a minimax game, an idea which precipitated a break-through in economics in the 20th century (Myerson). The GAN applies game theory to DL training, as a competition between two entities drives up the performance of each. One entity, called the generator, creates simulated, fictitious data that resembles the real training data. The other entity, called the discriminator, looks at real and simulated data and tries to determine which are which. As the discriminator improves at guessing, this causes the generator to generate more convincing fakes, which causes the discriminator to continue to become a better guesser, and so on. This equates to a game where each player maximizes its own chances of winning by minimizing its opponent's chances, also known

as a minimax game (Goodfellow). Though mainly leading to breakthroughs in the computer vision domain, GANs have also shown promise in other applications such as differential privacy with DPGAN (Xie), generating private tabular health record data with medical GAN (MedGAN) (Choi), and synthesizing table data with TableGAN in (Park). (Xie), (Choi), and (Park) demonstrate a GAN's ability to generate data that is statistically similar to the training data but introduces enough noise that connections between real people and the data are broken, protecting individuals' privacy. MedGAN combines a GAN with an autoencoder to apply this concept to health records, where privacy is a clear concern that may inhibit individuals' willingness to share data, but also where abundant data can make ML more useful in helping cure illness (Choi).

CTGAN is called a conditional GAN because it uses a conditional generator and training-by-sampling to address class imbalance in discrete columns. CTGAN introduces a technique called "mode-specific normalization" to normalize columns individually, and its underlying networks are fully connected (Xu).

We also investigate the usefulness of variational autoencoders (VAEs) in synthesizing table data, as VAEs are another class of generative neural network that has received a lot of attention and shown promising results in recent years (Xu). Specifically, this work uses the tabular variational autoencoder (TVAE) proposed in (Xu), which adapts a plain VAE to address distribution concerns particular to generating table data.

Another somewhat recent development that helped our pipeline produce some of its best results is extreme gradient boosting or the XGBoost model. Generally, boosting combines an ensemble of weak learners to create one strong learner. One popular boosting algorithm called Adaboost often works well with random forest (RF) or decision tree (DT) models trained iteratively, where each iteration improves on the errors of the previous by weighting incorrect

predictions higher in subsequent iterations, until some predefined depth or maximum accuracy is reached (Freund). In XGBoost, DTs are trained to minimize prediction errors using gradient descent, creating an optimized strong learner, and this results in extremely fast training times, facilitating training on very large datasets (Chen).

Materials and Methods

Our alfalfa yield data come from publicly available alfalfa variety trials conducted by the University of Georgia (University of Georgia), University of Kentucky (University of Kentucky), South Dakota State University (South Dakota State University), and Ohio State University (Ohio State University). Our weather data in GA and KY come from various NOAA resources, while our weather data for SD and OH comes from the Daymet tool, provided by Oak Ridge National Laboratory in Oak Ridge, TN and supported by NASA (Daymet). We provide our curated, aggregated datasets along with source code at the Github repository for this project: www.github.com/thejonathanvancetrance/SNLT (accessed 12/20/2022). Every record in the original KY and GA datasets include the following 15 features: State, City, Date Sown, Variety, Date of Cut, Julian Day, Yield (tons/acre), Time Since Sown (Days), Time Since Last Harvest (Days), Total Radiation (MJ/m2), Total Rainfall (mm), Avg Air Temp (°C), Avg Min Temp (°C), Avg Max Temp (°C), Avg Soil Moisture (%). Table 4.1 depicts the five most salient features we settled on for training for KY and GA experiments, minus class, year, and yield: average minimum temperature, average maximum temperature, total rainfall, solar radiation, and soil moisture. In SD and OH experiments, we omitted soil moisture, because we cannot consistently obtain soil moisture data for most locations as explained in Section 5. We set the thresholds of our yield classes according to standard deviation, where we label yields less than −1 standard deviations

below the mean as low (class 1), yields between −1 and +1 standard deviations as medium (class 2), and yields more than +1 standard deviations above the mean as high yield in GA and KY. Our yield thresholds for all locations are summarized in Table 4.3. We settled on the hyperparameters shown in Appendix A via experimentation.

We began the current work with the same GA and KY data, plus the code base and API that we used in (C. V. Whitmire) and (J. R. Vance). We reduced the target GA dataset to seven records by including only one variety per year and annualizing the data. A real-world farm's dataset may not include multiple varieties, so we attempt to simulate this in our target datasets. Additionally, annualizing our data helps reduce the influence of short-term anomalous weather conditions on our models. Furthermore, some variety trial data, such as PA data, are only reported annually, so some of our data were originally annualized to be compatible with those. We expect this annualized data to simulate accuracies we might expect from PYCS when using already-trained models to estimate very small datasets. In GA and KY, the alfalfa yields are total annual yields. As this project is ongoing and evolving, we used total yields originally to match data from other states that report annually. As we expanded our data to include OH and SD, we decided that the average yield from all cuts in a year may be more informative when multiple cut information is available, mainly because some years have more cuts than others, potentially inflating the total annual yield. The current work is intended as a step toward forecasting future yields, but we are not truly forecasting yet, as we train with data from the entire year up to the date of the final cut in GA and KY. Therefore, our total rainfall, total solar radiation, average minimum and maximum temperature, and average soil moisture data typically represent about the first eight or nine months of that year in GA and KY. In SD and OH, those features represent the totals and averages over the lifetime of one planting, which may be several years, up to the last cut of that year. Table 4.1

lists “Solar Radiation” and “Total Rainfall” while Table 4.2 lists “Total Accumulated Radiation” and “Total Accumulated Rain” because of these differing approaches, where Table 4.1 reflects one year’s weather until the final cut, and Table 4.2 reflects weather totals over the lifetime of that planting up to the final cut of that year. The average minimum and maximum daily temperatures are sums of each day’s minimum and sum of each day’s maximum temperatures, divided by the number of days considered. Table 4.1 weather data is sourced from the NOAA, which reports solar radiation as solar irradiation in $MJ/m^2$, while Table 4.2 weather data is sourced from Daymet, which reports solar radiation as solar irradiance in $W/m^2$. Though Tables 4.1 and 4.2 reflect slightly different annualization processes, we think both are reasonable approaches.

The annualization process reflected in Table 4.2 resulted in a target dataset of 10 records of OH data, then we removed two arbitrary records to create an eight-record dataset for better comparison with GA results. Once again, we followed this process with SD data, creating a local target ten-record dataset of data from only Highmore, SD, leaving the other SD locations for non-local training and synthesis; then, we again arbitrarily removed two records to create an eight-record Highmore dataset for comparison. We also annualized the SD training dataset.

We modified the original code base to build classification models instead of their regression counterparts, using the ScikitLearn (Pedregosa) ML library for the Python programming language along with NumPy (Oliphant). We swapped (Pedregosa)’s k-nearest neighbors (KNN) regressor for a KNN classifier, random forest (RF) regressor for a RF classifier, linear regression for logistic regression (LR), support vector regressor for a support vector classifier (SVC), decision tree regressor for decision tree classifier (DT), and the multilayer perceptron (MLP) regressor for an MLP classifier (which we refer as the artificial neural network or ANN interchangeably). We added ScikitLearn’s XGBoost classifier, not tried in our previous

work, to our cast of models. Since the task is now one of classification, we modified the API used in (C. V. Whitmire) to provide accuracy percentage scores indicating percent of estimates correct, rather than correlation coefficients or mean absolute errors (MAEs). We used Matplotlib to generate confusion matrices (Hunter).

Table 4.3. Class labels by standard deviation. We calculated the standard deviations (stdvs) of the yields to assign class labels for yields in each location; we used three different thresholds (1, 0.8, 0.5) to avoid high class imbalance.

| Location | Class 1 (Low Yield) | Class 2 (Medium Yield) | Class 3 (High Yield) |
|---|---|---|---|
| KY & GA | <−1 stdvs below mean | −1 < +1 stdvs below/above mean | >+1 stdvs above mean |
| SD & OH | <−0.5 stdvs below mean | −0.5 < +0.5 stdvs below/above mean | >+0.5 stdvs above mean |
| SD & Highmore, SD | <−0.8 stdvs below mean | −0.8 < +0.8 stdvs below/above mean | >+0.8 stdvs above mean |

For local training, we trained all models on our very small GA, OH, and SD datasets reflected in Tables 4.1 and 4.2, then we used those models to classify those same datasets, resulting in low accuracies as shown in Tables 4.4–4.8. For regular non-local training, we trained all models on: the KY dataset with 183 records from Lexington, KY; the complete SD dataset with 767 records; and the SD dataset minus Highmore with 604 records. Next, we used those models to classify the local GA, OH, and Highmore, SD datasets, resulting in the different but mostly unimproved accuracies shown in Tables 4.4–4.8.

Table 4.4. Train KY, test GA. Best accuracies from GA local training, non-local training (KY), and SNLT with CTGAN and TVAE (KY), each for three class labels (high, medium, and low yield), with synthetic dataset size of 1000 samples, except XGBoost, which is 200,000 samples. Highest accuracy is bold. *Indicates the model fails to train.

| **Model** | **Local** | **Non-Local** | **SNLT (TVAE)** | **SNLT (CTGAN)** |
|---|---|---|---|---|
| ANN | 28.6% | 42.8% | 57.1% | 57.1% |
| KNN | 28.6% | 28.6% | 42.8% | 57.1% |
| LR | NA * | 14.3% | 14.3% | 57.1% |
| DT | 42.8% | 28.6% | 57.1% | **85.7%** |
| SVC | 14.3% | 14.3% | 57.1% | 57.1% |
| RF | 28.6% | 42.8% | 42.8% | 57.1% |
| XGB | 28.6% | 28.6% | 28.6% | 57.1% |

Table 4.5. Train SD, test OH with 8 samples. Best accuracies (bold) from OH local training, non-local training (SD), and SNLT with CTGAN and TVAE (SD), each for three class labels (high, medium, and low yield), plus size of synthesized dataset. *Indicates model failed to train.

| **Model** | **Local** | **Non-Local** | **SNLT (TVAE)** | **SNLT (CTGAN)** | **Sample Size** |
|---|---|---|---|---|---|
| ANN | 25.0% | 12.5% | 62.5% | 62.5% | 5000 |
| KNN | 25.0% | 12.5% | 62.5% | 50.0% | 2000 |
| LR | 25.0% | NA* | 25.0% | 50.0% | 2000 |
| DT | 37.5% | 12.5% | 62.5% | 62.5% | 3000 |
| SVC | 25.0% | 12.5% | 50.0% | 50.0% | 2000 |
| RF | 12.5% | 12.5% | **75.0%** | 62.5% | 2000 |
| XGB | NA * | 50.0% | **75.0%** | 62.5% | 5000 |

Table 4.6. Train SD, test OH with 10 samples. Best accuracies (bold) from OH local training, non-local training (SD), and SNLT with CTGAN and TVAE (SD), each for three class labels (high, medium, and low yield), plus size of synthesized dataset.

| **Model** | **Local** | **Non-Local** | **SNLT (TVAE)** | **SNLT (CTGAN)** | **Sample Size** |
|---|---|---|---|---|---|
| ANN | 20.0% | 10.0% | 50.0% | 40.0% | 2000 |
| KNN | 10.0% | 40.0% | 50.0% | 40.0% | 2000 |
| LR | 10.0% | 10.0% | 30.0% | 40.0% | 2000 |
| DT | 10.0% | 20.0% | 50.0% | 50.0% | 2000 |
| SVC | 20.0% | 20.0% | 40.0% | 60.0% | 2000 |
| RF | 30.0% | 30.0% | 50.0% | 50.0% | 2000 |
| XGB | 10.0% | 50.0% | **70.0%** | 60.0% | 5000 |

Table 4.7. Train SD, test Highmore, SD w/10 samples. Only one variety per year resulting in 10 target samples: best accuracies (bold) from Highmore local training, non-local training (rest of SD), and SNLT with CTGAN and TVAE (rest of SD), each for three class labels (high, medium, and low yield), plus size of synthesized dataset.

| **Model** | **Local** | **Non-Local** | **SNLT (TVAE)** | **SNLT (CTGAN)** | **Sample Size** |
|---|---|---|---|---|---|
| ANN | 20.0% | 60.0% | 50.0% | 50.0% | 2000 |
| KNN | 40.0% | 40.0% | 50.0% | 50.0% | 2000 |
| LR | 30.0% | 50.0% | 50.0% | 60.0% | 2000 |
| DT | 30.0% | 50.0% | 60.0% | 50.0% | 2000 |
| SVC | 40.0% | 50.0% | 60.0% | 50.0% | 2000 |
| RF | 20.0% | 60.0% | 60.0% | 50.0% | 2000 |
| XGB | 30.0% | 60.0% | **70.0%** | 50.0% | 5000 |

Table 4.8. Train SD, test Highmore, SD w/8 samples. Only one variety per year and 8 target samples: best accuracies (bold) from Highmore local training, non-local training (rest of SD), and SNLT with CTGAN and TVAE (rest of SD), each for three class labels (high, medium, and low yield), plus size of synthesized dataset.

| Model | Local | Non-Local | SNLT (TVAE) | SNLT (CTGAN) | Sample Size |
|---|---|---|---|---|---|
| ANN | 25.0% | 37.5% | 62.5% | 50.0% | 2000 |
| KNN | 12.5% | 50.0% | 50.0% | 50.0% | 2000 |
| LR | 25.0% | 50.0% | 50.0% | 50.0% | 2000 |
| DT | 12.5% | 25.0% | 62.5% | 50.0% | 2000 |
| SVC | 37.5% | 37.5% | 62.5% | 50.0% | 2000 |
| RF | 12.5% | 37.5% | 62.5% | 50.0% | 2000 |
| XGB | 25.0% | 50.0% | **75.0%** | **75.0%** | 5000 |

Our third and featured set of experiments explored our SNLT pipeline, depicted in Figure 4.1. SNLT begins by choosing a dataset that is not local to the target area that is being classified as high, medium, or low yield. That non-local dataset is fed to a data synthesizer, which in the case of the current work is either CTGAN or TVAE. Figure 4.1 depicts SNLT using a fully connected deep CTGAN, but other synthesis techniques could be used in place of this in the SNLT pipeline. The synthesizer outputs a larger training dataset than the original non-local dataset, and the size of the new dataset is an adjustable parameter of SNLT. As our experiments confirmed, these larger synthesized datasets usually train more accurate classifiers than those trained by the non-local data. Next, a model is selected for training with the synthesized data. We experimented with ANN, KNN, RF, DT, SVC, LR, and XGBoost, and since DT and XGBoost produced the best results, Figure 1 depicts SNLT as using a tree-style model, though the others are also valid. Finally, the trained model is used to classify the target dataset's alfalfa yield. This dataset would typically be too small to adequately train most models or effectively split into test and training. Based on sun, rain, temperature, and soil moisture when available, the model classifies target alfalfa yields as low, medium, or high, and SNLT produces accuracy scores as compared to the true class labels.

Our experiments begin with the same labeled KY and SD non-local training data as before, but we feed them through a generative model that synthesizes the data, resulting in larger datasets of 1000 to 200,000 records. The generative model in the pipeline is interchangeable with any generative model or synthesis technique, but the current work explores CTGAN and TVAE (Xu). Next, we use these synthetic source states' data to train all our models, and we again use those trained models to classify labels for our small GA, OH, and SD targets, resulting in the promising accuracies depicted in Tables 4.4–4.8.

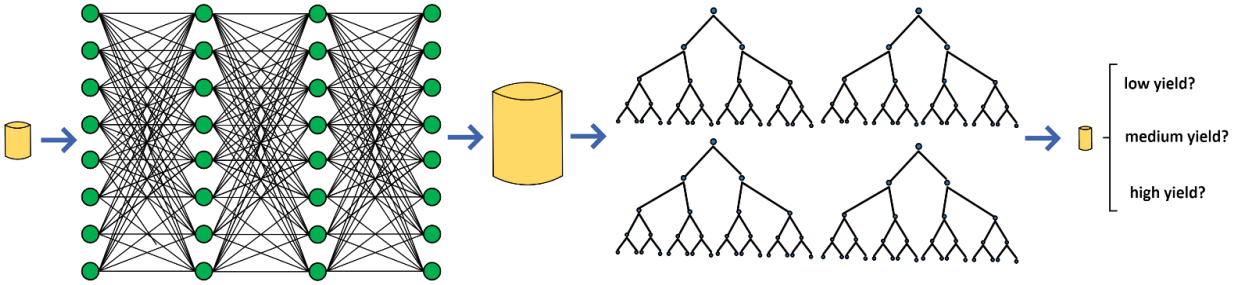


Figure 4.1. The SNLT Pipeline. Original data fed to a data synthesizer like CTGAN, which outputs lager dataset. Synthesized dataset used to train models like XGBoost. Trained model classifies very small target dataset.

Results

Whereas we previously found non-local training unviable for regression (J. R. Vance), our classification version of non-local training resulted in meaningful, though low accuracies which do not always beat local training, as indicated in Tables 4.4–4.8. Our SNLT pipeline produced the highest accuracy score of 85.7% (Table 4.4) out of these three general approaches, training a DT model with 1000 samples of data synthesized from KY using CTGAN, and classifying a very small target GA dataset of seven records. Figure 4.2 depicts these results as a confusion matrix. On other

states' datasets of sizes 8 and 10 samples, SNLT achieved top accuracies of 75.0% and 70.0%, respectively. As Table 4.5 shows, SNLT achieved 75.0% accuracy training on data synthesized from SD and classifying OH, training RF and XGBoost models with 2000 and 5000 records, respectively, using TVAE, and classifying a very small target OH dataset of eight records. Figure 4.3 depicts these results as a confusion matrix. As Table 4.6 shows, SNLT achieved 70.0% accuracy training on data synthesized from SD and classifying OH, training an XGBoost model with 5000 records, using TVAE, and classifying a very small target OH dataset of ten records. As Table 4.7 shows, SNLT achieved 70.0% accuracy training on data synthesized from most of SD and classifying Highmore, SD, training an XGBoost model with 5000 records, using TVAE, and classifying a very small target Highmore, SD dataset of ten records. Figure 4.4 depicts these results as a confusion matrix. Finally, as Table 4.8 shows, SNLT achieved 75.0% accuracy training on data synthesized from most of SD and classifying Highmore, SD, training an XGBoost model with 5000 records, using TVAE and CTGAN, and classifying a very small target Highmore, SD dataset of eight records. Even when SNLT resulted in underwhelming accuracies of 60.0% and 62.5%, it beat regular non-local training and local training soundly and consistently as indicated in Tables 4.4–4.8. We were pleased to see XGBoost surface as a consistent top performer, as this was the only model in our cast that would train in an acceptable timeframe on tens to hundreds of thousands of records, and the only model that did not drop in accuracy with larger sample sizes. Almost everywhere our results indicate good scores using SNLT with XGBoost using 5000 samples, we received the same results using 100,000 and 200,000 samples, though we emphasize the smaller sample sizes of 5000 because they are effectively cheaper. Each run of SNLT produces a new synthetic dataset and a new trained model, and while not all runs are equally successful, once we produce a relatively successful model and dataset, we can store them for repeated classifications.

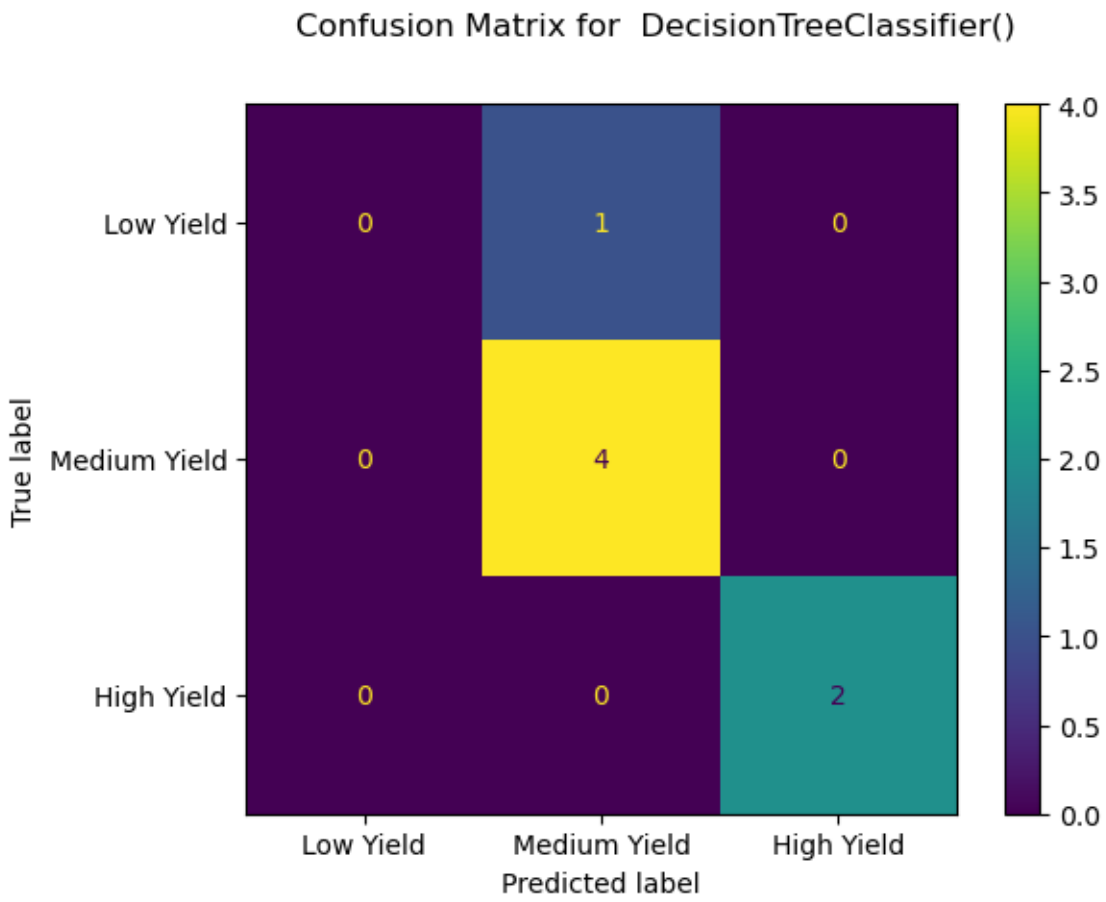


Figure 4.2. Confusion matrix for Table 4.4 best accuracy of 85.7% using DT and CTG with 1000 samples.

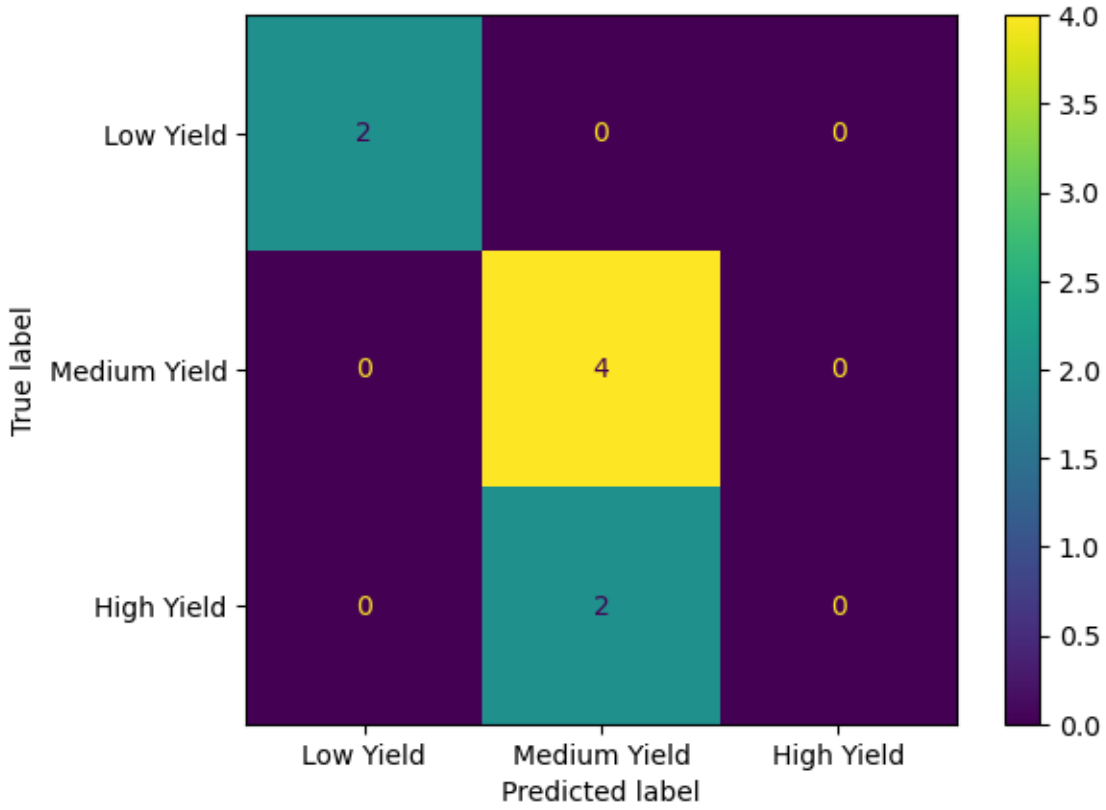


Figure 4.3. Confusion matrix for Table 4.5 best accuracy of 75.0% using RF and TVAE with 2000 samples.

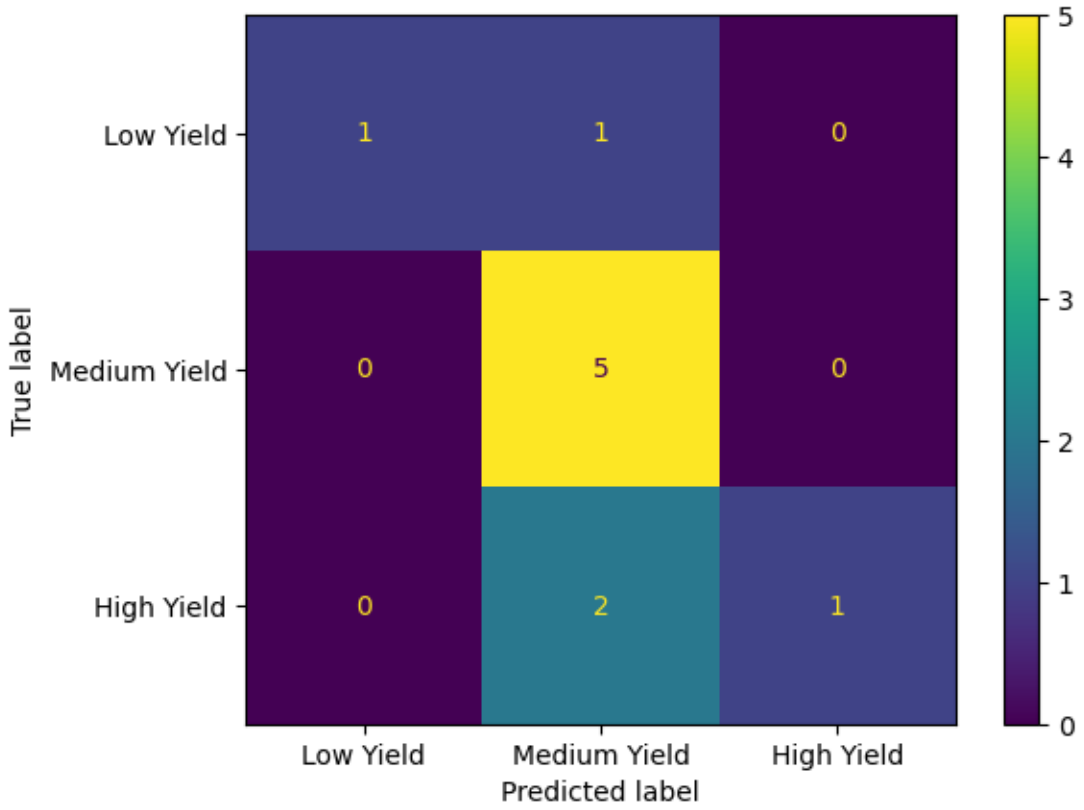


Figure 4.4. Confusion matrix for Table 4.7 best accuracy of 70.0% using XGB and TVAE with 5000 samples.

Discussion

Our results show that we can significantly improve estimation accuracy using our SNLT pipeline over training with non-local data or with training and estimating on the same local target dataset if it is very small. It was key for us to convert our regression problem to a classification problem. Our results are encouraging, and we think further efforts may increase our accuracies. For example, our best accuracy is higher in KY and GA experiments, where the soil moisture feature is included, and this leaves us wondering whether soil moisture is key to higher accuracies. Unfortunately, we found it difficult to obtain soil moisture data consistently throughout the U.S. Compared to our previous work, the current work aggregates yield data with much more precise weather data for SD and OH based on latitude and longitude, using the Daymet tool (Daymet). Before, our weather

and soil moisture data came from those NOAA stations nearest to where each variety trial was conducted, which is often dozens of miles away or more. These NOAA data often include soil moisture data, but it is not always pre-sent or complete. Ideally, we would harvest data from a source that allows us to look up weather data by geolocation and would also include soil moisture data for that precise location, so we note this gap in publicly available data as well as this opportunity for someone to create such a tool.

Our results are promising enough to warrant further development of PYCS, which is still in its infancy, and the results facilitate a better understanding of how PYCS might perform and be useful in the real world. For future work, we are continuing to develop PYCS, which will provide predictions for future yields based on the past, thus providing actual crop forecasting. We have reported our best results, which come from combinations of our most accurate models with our best synthesized datasets. Our pipeline features storable models and datasets for reusability, so the live PYCS tool can estimate unseen locations using these top performers once they are created.
We are also working on gradually expanding our dataset to cover the entire contiguous United States (CONUS), or at least those where alfalfa variety trials are per-formed, especially where the crops are rainfed and not exclusively “Roundup Ready”, as explained below. Using the Daymet (Daymet) tool has vastly increased our efficiency with curating data and allows us to obtain weather data for any precise location where we have access to variety trials, so we expect to accelerate the growth of our datasets going forward. Since we are trying to contribute to mitigating the effects of climate change, we are more interested in rainfed crops, as rain is a climate factor, not to mention that artificial irrigation might cloud the significance of rain as a feature.

Glyphosate herbicide, known under the trade name Roundup, is a very common tool for battling pests in many crops, including alfalfa. Glyphosate kills virtually any plant it contacts,

except those whose seeds are genetically engineered to be immune to glyphosate, and such modified seeds are called “Roundup-Ready.” Crops grown from Roundup-Ready seeds are tolerant to glyphosate, meaning that fields where these crops are grown can be treated with the chemical, which will kill surrounding plants and weeds but not the crop itself (Duke and Powles). However, in recent years, the World Health Organization determined that glyphosate is a “probable carcinogen” that may negatively affect the health of humans, animals, and our ecosystem, and that more research is necessary to determine the health consequences of such a ubiquitous product (Duke and Powles). While research connecting glyphosate and related pesticides to cancer is not proven, our team generally subscribes to a non-invasive approach to precision farming, adopting the perspective that we can more effectively mitigate climate change by adapting to our environment than manipulating it. Noting this recent controversy surrounding Monsanto’s Roundup herbicide, its active chemical glyphosate, and its potential carcinogenic qualities and negative environmental impacts, our work con-siders this product to be potentially at odds with efforts to combat climate change in a non-invasive way; therefore, our work focuses on crops that do not use glyphosate.

The next phase of our team’s research will focus on true forecasting of future alfalfa yields based on time series data and models. We are exploring known time series models like autoregressive integrated moving average (ARIMA) and vectorized auto-regression (VAR), but we are also investigating a pretraining technique where we re-tune models already trained according to a sliding time window. We are happy to pro-vide our data and code publicly at https://www.github.com/thejonathanvancetrance/SNLT (accessed 12/20/2022) so that others may reproduce and extend this work.

## Conclusions

The main contributions of this work are: we show that training classifiers with synthesized data often produces more accurate classifiers than training with real data, when the synthesized dataset is larger than the real one, and the target dataset is very small; we propose the novel SNLT pipeline, which trains more accurate models than local and non-local training; we propose the PYCS application, which offers a graphical user interface to perform crop yield estimation; we provide a publicly available, growing dataset of aggregated alfalfa yield and weather data. Furthermore, this work sets the stage for true forecasting of future yields, which we plan to include in future iterations of PYCS. SNLT produced the highest accuracy of 85.7% with a decision tree classifier, scored above 70% accuracy with an XGBoost classifier, and beat non-local and local training by over 40%. These results indicate that the SNLT pipeline is useful, and that it is a good feature to include in the PYCS application. When a farmer using PYCS wants to estimate alfalfa yields for a location where data is too scarce to train ML models, and when non-local training data from a nearby location is more abundant, yet too scarce to train strong models, we can synthesize larger datasets, which greatly improve the classification accuracy of our ML models. Furthermore, as researchers, we can expand the data and train and store more models as we use SNLT to conduct further experiments on new locations. Admittedly, we would like to see higher accuracies on larger target datasets with greater consistency, but we may have merely gotten our foot in the door with demonstrating this technique's usefulness. XGBoost has consistently produced encouraging results and trained very fast on very large datasets, so we plan to include it and similar models in future work, moving beyond the traditional models in our early work. A deeper study of XGBoost

may reveal tuning or modifications we should perform to achieve better results. Most importantly, our results show 70% or greater accuracy in multiple locations on multiple runs, fairly well convincing us that our pipeline is useful and merits further exploration.

# CHAPTER 5

# UTILITY OF DOMAIN ADAPTATION FOR BIOMASS YIELD FORECASTING [3]

[3] Vance, J., Rasheed, K., Missaoui, A., Miller, J. A., Arabnia, H.R., & Maier, F. W. 2022. To be submitted to *AI*.

Abstract

In previous work, we used machine learning (ML) to estimate past and current alfalfa yields, and we showed that domain adaptation (DA) with data synthesis shows promise in classifying them. In the current work, we use similar techniques to forecast future alfalfa yields. We propose a novel technique for forecasting alfalfa time series data that exploits stationarity and predicts differences in yields rather than the yields themselves, using a non-specific ML model, then adds the predictions to an average of yields from previous years to output forecasts. We show that our forecasting technique generally provides more accurate forecasts than the established ARIMA family of forecasters for both univariate and multivariate time series. Furthermore, our ML-based technique is potentially easier to use than the ARIMA family of models. Also, we extend our previous work by showing that DA with data synthesis also works well for predicting continuous values, not just classification. We propose our own novel scale-invariant tabular synthesizer (SITS) that is competitive with or superior to other established synthesizers in producing data that trains strong models. We show that our synthesis algorithm leads to R scores over 100% higher than the established synthesizer in this domain, and our ML-based forecasters beat the ARIMA family with sMAPE scores as low as 12.81%. Finally, we combine ML-based forecasting with DA (ForDA) to create a novel pipeline that improves forecast accuracy with sMAPE scores as low as 9.81%. As alfalfa is crucial to the global food supply, and as climate change creates challenges with managing alfalfa, we hope this work can help address those challenges and contribute to the field of ML.

Introduction

As the Earth's climate changes, its population grows, and environments and weather patterns become increasingly unpredictable, problems in efficient crop production are among the most important to solve. Alfalfa, a particularly important livestock feed, has been called "Queen of the forage crops", and it is the crop of focus for the current work (El-Ramady). Alfalfa is crucial to global food security because it is a valuable, sustainable, and protein-rich livestock feed as well as a nutritious food for people worldwide. In other words, alfalfa is not just food for people, but food for people's food. Meanwhile, climate change directly threatens alfalfa's efficient and sustainable cultivation (Kulkarni). The importance of alfalfa is evidenced by the attention it receives from many land-grant universities, which cultivate alfalfa and collect and publish data on it in the form of variety trials. These variety trials, along with weather data, are the data underlying the current work.

Meanwhile, in 2015, the United Nations (U.N.) agreed on and published the "Sustainable Development Goals", 17 goals that "are the blueprint to achieve a better and more sustainable future for all. They address the global challenges we face, including poverty, inequality, climate change, environmental degradation, peace and justice." The U.N. celebrated the 5$^{th}$ anniversary of their adoption in 2020. Most specifically, the current work attempts to contribute toward "Goal 2: Zero Hunger," and "Goal 13: Climate Action" (United Nations). The current work's technical focus is forecasting alfalfa yield data as time series. We also researched domain adaptation (DA) along the way as a steppingstone between crop yield estimation and forecasting. We present this

work in three phases. The first phase is DA, the second is ML-based forecasting, and the third phase is combining ML-based forecasting with DA.

The first phase of the current work focuses on estimating past and current yields with DA. DA is a close relative of transfer learning (TL) and seeks to solve data scarcity issues by using training data, called the source, that is distinct from the test data, called the target. In DA, the source data come from some other domain than the target, having the same feature space but a different data distribution. TL also uses source data from some other domain than the target, having a different feature space from the target. In the current work, our sources and targets are sometimes different U.S. states, and sometimes different locations within the same state, so these datasets have the same feature space and different data distributions. There is also typically some data augmentation step in the DA process (Kouw). Sometimes, a model is pretrained with a larger source dataset before being trained further on the target dataset with the scarcity issue (Bashar). Other times, a new, larger dataset is generated or synthesized from the source dataset, which can lead to improved training of ML models (Peng, Visda: The visual domain adaptation challenge). We propose a novel scale invariant tabular synthesizer (SITS), which shows improvement over established synthesizers. The DA approach in the current work combines both these techniques, expanding the training data with SITS and pretraining extreme gradient boost models (XGBoost or XGB). When there is no data augmentation step, but simply training on a source that is different from but similar to the target, as we did in previous work, we call that trivial domain adaptation (TDA) (Vance, Whitmire and Rasheed).

The second phase of the current work focuses on forecasting future alfalfa yields using ML and enforcing stationarity. Univariate time series models analyze single series of values to create forecasts, while multivariate time series models analyze cross-correlations among more than one

series of values to create forecasts. To offer an intuitive definition, time series data may consist only of one feature, which is known as univariate time series, or it may have several features, which is known as multivariate time series (Du Preez). When the features do not directly cause one another, they may be known as exogenous features or variables, and when they do cause one another, they may be referred to as endogenous features or variables, roughly speaking. For example, in the current work, we focus on precipitation, solar radiation, temperature, and yield, none of which directly cause each other, though solar radiation affects temperature somewhat. On the other hand, yield is directly affected by the size of the growing site, so that would be more of an endogenous variable. Lütkepohl offers a formal mathematical definition of exogenous and endogenous variables in his book on time series (Lütkepohl). We propose both univariate and multivariate versions of our forecaster, and we compared them with the traditional family of auto-regressive integrated moving average (ARIMA) statistical forecasting models. We compared our univariate ML-based forecaster to the univariate ARIMA and seasonal ARIMA (SARIMA) models, and we compared our multivariate ML-based forecaster to the multivariate SARIMA with exogenous variables model (SARIMAX). We found that our ML-based forecaster produced more accurate forecasts than the ARIMA family, as detailed in the results section, with symmetric mean absolute percent errors (sMAPE) as low as 16.94% with ours versus a best of 54.46% with ARIMA. We compared univariate with multivariate models, and we show an improvement in sMAPE from 16.94% to 12.81% when we included the exogenous variables in our ML-based forecaster, while SARIMAX produced a sMAPE of 34.64%.

The third phase of this work combines our ML-based forecasting technique with DA using SITS, and shows our most promising sMAPE scores of all, as low as 9.81% in the best results. The journey from our marginally successful DA work in phase one to our impressive results in phase

three is the main point of this paper. Our early DA experiments show that this technique may be helpful when target data are scarce, and they focus on the very difficult problem of using data with one distribution to estimate that of a different distribution. As we progressed to phase three, our results became very promising, but the problem became easier in that we used local training after the initial pretraining step, so even though forecasting is more challenging than estimating the past and present, the local training makes it successful.

We cite four significant contributions made by the current work. First, we propose a novel DA technique that combines data synthesis with pretraining an XGBoost model and substantially increases estimation accuracy. Second, we propose a novel data synthesis algorithm that shows improvement over established models. Third, we propose a novel ML-based forecasting algorithm that exploits stationarity. Fourth, we combine our ML-based forecasting technique with our DA pipeline to produce more accurate forecasters.

## Related Work

One related work by Jin et al. proposes a domain adaptation forecaster (DAF) that also uses data synthesis and ML with attention sharing to produce forecasts in four benchmark domains: household electricity consumption, traffic patterns, grocery sales, and visits to Wikipedia pages. Like the current work, they generate synthetic datasets for training and compare the results to those without synthesis. Their attention module "captures domain-dependent patterns by context matching using domain-invariant queries and keys," which roughly means they adapt values in one domain to another by looking for and weighting these keys rather than the actual values. They report promising results, but the primary metric they feature is normalized deviation (ND), which

is difficult to compare with the current work's results, and which may not be as pervasive in time series literature as sMAPE and others (Jin).

Bashar et al. explore DA with pretraining models like XGBoost and others to detect hate speech on social media post-COVID-19. They argue that the landscape of social media language has changed since the pandemic, making it more difficult to detect hate speech directed toward the East Asian community, and that DA can help adapt older data to make it more useful in the present day. They propose a progressive transfer learning technique that is able to incorporate knowledge from multiple datasets before being applied to the classification task at hand. They report an improvement in classification accuracy, but unlike the current work or Jin's, they do not attempt to forecast patterns into the future (Bashar).

There are many recent papers that show progress in the field of DA using synthesis, but it may be much easier to find related work using DA and synthesis for image classification and segmentation, which are very different problems from the one tackled by the current work, where we need to predict continuous values. Peng et al. (Peng, Visda: The visual domain adaptation challenge) created the Visual Domain Adaptation (VisDA) dataset and challenge, where they provide synthetic images for use as source data in domain adaptation along with real images of the same kinds of objects for target data. These are synthetic and real images of everyday objects like cars, airplanes, trains, horses, and others. They also provide baseline results in classifying and segmenting these images for anyone interested in accepting the challenge and beating state-of-the-art (SOTA). This work motivates us to attempt DA for our agricultural domain and regression problem (Peng, Visda: The visual domain adaptation challenge). Another synthetic dataset designed to facilitate DA in image classification, called the Synthetic person Re-Identification (SyRI) dataset is presented by Bak et al. in their 2018 work. Specifically, they address the problem

of re-identifying people in images already identified from images in different cameras, lenses, changing lighting conditions, different angles, and other complexities that constitute challenges in computer vision (Bak). Cao et al. use DA with data synthesis to classify English-language handwritten characters in common benchmark datasets. They focus on disentanglement analysis, or using ML to distinguish between features specific to one domain from those common to the source and the target. Using this information from the source and target, they can synthesize more of the labeled target data, for improved DA performance (Cao). Choudhary et al. published a 2020 survey of state of the art in using DA for medical imaging, including a chapter on leveraging data synthesis for DA in that domain, which again focuses on computer vision, classification, and segmentation (Choudhary). Li et al. take inspiration from Cao (Cao) in their work using DA and data synthesis to estimate animal poses, and they report their models' percentages of correctly classified animal body parts like chin, leg, knee, hooves, and others for horses, tigers, sheep, and other animals. They report a large margin of improvement over other work in this domain; however, they report a mean accuracy of 57.23% over classifying all body parts for an unseen Zebra dataset, which may not be high enough to be useful (Li). Overall, this very brief survey of the literature shows that DA with synthesis is an active research area in the domain of computer vision, mainly for classification and segmentation tasks, but DA with synthesis in the domain of crop yield estimation for regression tasks is relatively unexplored, which motivates the current work.

In previous work, the current authors presented results of DA with synthesis classification experiments performed using a conditional tabular generative adversarial network (CTGAN) and tabular variational autoencoder (TVAE) proposed by Park et al. (Park), and we provided a more detailed description of those networks in that previous work (Vance, Rasheed and Missaoui).

While these synthesizers motivated us to try data synthesis in our own problem, our own SITS synthesizer produced the best results in the current work.

Kastens et al. studied crop yield time series forecasting using computer vision with masking, and while they report relative success, they do not explore any forecasting models beyond linear regression, and they do not compare their results to other forecasting models like the ARIMA family. They forecast metric tons per acre for corn, wheat, and soybean, but not tons per acre of alfalfa, and their metric of choice is mean absolute error (MAE), so it is difficult to compare the quality of their results to the current work's. Kasten et al. claim to require at least 11 years of training data to train a strong forecaster, while the current work produces very low sMAPE scores with as few as six years of training data. Finally, their work is another example of computer vision, which involves extra complexity and expensive equipment beyond the needs of the current work (Kastens). Choudhury and Jones compared statistical forecasting techniques to forecast maize yields in Ghana, and they report a significant contribution in this domain with mean squared errors (MSEs) as low as .03 metric tons per hectare using an Auto Regressive (AR) model, though that text does not appear to address or explain the table that features this metric. Furthermore, they emphasize coefficient of determination ($R^2$) scores in that paper's discussions, which our experiments have shown to not be a very reliable metric for measuring the quality of time series forecasts. We elaborate on this in our results and discussion sections. Choudhury and Jones also limit their results to those from AR models and a few varieties of exponential smoothing, though they refer to it as an autoregressive and moving average (ARMA) model in their discussions, and they do not explore multivariate models (Choudhury). Bose et al. propose a spiking neural network (SNN) to forecast winter-wheat crop yields from multispectral imaging time series data, and the SNN "encodes temporal information by transforming input data into trains of spikes that represent

time-sensitive events" into a positive or negative binary value called a spike. That work reports high average prediction accuracies and R scores, and it reports beating linear regression (LR), k-nearest neighbors (KNN), and support vector regression, (SVR), but they do not compare their model to SARIMAX or other non-ML multivariate models. Most important, that work forecasts only one yield per year, six weeks into the future, while the current work forecasts three or four yields per year usually at least nine months into the future. Also, their R score appears to be calculated over the 14 years of predictions, so it bears little relevance to any R scores reported in the current work, which are calculated from a year's forecast of several time points and the true values for that year. Also, although Bose et al. report that their SNN produces MAE scores as low as 0.24 tons per hectare, the current work's ML-based technique produces MAE scores as low as 0.15 tons per acre, which is competitive even though the current work tackles an arguably more difficult forecast (Bose). Pavlyshenko proposed a model stacking technique that showed good results in predicting future sales and that their approach beat ARIMA and other ML models (Pavlyshenko). However, that work does not report being effective for forecasting crop yields.

The traditional forecasting models against which we compare our ML-based forecasting technique are the family of autoregressive integrated moving average (ARIMA). These are ARIMA, seasonal ARIMA (SARIMA), and SARIMA with exogenous variables (SARIMAX). These popular models appear frequently in time series literature and are formally defined in many texts including one by Vishwas and Patel. In layman's terms, ARIMA uses values in previous time points, called lags, to predict future values, and this is the autoregressive (AR) part. For the moving average (MA), ARIMA uses error lags, or differences between predictions and true values to predict future errors. The integration (I) step involves calculating differences between related time points to reduce seasonality and other statistical issues, called enforcing stationarity, on the time

series. For example, in the current work, alfalfa yields tend to be much higher at the beginning of the season, then lower toward the end, so there is a seasonal element that creates a challenge with using AR and MA alone. AR and MA work best on stationary data, or data that has the same distribution in any two equal size windows of time, and seasonal data like alfalfa data typically does not do so. To mitigate this and introduce stationarity, one may consider the differences between values in the time series instead of the actual values, and this is the I step in ARIMA. Any meaningful description of an ARIMA model should be followed by its parameters in parentheses, such as ARIMA(1, 0, 0), where 1, 0, and 0 represent values for the parameters p, d, and q respectively. The number of AR lags is p, the number of MA lags is q, and the number of times differencing is performed is d. SARIMA is designed to further address seasonality, and it accepts a second set of seasonal parameters as in SARIMA(p, d, q)(P, D, Q, m), where P is the number of seasonal AR lags, Q is the number of seasonal MA lags, D is the number of times seasonal differences are taken, and m is the number of time points per season. Finally, and again roughly speaking, SARIMAX accepts one more parameter, a vector of exogenous variables from which cross-correlations are exploited to improve forecasts. For precise mathematical definitions of these models, and even Python implementation details, one may refer to the above-mentioned text by Vishwas and Patel (B V Vishwas).

## Materials and Methods

We explore three main approaches to predicting alfalfa yields: (1) DA with data synthesis with pretraining (DASP) for estimating past and current yields, (2) an ML-based alfalfa forecasting technique for alfalfa time series, and (3) a combination of our forecasting technique and DA (ForDA). For (1), we show that using DASP can provide better results than trivial DA (TDA), or

simply training with a source that is not local to the target region, and that our scale-invariant tabular synthesizer (SITS) produces better results than established synthesizers. For (2), we compare univariate and multivariate versions of our forecasting technique to the ARIMA family of forecasters, and we show that our ML-based approach can produce more accurate forecasts, especially the multivariate version. For (3), we show that when forecasting future yields with non-local source data, ForDA produces our most accurate forecast, and our SITS synthesizer leads to the best results.

*Data*

The data in this study come from alfalfa variety trial reports published by land-grant universities combined with weather data obtained from the Daymet online tool published by Oak Ridge National Laboratory in Tennessee (TN) in cooperation with NASA (Daymet). The five states of focus for the current work are Michigan (MI), Ohio (OH), South Dakota (SD), Kentucky (KY), and Georgia (GA). Our MI data come from East Lansing, Lake City, and Chatham. Our OH data come from Wooster, North Baltimore, and South Charleston. Our SD data come from Watertown, Highmore, and Beresford. Our KY data come from the city of Lexington, as this is the only location in KY where the University of KY (UKY) grows alfalfa. Our limited GA data come from Athens, GA, as this is the only location in GA where the University of GA grew enough alfalfa to use for time series. We used univariate and multivariate time series models, the latter including average per-harvest yields, total accumulated solar radiation in $W/m^2$, total accumulated rainfall in mm, and average minimum and maximum temperatures, over the life of each seeding which is usually three or four years. The time series in the current work consist of three or four-month years, which are the summer months during which alfalfa is harvested, usually June, July, August, and September. We only use three datapoints per year in our experiments using ForDA

with source Watertown and target Highmore, for example, because that is how the data was reported in those locations. We only consider rainfed alfalfa, not artificially irrigated, and we avoid "Roundup-Ready" varieties, or those treated with the herbicide glyphosate. We chose Athens because GA is our home state, and Athens produced the best variety trial reports in GA. Also, using GA and KY extends our previous work with those states, and they are both southern states with similar climates. We chose MI and OH because they are neighboring states in the north, where temperatures are colder, and data is abundant there. We chose SD because it is a third region of the U.S., the northwest, so it adds variety to our study, and data is also abundant there.

Local training and testing on one city or town's grow site is ideal, when possible, but when data scarcity prohibits training good models on the local source, a trivial approach to DA (TDA), or non-local training, has been shown to be useful. As one important end goal for this project is for our techniques to power an interactive software application called Predict Your CropS (PYCS), this work envisions real-world farmers as the end users. Since these real-world farmers will likely have smaller, less curated datasets than land-grant universities, accessing enough local data for ML training will likely be necessary for end users. Furthermore, synthesizing non-local training data to increase sample sizes can lead to more accurate estimators than TDA. Initially, we used CTGAN and TVAE to synthesize data as in the previous work, but the current work extends that technique from classification only to predicting continuous values or regression (Vance, Rasheed and Missaoui). However, those regression results on average, were underwhelming, and they motivated us to design our own SITS algorithm, which is explained in the DASP subsection.

*Packages and Models*

For estimation and forecasting experiments, we mainly used the Python programming language with the Jupyter Notebook coding interface (Foundation) (Kluyver). The main package

used for ML models was Scikit-learn, and the main package we used for the ARIMA family of time series models was SkTime, which provides an interface for Statsmodels and is compatible with Scikit-learn (Pedregosa) (Király) (Seabold) (Pedregosa) (Király) (Seabold). We also used the Weka application for some early experiments, and while it performed well, Weka's advantages were not sufficient to convince us to alter our established workflow and Python integration (Eibe Frank). We ran many experiments with deep neural networks (DNNs) using Tensorflow Keras, but again, its results were comparable to those of the models already integrated into our workflow using Scikit-learn models, so we did not include them in our final battery of systematic experiments (Martín Abadi). The models we worked with were random forest (RF), k-nearest neighbors (KNN), decision tree (DT), multi-layer perceptron (MLP), linear regression (LR), Bayesian ridge regression (BRR), support vector regressor (SVR), and XGBoost.

For all ML experiments except ForDA and those involving DASP, where we used only XGBoost, we validated our models by taking the average of multiple training runs controlled with a tunable parameter that usually varied between 4 and 10. Within those, to find the best parameters for each model, we applied grid search with 5-fold cross-validation while training our models, and we scaled our data by removing the mean and scaling the values to unit variance. As XGBoost showed good results and fast training times in previous work by Vance et al. when training on synthesized datasets in the tens-of-thousands, we focused on this model during preliminary experiments to determine if SITS had value (Vance, Rasheed and Missaoui). We tested these synthesizers on target datasets from OH having hundreds of samples.

*DASP*

Our DA with data synthesis and pretraining (DASP) technique consists mainly of two steps, which are the synthesis and the pretraining. First, a tabular data synthesizer creates a new, larger

dataset from the original training set, and this becomes the DA source data. Second, that new data is used to pretrain an XGBoost model. In our experiments, we trained and tested on the target data with a training/test split that is dictated by the forecast horizon in time series experiments or approximately 70/30 (usually closer to 66/33 due to small datasets) in our strictly DASP experiments. In those, we also used leave-one-out cross-validation to ensure that our results were not anecdotal, where we rotated the training and testing among the three locations in Ohio (OH) mentioned above. We report results from experiments using DA with synthesis alone as well as from experiments with DASP in the results section.

SITS is a simple algorithm that requires labeled yield data. In previous work, Vance et al. described how they labeled yield data according to the number of standard deviations to test ML's ability to classify very small target datasets (Vance, Rasheed and Missaoui). We take that same approach to label yield data in the current work, and our early experiments showed that continuing to use three classes rather than more led to the best results. SITS accepts the input dataset to be synthesized plus two manually tunable parameters, a float *s* and integer *n*. The *n* parameter is the desired size of the output dataset we wish to synthesize or generate. The *s* parameter is the maximum number of standard deviations above or below the mean between which the final synthesized randomly generated value will be. Our experiments have shown that SITS usually works best with $0.1 <= s <= 0.9$, but it sometimes works well with $s >= 1$. For each class label in the input dataset, a percentage of the input that this class represents is calculated and multiplied by *n* to determine how many records to generate for each class. For each record to be generated in that class, a value is calculated for each feature. That value is a random value bounded by the mean value for that feature plus or minus the product of the standard deviation for that feature and *s*.

When this is done for each feature, the new record is added to the synthesized dataset, until all new records are added. Figure 5.1 shows the pseudocode for SITS.

```
# d -> input dataset to be synthesized
# int n -> number of records in output data
# float s -> max stds +/- the mean between which
# synthesized randomly generated value will be

func SITS(d, n, s):
    output = null

    for each class in d:
        potcii = d.percentageOfThisClassInInput()
        amt_to_synth = potcii * d.size()

        for amt_to_synth:
            new_record = null
            for each feature f:
                mean = f.mean()
                std = f.std()
                new_f_val = random(min: mean - std * s,
                                   max: mean + std * s)
                new_record.add(new_f_val)

            output.add(new_record)

    return output
```

Figure 5.1. SITS pseudocode.

*ML-based Forecasting*

In our ML-based forecasting technique, we exploit stationarity by training models on differences between annual yields at each of the three or four corresponding time points. For example, given four yields per year from 2000 to 2005, instead of training using the yields from those years, we would train on the difference between yield 1 in 2000 and yield 1 in 2001, the difference between yield 2 in 2000 and yield 2 in 2001, and so on for yields 3 and 4, then so on for each year in the training window. This introduces some level of stationarity in the training data

(Du Preez). Our early experiments training with the yields themselves left room for improvement, and as the literature shows that enforcing stationarity is often helpful, we follow the above approach, then we predict the expected differences in the forecast horizon (FH) years. Finally, we take the average yield at each time point across all the training years, and we add the predicted differences to those values to produce the final forecast.

*ForDA*

Our novel forecasting with DA (ForDA) works by starting with the DASP technique, which uses a synthesizer to generate extra training data, then uses that to pretrain an XGBoost model. Next, we use the booster function in Scikit-learn's XGBoost model to provide new training data from the target location, and then we use the model to forecast the target test data. We split the target data between training on past years and testing on the final forecast year. As the results section details, ForDA may lead to more accurate forecasts than ML-based forecasting alone, and it produces some of the most promising results and lowest sMAPE scores in the current work.

*Metrics*

The current work focuses on sMAPE scores when forecasting because this is a fairly standard metric used in recent related literature such as work with forecasting and ML by Javeri et al. and Toutiaee et al. (Javeri) (Toutiaee), and it has been one of the metrics used by the M3 and other Makridakis series competitions (Makridakis) as well as by Taieb et al., to name a few (Taieb SB). Some advantages of measuring forecast accuracy with sMAPE are that it is a percentage, so where disparate locations exhibit different levels of yields, this unitless percentage makes them comparable to each other, whereas mean absolute error (MAE) is in tons per acre in our domain,

which may not always be comparable between locations. The symmetric aspect of sMAPE is that it ostensibly treats forecasting too high and forecasting too low equally, while non-symmetric mean absolute percent error (MAPE) does not attempt to be fair in this way. Roughly speaking, sMAPE is symmetric in that it should produce the same error percentage if, for example, the truth is 30 but the forecast is 50, as if the truth is 50 but the forecast is 30 (Goodwin).

While we settled on sMAPE as our metric of focus for forecasting, we measure the accuracy of our estimators of past and current yields mainly with correlation coefficients (R) and mean absolute errors (MAE) because of related literature that motivated this work and to which we compare our results, and our previous results, all use those metrics. As sMAPE scores are less common in related literature using ML to estimate crop yields, we did not report them in all our estimation experiments (Chlingaryan). Furthermore, our forecasting experiments revealed that R paints a confusing picture when used to measure prediction accuracy. For example, when used to measure the correlation between truth and predictions for a series of datapoints unrelated by time, closer to R = 1 is better, and close to R = -1 may still be useful; however, with time series, R can be very close to 1 or -1 without being particularly accurate or inaccurate, and in our experiments R seems to predominantly reflect whether the direction between values at adjacent time points, up or down, is correct but not the actual difference.

## Results

First, we present results from DA with data synthesis experiments, comparing several ML models in a set of target and source locations with leave-one-out cross-validation. Tables 5.1, 5.2, 5.3 and 5.4 present these DA results. Second, we present results from experiments with our ML-based forecaster compared to ARIMA family models. Tables 5.5, 5.6, 5.7, and 5.8 present these time

series forecast results. Finally, Tables 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, and 5.15 present results demonstrating that ForDA can produce better forecasters than ML-based forecasting without DA, and that SITS beats established synthesizers in this task. We present scatterplots to provide a visual sense of the estimators' quality and whether they over-predict or under-predict, and we present time plots for a few representative forecasters.

Table 5.1 depicts our preliminary results using DASP and synthesizing training datasets of 50,000 samples generated from an initial dataset of about 2,000 records from SD. We chose 50,000 because we saw improving results up to around that point but diminishing results after. Table 5.1 shows that our SITS algorithm is competitive with or superior to CTGAN and TVAE in this domain. While CTGAN sometimes beats SITS anecdotally, SITS on average generates datasets that train more accurate estimators than the others. Table 5.1 does not reflect forecasting or use time series data, but trains on a separate source and estimates unseen values anywhere in the timeline of the target data. Estimating these past and current crop yields can help reveal which states may be good candidates for ForDA and provide a steppingstone toward designing that technique. Table 5.1 is approximately a 66/33 training/test split, with pre-training in all of SD followed by the boost step in XGBoost with a subset of Wooster, OH data.

Table 5.1. SITS vs TVAE, CTGAN. XGBoost pre-training, 20k samples, source: SD, target: Wooster, OH, 32 samples. Estimating past & current yields. SITS has best average R (bold).

| **Synth** | **Avg R** | **Avg MAE** | **Avg sMAPE** |
|---|---|---|---|
| SITS | **0.68** | 0.48 | 30.24 |
| TVAE | 0.53 | **0.42** | **25.86** |
| CTGAN | 0.54 | 0.47 | 29.41 |
| TDA | 0.25 | 0.60 | 36.87 |

For the next round of experiments, we trained and tested on locations that are within-state, but still non-local, and we skipped the boosting step because early experiments showed no benefit from boosting with further within-state data versus training with it all at once. We also included a comparison of TDA along with the same three flavors of synthesizers as before. In these experiments, we chose to generate 10,000 samples because early tests showed increasing gains up to that point but diminishing results after on this dataset. Our source training data came from all of SD except the town of Highmore, and our target test data came from Highmore, SD. The results from these closer source and target neighbors were also promising, but not quite as strong as more remote SD to OH results with the boosting step. As Table 5.2 shows, SITS again generates datasets that train more accurate estimators than CTGAN or TVAE, not only on average but the best overall as well. CTGAN and TVAE beat TDA, however. SITS average R was 0.63, and the average MAE was 0.43. Figure 5.2 depicts corresponding scatterplots for these results.

Table 5.2. SITS vs TVAE, CTGAN, TDA. XGBoost w/no pre-training, 10K samples. Source: SD, target: Highmore, SD. Estimating past & current yields. SITS beats others (bold).

| **Synth** | **Avg R** | **Avg MAE** | **Avg sMAPE** |
|---|---|---|---|
| SITS | **0.63** | **0.43** | **31.26%** |
| TVAE | 0.36 | 0.48 | 32.65% |
| CTGAN | 0.30 | 0.47 | 32.50% |
| TDA (none) | 0.34 | 0.70 | 41.76% |

For our last and most systematic round of DA experiments, we used leave-one-out cross-validation (LOOCV) to demonstrate average estimation accuracies over many locations, to identify any locations that stood out as significantly weak or strong trainers or targets, and to identify the best and worst models and synthesizers. For conciseness, we do not include every training and testing combination or every package and model with which we experimented. For

example, we experimented with Tensorflow Keras deep neural networks (DNNs), but they produced results roughly equivalent to our best Scikit-learn models, so we did not include them in our leave-one-out tests. Similarly, leave-one-out validation revealed Weka models to perform about as well as their Scikit-learn counterparts. Occasionally, Weka models produce better results, but extensive experimentation leads us to conclude that Scikit-learn models work as good or better on average and are generally good enough to focus on for the remainder of our work. We generated 20,000 samples for all leave-one-out tests. Table 5.3 details our LOOCV tests for OH, and Table 5.4 shows the averages. Though TVAE with LR had the highest anecdotal R score of 0.61, overall SITS results in slightly better R and MAE scores than TVAE, while CTGAN came in last. The three OH locations tested in Table 5.3 are Wooster (W), North Baltimore (NB), and South Charleston (SC).

Table 5.3. DA with Synthesis, averages for OH with LOOCV. 20,000 training samples, best in bold. Wooster (W), North Baltimore (NB), and South Charleston (SC).

| **SITS R / MAE** | **TVAE R / MAE** | **CTGAN R / MAE** | **Model** | **Source : Target** |
|---|---|---|---|---|
| **0.48 / 0.46** | 0.39 / 0.54 | 0.16 / 0.64 | XGB | W, NB : SC |
| **0.52 / 0.50** | 0.38 / 0.53 | 0.15 / 0.71 | KNN | W, NB : SC |
| **0.56 / 0.51** | 0.55 / 0.51 | 0.46 / 0.59 | BRR | W, NB : SC |
| **0.57 / 0.51** | **0.61 / 0.52** | 0.32 / 0.67 | LR | W, NB : SC |
| **0.42 / 0.59** | 0.41 / 0.60 | 0.21 / 0.64 | XGB | W, SC : NB |
| **0.39 / 0.59** | 0.30 / 0.64 | 0.20 / 0.64 | KNN | W, SC : NB |
| 0.26 / 0.67 | **0.27 / 0.66** | 0.27 / 0.73 | BRR | W, SC : NB |
| 0.26 / 0.67 | **0.27 / 0.66** | 0.26 / 0.62 | LR | W, SC : NB |
| **0.47 / 0.49** | 0.20 / 0.54 | 0.24 / 0.60 | XGB | SC, NB : W |
| **0.45 / 0.50** | 0.28 / 0.51 | 0.15 / 0.52 | KNN | SC, NB : W |
| 0.44 / 0.97 | **0.46 / 0.43** | 0.24 / 0.55 | BRR | SC, NB : W |
| 0.44 / 0.97 | **0.45 / 0.44** | 0.46 / 0.51 | LR | SC, NB : W |

Table 5.4. DA with Synthesis. Averages of best LOOCV in Table 5.3.

| SITS<br>R / MAE | TVAE<br>R / MAE | CTGAN<br>R / MAE | Model |
|---|---|---|---|
| **0.46 / 0.51** | 0.33 / 0.56 | 0.20 / 0.63 | XGB |
| 0.45 / 0.53 | 0.32 / 0.56 | 0.17 / 0.62 | KNN |
| 0.42 / 0.72 | 0.43 / 0.53 | 0.32 / 0.62 | BRR |
| 0.42 / 0.72 | 0.44 / 0.54 | 0.35 / 0.60 | LR |

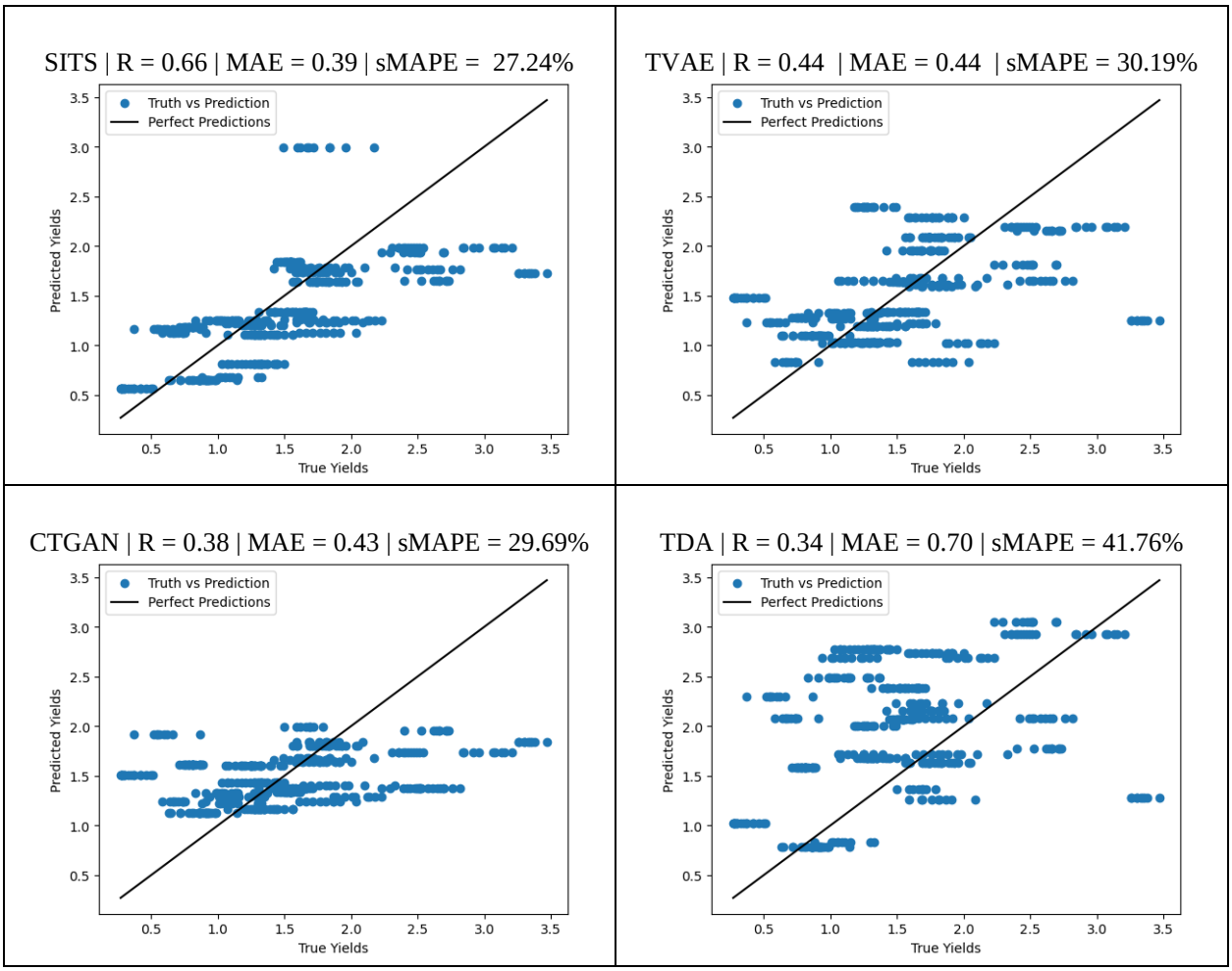

Figure 5.2. Scatterplots from Table 5.2's best. DA w/SITS fits the line best, TDA the worst.

The remaining results are from time series experiments, where forecasters are trained only on historical data to predict yields in future years that the model has not previously seen. We highlight select time plots in Figures 5.3 through 5.11, and we include an exhaustive series of time

plots and ARIMA parameters in Appendix D. Table 5.5 depicts results from our preliminary univariate time series experiments comparing our ML-based technique to ARIMA. We used one long time window from 1999 to 2010 for training and forecasted 2011 alfalfa yields. The resulting sMAPE scores show that our ML-based model produced more accurate forecasts than ARIMA or SARIMA, with BRR producing the best average score of sMAPE = 16.94%.

Tables 5.5 and 5.6 depict our preliminary experiments with univariate and multivariate time series respectively in Beresford, SD. Table 5.6 compares the results of the univariate version of our ML-based technique with ARIMA and SARIMA, and Table 5.7 compares the multivariate version of our technique with the multivariate version of ARIMA called SARIMAX. Again, we use one long training time window of 1999 to 2010, and the resulting sMAPE scores indicate that the ML-based approach produces more accurate forecasts than ARIMA, SARIMA, and SARIMAX. The multivariate approaches produce better results with almost every model, which is expected since they are provided with more training information. These experiments omit years 2002, 2005, and 2006, as these years report three cuts instead of the normal four.

Table 5.5. Univariate time series. Beresford, SD, 1999 to 2010 training to forecast 2011, best in bold.

| **Model** | **sMAPE** | **R** | **MAE** |
|---|---|---|---|
| ARIMA(30,1,15) | 47.52 | 0.73 | 1.10 |
| SARIMA(1,0,0)(5,0,0,4) | 34.67 | 0.83 | 0.71 |
| RF | 20.76 | 0.69 | **0.30** |
| KNN | 18.18 | **0.87** | 0.34 |
| DT | 24.97 | 0.77 | 0.33 |
| XGB | 27.86 | 0.24 | 0.39 |
| MLP | 21.58 | **0.87** | 0.33 |
| LR | 17.68 | **0.87** | 0.33 |
| BRR | **16.94** | **0.87** | 0.32 |
| SVR | 20.51 | **0.87** | 0.32 |

Table 5.6. Multivariate time series. Beresford, SD, 1999 to 2010 training to forecast 2011, best in bold.

| **Model** | **sMAPE** | **R** | **MAE** |
|---|---|---|---|
| SARIMAX(1,0,0)(5,0,1,4) | 28.08 | **0.98** | 0.59 |
| RF | 15.58 | 0.83 | 0.25 |
| KNN | **12.81** | 0.95 | 0.25 |
| DT | 36.47 | 0.86 | 0.70 |
| XGB | 34.59 | 0.89 | 0.51 |
| MLP | 14.25 | 0.89 | **0.23** |
| LR | 26.57 | 0.88 | 0.48 |
| BRR | 16.87 | 0.87 | 0.32 |
| SVR | 18.88 | 0.88 | 0.38 |

Table 5.7 summarizes the results of our more systematic univariate time series experiments using a sliding window, or rolling validation with multiple forecast horizons, which is a common validation approach in time series literature (Javeri) (Toutiaee). We used this sliding window of training data to compare several ML models trained on only one feature, alfalfa yield in tons per acre. Arguably, the yield's position in the time series may be thought of as a second feature. We compare the results of our univariate approach to the results of the univariate statistical model known as autoregressive integrated moving average, or ARIMA, and its seasonal counterpart SARIMA. Table 5.7 shows that our ML-based technique almost always produces more accurate forecasts than ARIMA and SARIMA. This table presents symmetrical mean absolute percent errors (sMAPEs) from a six-year sliding window in Beresford, SD, with a forecast horizon (FH) of 1 year at a time. It depicts forecasts starting with source 1999 to 2007 and target 2008, then the source years' window slides forward one year at a time up to forecasting 2011. We use training windows of six years because that is the width required before our ML-based forecasting technique begins to show an advantage over ARIMA and SARIMA. With training windows of smaller than six years, ARIMA models usually forecasted more accurately in our experiments. We optimized

the p and q and seasonal P and Q parameters through experimentation for each step of the sliding windows, so these parameters are different for each row, and they are detailed in Appendix D. The ARIMA family of models, in our experiments, exhibit a threshold of about six lags beyond which the results become worse instead of better as more lags are added; however, we provided ARIMA with the same available lags as the ML models and SARIMA with the same available seasonal lags as the ML models to be fair. As Table 5.7 shows, our ML-based forecasting technique is consistently competitive with or more accurate than ARIMA and SARIMA as measured by sMAPE, across every sliding window tested. Highlights include SVR with sMAPE = 13.77%, RF with sMAPE = 19.97%, and KNN with sMAPE = 18.53% on training window 2003 to 2010. Table 5.7 shows sMAPE scores on top, R on bottom. Figures 5.3 and 5.4 depict time plots associated with our forecasters.

Table 5.7. Univariate sliding window validation results. Beresford, SD. 4 training windows of 6 years each forecast following year from 2008 to 2011. Top 3 average results in bold.

| **Target Year** | **ARIMA** | **SARIMA** | **KNN** | **DT** | **SVR** | **XGB** | **MLP** | **RF** | **LR** | **BRR** |
|---|---|---|---|---|---|---|---|---|---|---|
| 2008 | 49.47<br>0.62 | 39.05<br>0.88 | 42.22<br>0.99 | 36.62<br>0.99 | 37.04<br>0.74 | 39.38<br>0.79 | 43.18<br>0.99 | 41.22<br>0.93 | 41.01<br>0.99 | 37.46<br>0.99 |
| 2009 | 34.96<br>0.28 | 50.64<br>0.13 | 37.45<br>0.18 | 36.92<br>0.51 | 46.08<br>0.16 | 54.01<br>0.36 | 32.86<br>0.16 | 33.80<br>0.26 | 43.81<br>0.16 | 46.20<br>0.16 |
| 2010 | 29.80<br>0.86 | 14.43<br>0.90 | 27.71<br>0.85 | 59.04<br>0.44 | 25.89<br>0.82 | 71.04<br>0.34 | 26.83<br>0.85 | 39.34<br>0.65 | 23.65<br>0.85 | 22.56<br>0.85 |
| 2011 | 29.52<br>0.59 | 25.91<br>0.79 | 18.53<br>0.91 | 24.25<br>0.48 | 13.77<br>0.85 | 24.14<br>0.48 | 31.04<br>0.85 | 19.97<br>0.76 | 24.10<br>0.85 | 20.83<br>0.86 |
| **Average** | 35.94<br>0.59 | 32.51<br>0.68 | **31.48**<br>0.73 | 39.21<br>0.48 | **30.70**<br>0.64 | 47.14<br>0.49 | 33.48<br>0.71 | 33.58<br>0.65 | 33.14<br>0.71 | **31.76**<br>0.71 |

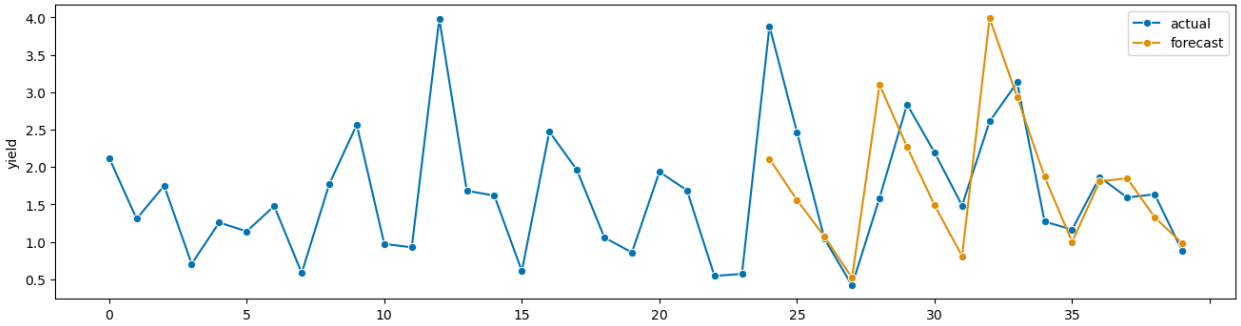


Figure 5.3. Time plot, Table 5.7 SVR. Forecasts 2008 – 2011, true alfalfa yields 1999 - 2011.

Table 5.8 shows the results of the multivariate version of our sliding-window validation experiments, where we compare models trained on more than one feature. The established model against which we compare the multivariate version of our ML-based forecasting technique is SARIMA with exogenous variables, or SARIMAX. We compare results from the same ML models as in the univariate version. These data include precipitation, solar radiation, and temperature, and the location is Beresford, SD from 1999 to 2011, omitting 2002, 2005, and 2006 because those years did not feature four cuts. Table 5.8 shows that our ML-based technique with stationarity results in more accurate forecasts than SARIMAX. Top row scores are sMAPE, bottom row scores are R. RF was the top scorer in these tests, with an average sMAPE of 22.38% over all windows.

Table 5.8. Multivariate sliding window validation results. Beresford, SD. 4 training windows of 6 years each forecast following year from 2008 to 2011. Top 3 average results in bold.

| Target Year | SARIMAX | KNN | DT | SVR | XGB | MLP | RF | LR | BRR |
|---|---|---|---|---|---|---|---|---|---|
| 2008 | 29.33<br>0.97 | 27.47<br>0.96 | 9.08<br>0.99 | 29.61<br>0.99 | 28.24<br>0.98 | 30.90<br>0.99 | 17.23<br>0.98 | 43.79<br>0.84 | 32.13<br>0.99 |
| 2009 | 27.29<br>0.30 | 35.45<br>0.11 | 34.52<br>0.85 | 30.37<br>0.20 | 38.43<br>0.78 | 26.83<br>0.27 | 27.16<br>0.45 | 36.33<br>0.17 | 45.11<br>0.16 |
| 2010 | 29.54<br>0.54 | 16.06<br>0.87 | 48.18<br>0.49 | 21.70<br>0.84 | 36.61<br>0.92 | 18.93<br>0.86 | 23.39<br>0.89 | 31.36<br>0.86 | 22.16<br>0.85 |
| 2011 | 40.00<br>0.79 | 23.80<br>0.71 | 44.13<br>0.99 | 21.89<br>0.86 | 52.26<br>0.86 | 20.04<br>0.76 | 21.72<br>0.94 | 24.99<br>0.86 | 19.29<br>0.86 |
| **Average** | 31.54<br>0.65 | **25.70**<br>0.66 | 33.98<br>0.83 | 25.89<br>0.72 | 38.89<br>0.89 | **24.18**<br>0.72 | **22.38**<br>0.82 | 34.12<br>0.68 | 29.67<br>0.72 |

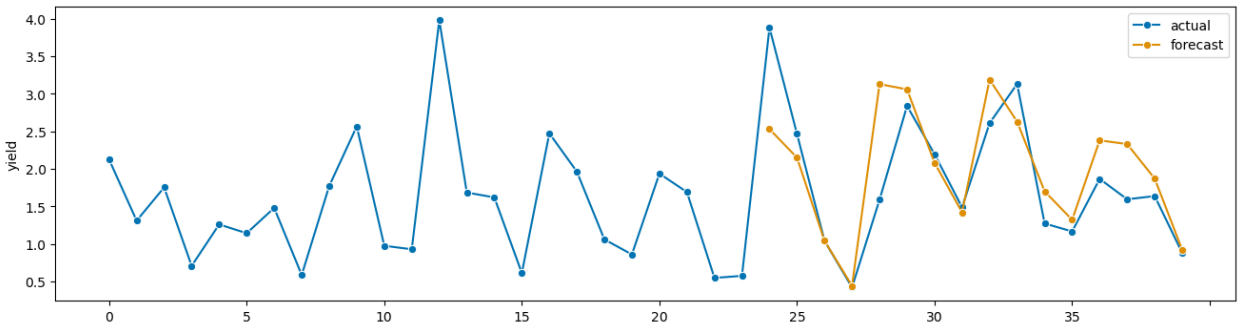


Figure 5.4: Time plot, Table 5.8 RF. Forecasts 2008 – 2011, true alfalfa yields 1999 - 2011.

Table 5.9 presents results from ML-based forecasting with DA (ForDA), which produced our best results, like sMAPE = 9.81%. It shows that ForDA beats ML-based forecasting without DA, and our SITS synthesizer leads to more accurate forecasts than CTGAN or TVAE. The source data come from Watertown, SD 1999 to 2011 (no 2003 due to insufficient data), and the target is Highmore, SD 1999 to 2011 (no 2000 to 2003, 05, 06, 09 due to insufficient data). When data synthesis is useful due to data scarcity, these synthesis techniques will likely improve results over

forecasting with TDA as previous experiments have done. Table 5.10 presents results from the same technique, but with source Highmore and target Watertown, for validation. Figures 5.5 and 5.6 depict the time plots for our most successful ForDA runs in Table 5.9 and Table 5.10, respectively. Table 5.11 depicts the averages of Tables 5.9 and 5.10.

Table 5.9. ForDA with SITS vs CTGAN, TVAE w/ XGBoost. 20k samples, SITS wins (bold). Source: Watertown, SD 1999 to 2011; Target: Highmore, SD, 1999 to 2011.

| ForDA Synth | sMAPE | R | MAE |
|---|---|---|---|
| SITS | **10.22** | **0.87** | **0.21** |
| TVAE | 16.75 | 0.74 | 0.33 |
| CTGAN | 15.38 | 0.75 | 0.29 |

Table 5.10. ForDA with SITS vs CTGAN, TVAE w/ XGBoost. 20k samples, SITS wins (bold). Source: Highmore, SD 1999 to 2011; Target: Watertown, SD, 1999 to 2011.

| ForDA Synth | sMAPE | R | MAE |
|---|---|---|---|
| SITS | **9.81** | **0.94** | **0.14** |
| TVAE | 19.20 | 0.92 | 0.26 |
| CTGAN | 18.50 | 0.90 | 0.25 |

Table 5.11. ForDA with SITS vs CTGAN, TVAE w/ XGBoost averages. 20k samples, SITS wins (bold). Averages of Tables 5.10 and 5.11.

| ForDA Synth | sMAPE | R | MAE |
|---|---|---|---|
| SITS | **10.01** | **0.90** | **0.18** |
| TVAE | 17.98 | 0.83 | 0.30 |
| CTGAN | 16.94 | 0.83 | 0.27 |

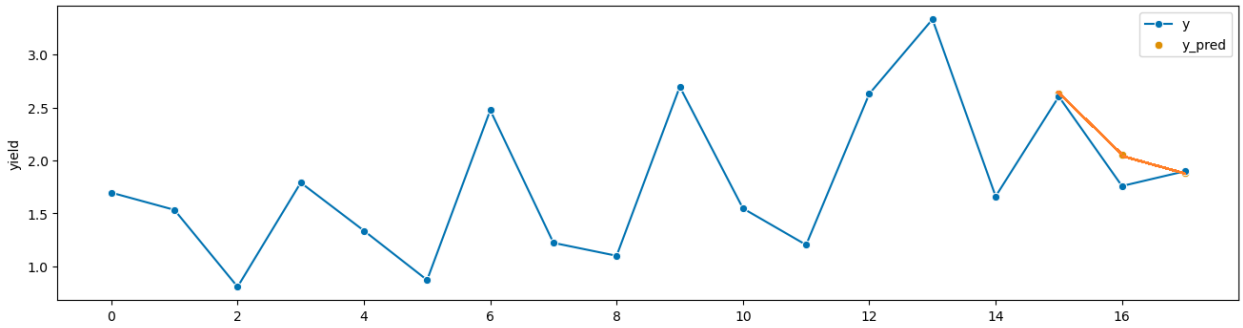


Figure 5.5: Best results from ForDA, Source: Watertown, SD 1999 to 2011; Target: Highmore, SD 1999 to 2011, sMAPE = 5.86%.

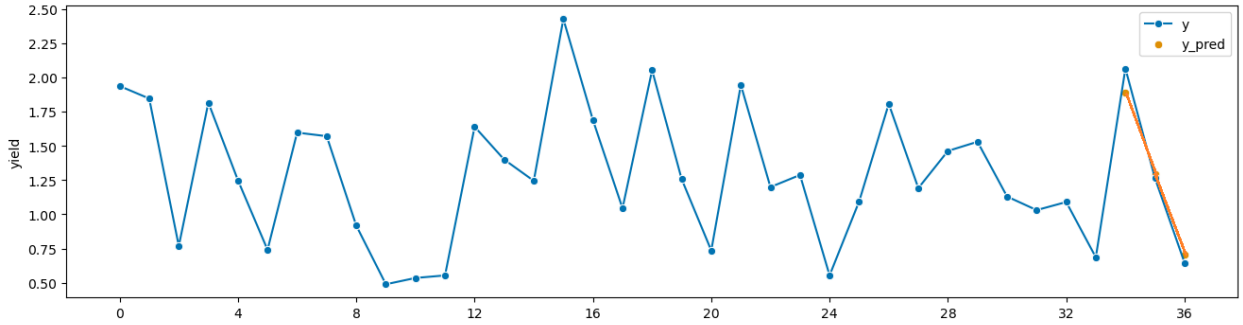


Figure 5.6. Best results from ForDA Source: Highmore, SD 1999 to 2011; Target: Watertown, SD, 1999 to 2011 sMAPE = 6.72%.

Table 5.12 shows average results from round-robin style experiments with ForDA in OH in Wooster (2011 to 2018), North Baltimore (2010 to 2018 without 2015 or 2017 due to insufficient data), and South Charleston (2010 to 2019 without 2017 due to insufficient data), where we give each one a turn at being the source and another its target. These experiments showed the best results on much smaller synthesized datasets, so we settled on 30 samples, and we decided to continue with SITS since earlier experiments convinced us that it produces better results than CTGAN or TVAE in our domain. The bottom row shows averages of each metric over all round-

robin experiments, so it represents our average forecast accuracy in OH. The Table 5.12 forecasts are not as striking as those where SD is the source and Beresford, SD is the target, but they are better than our ML-based forecasts without DA, and they are relatively strong forecasts. Also, Table 5.12 depicts four-point seasons. Figures 5.7 and 5.8 show the best two single runs, which produced sMAPE = 12.71%, MAE = 0.90 tons/acre, and R = 0.79 with source North Baltimore and target South Charleston, and (Figure 5.7), and sMAPE = 11.45%, MAE = 0.22 tons/acre, and R = 0.88 (Figure 5.8) with source Wooster and target South Charleston. Since Table 5.12 presents ForDA results with pre-training, all these source/target pairs use XGBoost; however, to determine whether pre-training helps, we also experimented with these same targets and the other seven models using synthesis only and no pre-training. We include those results and their corresponding time plots in Appendix E, and they show that pre-training almost always helps, and skipping it produces mostly inferior results, with overall average sMAPE = 25.70%, MAE = 0.38 tons/acre, and R = 0.79.

Table 5.12. ForDA results using SITS. Round-robin experiments in OH. Best 2 in bold. 2011 to 2018 in Wooster, 2010 to 2018 in North Baltimore, 2010 to 2019 in South Charleston.

| **Source : Target** | **SITS(*s*) parameter** | **sMAPE** | **MAE** | **R** |
|---|---|---|---|---|
| Wooster : North Baltimore | 0.5 | **16.47** | **0.19** | **0.98** |
| Wooster : South Charleston | 1.5 | 16.55 | 0.30 | 0.82 |
| North Baltimore : Wooster | 1.5 | 26.51 | 0.51 | 0.21 |
| North Baltimore : South Charleston | 1.0 | 18.33 | 0.55 | 0.76 |
| South Charleston : Wooster | 2.0 | 22.84 | 0.44 | 0.26 |
| South Charleston : North Baltimore | 0.5 | **16.54** | **0.23** | **0.86** |
| **Average** | - | 19.54 | 0.37 | 0.65 |

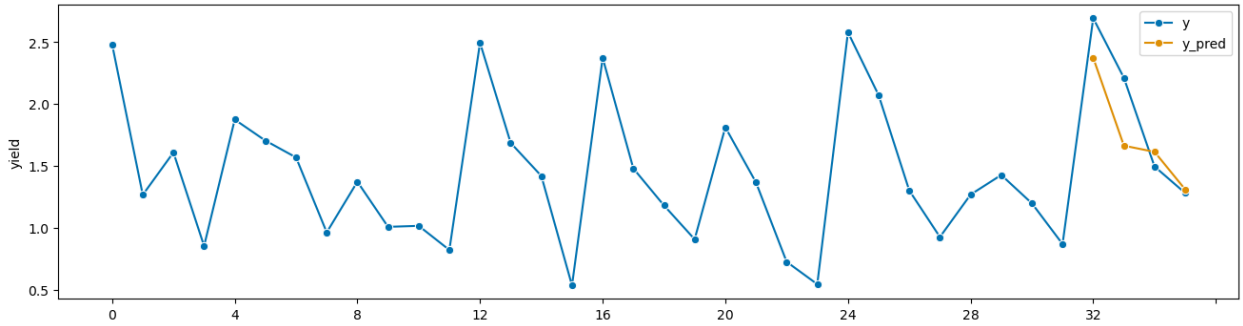


Figure 5.7. 2nd best run in OH round-robin tests. Source: North Baltimore 2010 to 2018, target: South Charleston 2010 to 2019 forecasting 2019, sMAPE = 12.71%, MAE = 0.90 tons/acre, R = 0.79.

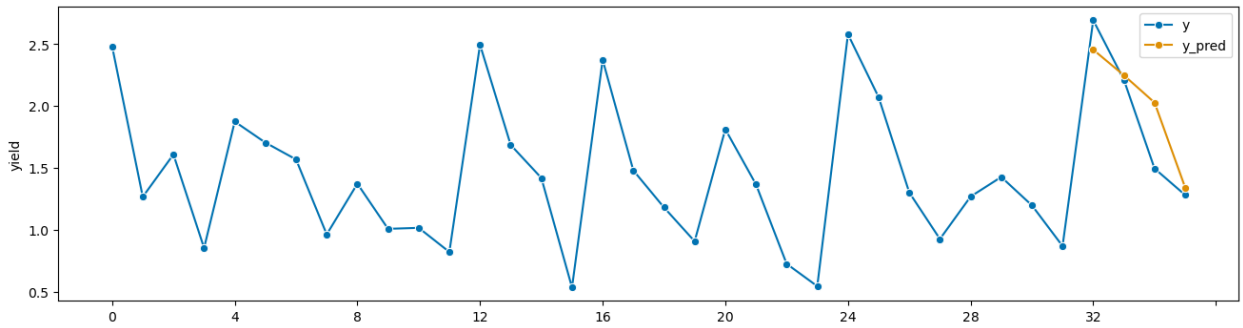


Figure 5.8. Our best run in OH round-robin tests. Source: Wooster 2011 to 2018, target: South Charleston 2010 to 2019, forecasting 2019, sMAPE = 11.45%, MAE = 0.33 tons/acre, R = 0.88.

Next, we ran a series of experiments designed to push the limits of ForDA, where we stacked time series from multiple disparate locations to see where ForDA's benefits may end, and whether the source must always be geographically near the target. First, we tried combining time series from several disparate states to predict yields in Athens, GA, where there are only three seasons of variety trials (2008 to 2010), making it a particularly difficult target. Our source was MI (1999 to 2022), OH (2010 to 2019), SD (1999 to 2011), and KY (2012 to 2018) combined,

with some years omitted due to insufficient data as previously noted. We found that our forecasts for Athens, GA are very poor, and that synthesizing data in these experiments did not improve results. The reason for this is almost certainly the lack of adequate data in Athens, especially for time series. As Figure 5.9 suggests, Athens also lacks any clear visually identifiable patterns. We performed further experiments with Athens as the target, both with East Lansing, MI as the source and Lexington, KY as the source, but these also produced poor forecasts, both with and without synthesis. Synthesis resulted in better R scores with source East Lansing, but otherwise synthesis did not clearly help. Table 5.13 shows these results.

As Table 5.14 shows, we next tried LOOCV at the state level for sources and the local level for targets, since it is not feasible to forecast per-cut at the state level, as cut dates, weather, and number of cuts vary within the state. We combined three states for each source and again experimented with and without synthesis, rotating among MI, OH, SD, and KY. Those results were promising in at least the case of target North Baltimore, OH, which produced a sMAPE = 15.06%, MAE = 0.16 tons/acre, and R = 0.98. Figure 5.10 depicts the best run from these tests, with target North Baltimore and sMAPE = 11.89%. The results in Table 5.14 were otherwise fair and much better than those with target Athens. Also, SITS data synthesis clearly improved results in these tests, with an average sMAPE = 25.85%, MAE = 0.36 tons/acre, and R = 0.75 with synthesis versus sMAPE = 33.62, MAE = 0.49 tons/acre, and R = 0.65 without synthesis.

Table 5.15 shows results from experiments with OH and MI, which we tested against each other because they are neighboring states both with plenty of data, suggesting that they may be good DA sources for each other. While those tests produced better results than those with Athens, GA as the target, they did not produce auspicious results with or without synthesis, and synthesis failed to improve these forecasts. The best average results from those experiments used source OH

without synthesis and target East Lansing, MI, with sMAPE = 25.63%, MAE = 0.49 tons/acre, and R = 0.99. However, as Figure 5.11 depicts, our anecdotally best model actually did use SITS, synthesizing from source OH and forecasting target East Lansing, MI, producing sMAPE = 25.38 and MAE = 0.33 tons/acre.

Table 5.13. Target Athens, GA. Poor forecasts due to very scarce target data. 2008 to 2010. With and without data synthesis.

| **Source : Target** | ***s* param** | **samples** | **sMAPE** | **MAE** | **R** |
|---|---|---|---|---|---|
| MI, OH, SD, KY : Athens, GA, no synth | - | - | 32.37 | 0.49 | 0.04 |
| MI, OH, SD, KY : Athens, GA | 1.5 | 700 | 49.62 | 0.53 | 0.40 |
| East Lansing, MI : Athens, GA, no synth | - | - | 48.70 | 0.62 | -0.43 |
| East Lansing, MI : Athens, GA | 1.2 | 100 | 51.40 | 0.68 | -0.45 |
| Lexington, KY : Athens, GA, no synth | - | - | 41.76 | 0.56 | -0.30 |
| Lexington, KY : Athens, GA | 1.5 | 80 | 42.65 | 0.56 | -0.22 |
| **Average no synth** | - | - | 40.94 | 0.56 | -0.23 |
| **Average w/synth** | - | - | 47.89 | 0.59 | -0.09 |

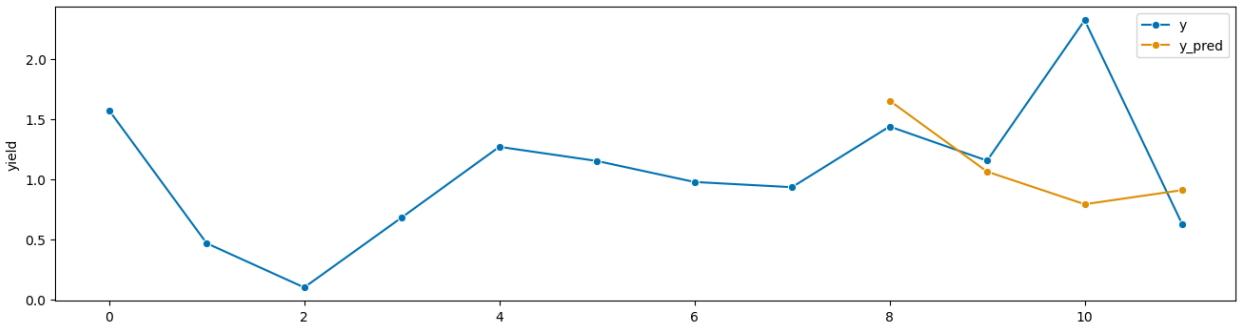


Figure 5.9. Source: KY, Target: Athens, GA. SITS w/80 samples synthesized, *s* = 1.5; sMAPE = 39.32, MAE = 0.53, R = -0.13. 2008 to 2010, forecasting 2010.

Table 5.14. LOOCV with MI (1999 to 2022), OH (2010 to 2019), SD (1999 to 2011), and KY (2012 to 2018).

| **Source : Target** | **SITS(*s*) param** | **samples** | **sMAPE** | **MAE** | **R** |
|---|---|---|---|---|---|
| MI, OH, SD : KY | 1.2 | 500 | 26.55 | 0.28 | 0.51 |
| MI, OH, SD : KY no synth | - | - | 19.82 | 0.19 | 0.71 |
| MI, SD, KY : NB, OH | 0.3 | 500 | **15.06** | **0.16** | **0.98** |
| MI, SD, KY : NB, OH no synth | - | - | 25.47 | 0.35 | 0.99 |
| OH, SD, KY : MI | 1.5 | 400 | 25.21 | 0.40 | 0.94 |
| OH, SD, KY : MI no synth | - | - | 39.39 | 0.56 | 0.88 |
| MI, OH, KY : SD | 1.2 | 500 | 36.59 | 0.59 | 0.55 |
| MI, OH, KY : SD no synth | - | - | 49.79 | 0.84 | 0.00 |
| **Average w/synth** | - | - | **25.85** | **0.36** | **0.75** |
| **Average no synth** | - | - | 33.62 | 0.49 | 0.65 |

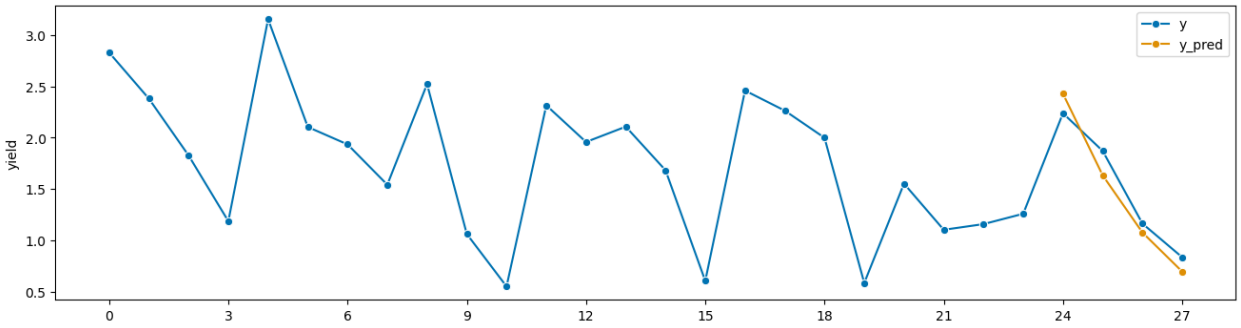


Figure 5.10. Source: MI, SD, KY, Target: OH. SITS w/500 samples synthesized, *s* = 0.3, sMAPE = 11.89, MAE = 0.16, R = 0.98. Forecasting 2019.

Table 5.15. Source: MI, Target : OH ; and vice-versa.

| **Source : Target** | ***s* param** | **samples** | **sMAPE** | **MAE** | **R** |
|---|---|---|---|---|---|
| MI : Wooster, OH, no synth | - | - | 29.65 | 0.59 | 0.26 |
| MI : Wooster, OH | 0.9 | 600 | 34.75 | 0.62 | 0.24 |
| OH : East Lansing, MI, no synth | - | - | 25.63 | 0.49 | 0.99 |
| OH : East Lansing, MI | 1.5 | 168 | 32.30 | 0.47 | 0.91 |
| **Average no synth** | - | - | 27.64 | 0.54 | 0.63 |
| **Average w/synth** | - | - | 33.53 | 0.55 | 0.58 |

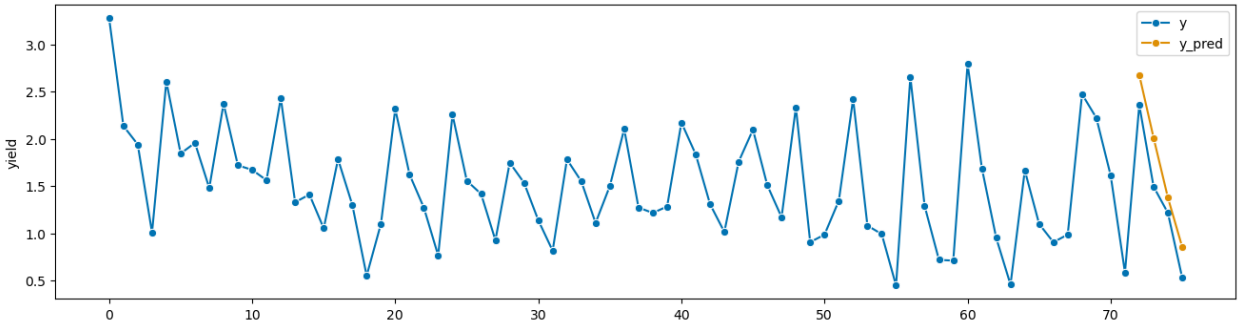


Figure 5.11. Source: OH, Target: East Lansing, MI. sMAPE = 25.38, MAE = 0.33 tons/acre, R = 0.98. Forecasting 2022.

## Discussion

Over the course of this work, we show marginal promise with the early DA experiments, but we take advantage of it to power the final ForDA technique, which yields very promising results.

One phenomenon we observe is that while models in the ARIMA family provide diminishing returns after considering too many lags in our experiments, our ML-based forecasting technique appears to only improve more as the training window widens. Another interesting observation is that ARIMA sometimes outperforms SARIMA, so trying to account for seasonality is not always helpful. On the other hand, though alfalfa yields display some seasonality, looking at our time plots reveals that the trends are not always consistent.

Until the current work, the authors have focused on estimating past and current yields as tabular data. In that problem, R and coefficient of determination ($R^2$) scores closer to 1 are the best and usually indicate a good model that makes accurate predictions. However, one significant take away from the time series experiments in the current work is that R is not nearly as informative when evaluating time series models. Our results show many R scores very close to 1, but that may be misleading. Instead, we focus on the sMAPE scores with time series, as they tell us more. On the

other hand, we continue to report R because if the sMAPE is good but R is wrong, the plot will not look convincing, which is also very important.

Readers may note that our results for forecasting the future are sometimes better than results for estimating past and current yields, which may seem counterintuitive; however, better forecasting results are expected when they are locally trained, while the estimators are trained on non-local data. Even though our ForDA technique pretrains on non-local source data, it ultimately trains on a small subset of data local to the target area. As the current authors showed in previous work, local training to estimate past and current yields is still arguably the most accurate, as one would expect, reporting $R^2$ scores over 0.98 (J. R. Vance).

While the results herein suggest that our multivariate ML-based forecasting technique and ForDA produce more accurate forecasts than the well-established SARIMAX model, this may be a little like comparing apples to oranges. We hypothesize that our technique wins because it is fitted to the forecast horizon's weather features during testing to make its predictions, so it has an advantage over SARIMAX, which only looks at exogenous variables in the lags. On the other hand, the vision for the current project has always been to build a what-if tool like PYCS, and the "ifs" are the weather features in the forecast horizon, whether they are hypothetical or known, and our results indicate that our techniques can forecast "what-if" with high accuracy. We elaborate on our vision for the Predict Your CropS (PYCS) what-if tool in Chapter 6. We observe that XGBoost often performs best with several thousand training samples, but it is not usually a top performer when the data size is very limited. On the other hand, when we use plenty of training data for pretraining and take advantage of the booster function, XGBoost becomes the winner. Overall, it would make this work easier if all the historical yield data was more consistent and tightly controlled, and if it went back further. While six years is enough training for our technique

to beat SARIMAX, that small window might not reveal the potential of ML-based forecasting, and more data and consistency would likely help.

When we pushed ForDA to its limits, using non-local and state-level sources to forecast targets in disparate regions, we found that ForDA is still potentially useful, but not as accurate as when sources are in-state. Therefore, and not too surprisingly, it is best to use in-state or otherwise very nearby sources and targets. While data synthesis and our SITS algorithm led to higher average accuracies than without synthesis in most experiments, synthesis did not demonstrate a clear advantage in those more challenging validation experiments. However, data synthesis continued to produce our anecdotally best models, and that may be important to pay attention to, as we can save these pre-trained models and reuse them in the final PYCS application, and we will likely keep and reuse those that lead to the anecdotally best results. Especially in the case of Athens, GA, it is clear that too-small training datasets will result in poor forecasters, and ForDA cannot remedy this.

## Conclusions

This work shows that ML can produce more accurate forecasts of alfalfa yields than some traditional models, and that DA can help improve those forecasts further. Also, we show that data synthesis is useful in helping build better estimators and forecasters, and that pre-training with a model like XGBoost can help improve results further. This suggests that these techniques could be used to forecast other crops, or other time series with similar properties to ours. Possibly the most important contribution of this work is ForDA, which produced forecasts with sMAPE scores below 6% at best, and 10.22% on average.

Early in this research, we emphasized R and $R^2$ scores as indicators of strong estimators. However, the later time series work painted a clear picture of why one cannot rely on correlation coefficients and coefficients of determination alone. Some of our time series experiments produced very high R scores but poor MAEs, and graphing the corresponding time plots revealed that when yields go up or down at the same time as the forecast goes up or down, high R scores result, even if the actual predicted values are significantly far off from the truth.

This work highlights the need for a more unified effort and standardized practices in conducting and reporting alfalfa variety trials. Many variety trial reports publish fall dormancy scores for varieties, but many do not, and it might be easier to take advantage of fall dormancy data if all trials published it. This is the case with fertilization information as well, which is sometimes available, and sometimes not. Similarly, some variety trials publish alfalfa yields as each variety's yield percentage of the vernal variety. Most studies publish crop yield in tons per acre, but some states, like Minnesota, publish a wealth of data going back for many years, but they do not appear to publish their yields in tons per acre, making it difficult to combine or use along with other states' data.

The authors hope the science presented here will be useful and power the resulting crop yield forecasting application. We have shown that our techniques could be potentially useful in providing the underlying science that powers a what-if tool, as our multivariate forecasts are based on features, real or hypothetical, in the future. Developing a fully automated version of the PYCS what-if tool would likely require a team of dedicated software developers, but this may be possible now that there is an underlying technique to support it. While the current work focuses on alfalfa yields, the final product could most likely be used to predict any crop or any other measurable metric of crop health.

# CHAPTER 6

# PYCS: PREDICT YOUR CROPS WHAT-IF FORECASTING TOOL

## Introduction

Since the initiation of this project, the ultimate goal has been to create a "what-if" tool that farmers can use to help plan for shortages and surpluses based on hypothetical weather scenarios. The high accuracy resulting from our experiments with Forecasting with Domain Adaptation (ForDA) strongly suggests that our work can be useful in that application. In this chapter, we renew our proposal from Chapter 4 for an interactive what-if application called Predict Your CropS (PYCS), but we now demonstrate use cases for PYCS, as its underlying science has since been solidified. We also present updates to the web-based graphical user interface (GUI) now that we have a better understanding of the features it should offer. It is important to note that the multivariate machine learning (ML)-based forecasting techniques proposed in Chapter 5 can only forecast multivariate target data, and it requires that the features are present in the target future forecasting period. In other words, ForDA must know next year's weather, or at least a hypothetical version of next year's weather, in order to forecast next year's yields. Only the univariate version of our ML-based forecaster can forecast next year's yields knowing only past information, and we know of no way to provide other features to our ML models only during training but not in testing. This is not to detract from our multivariate ML-based forecasters, since the ultimate vision for PYCS is that it is a what-if tool, where a hypothetical version of next year's weather features are expected as input for the tool to give useful feedback about hypotheticals. Furthermore, we can be confident that

PYCS what-if forecasts are reasonable, since the underlying ForDA technique provided sMAPE scores around 10% and below, as shown in Chapter 5.

## Application

### *PYCS GUI*

Figures 6.1 and 6.2 depict the current iteration of the PYCS GUI. The user may choose Annualized mode, where only the total yields for all cuts along with annualized weather data for each year are considered, but the application defaults to forecasting multiple cuts per year and expects weather data corresponding to each cut. At least two features are required for PYCS to provide any forecasting, (1) the date of the cut, or year of cut if in annualized mode, and (2) the yield for each cut or each year total if in annualized mode. Next, the user may select univariate forecasting mode, though the application defaults to multivariate forecasting. If univariate mode is selected, the required date or year of cut will be converted to an index and not used in training. If multivariate mode is selected, the date sown must also be provided at a minimum, and including only the dates sown and cut would imply that PYCS needs to lookup the weather data. We would like for future iterations of PYCS to provide a more interactive GUI, but for now, the application is essentially a form that is emailed to the author along with links to the user's data repository. This presents an opportunity to automate the application, which could be done by someone looking for an academic project, or it could potentially present a commercial opportunity. In a fully automated PYCS, a submission request would automatically trigger a job being run on graphic processing units (GPUs), and quick results would be returned. However, the current iteration requires that someone on the backend manually obtains the submitted data and trains and tests the ML models. The user may check a box for each type of feature they are providing, and Yield and Date of Cut are pre-selected since those fields are required at a minimum. In non-annualized mode,

the user must select the number of cuts per season from the dropdown, ranging from two to six cuts per year. This is a required selection, as this is not always obvious from the data provided, which often has years with differing numbers of cuts.

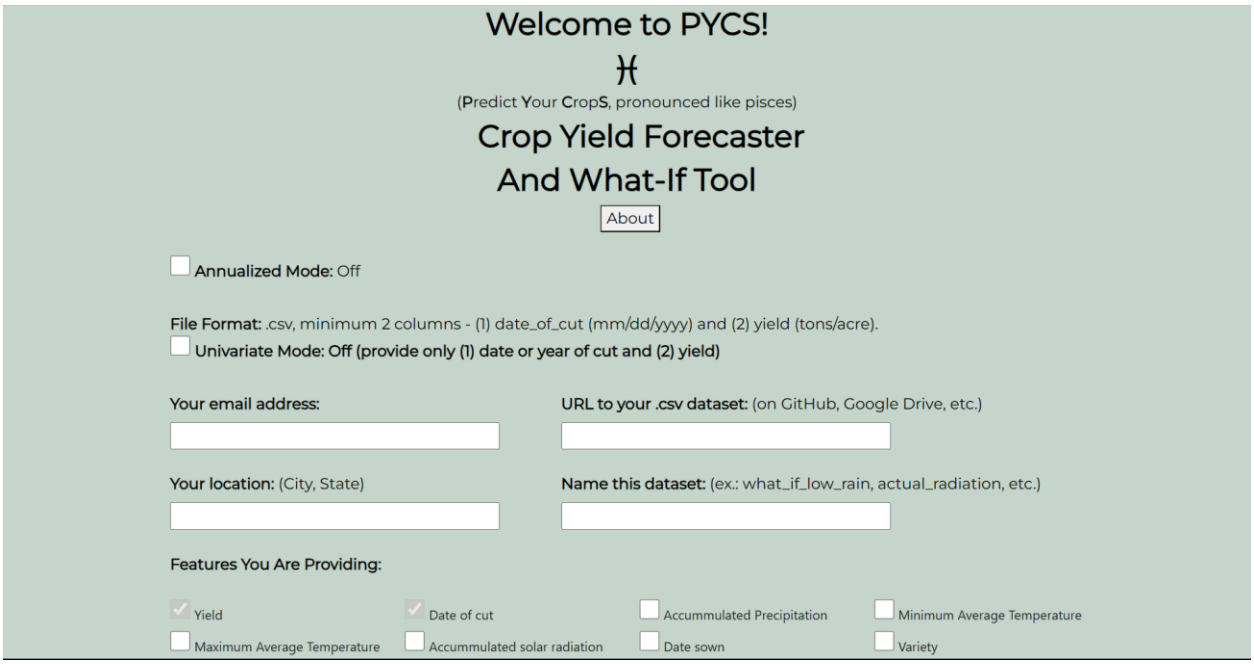


Figure 6.1. Top section of PYCS GUI.

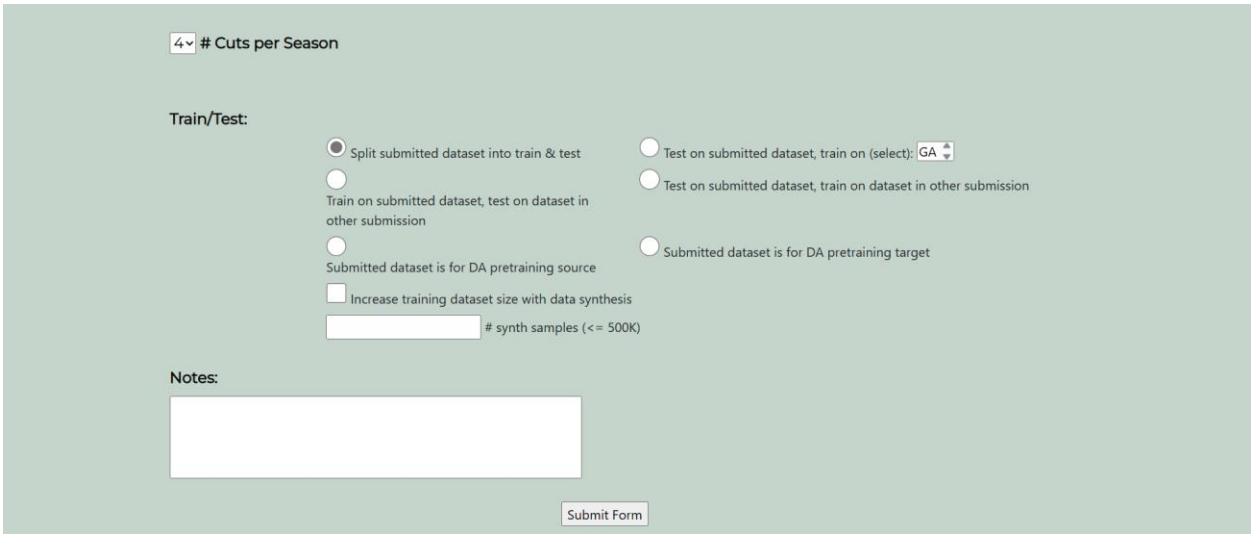


Figure 6.2. Bottom section of PYCS GUI.

The last section of the PYCS GUI provides selections for training and test details. If the user is submitting separate training and testing or source and target datasets, if they want to use DA, or if they are only submitting the target for DA, they will select that here. The user could request that we train and test on a split of the one dataset they provide; they could provide us with a test dataset and request that we train on a specific state or collection of states from our own data; they could provide separate datasets in two separate submissions, one for training and another for testing as indicated by their radio button selection; or, they could specify that their data submission is intended for either DA pretraining or as the DA target. They can choose to synthesize more samples from the provided dataset if it is training or pretraining data, with a current limit of 500,000 samples, which we chose because our experiments have not shown improved results beyond that size, and it is still small enough to process in a few minutes. Finally, we provide a notes section, so users can provide other details and help us learn what other features this developing system should provide.

*Example PYCS Use Cases*

In this section, we describe how one might use PYCS as a what-if tool. An end-user might complete the PYCS form and point our team to a dataset containing hypothetical precipitation records, and PYCS would return to them a forecast based on those “what-ifs”. In multivariate mode, for any feature provided in the training data, that feature must also be included in the test data, and no features beyond those included in the training data may be used in testing, due to the nature of ML. In future iterations, the user may not be required to provide historical weather data, because it is possible to automate retrieving historical weather data from the Daymet API; however, the user would still need to provide the hypothetical future feature values to match the

historical features provided or requested (Daymet). For now, if the user cannot provide weather data, the system authors would have to curate and aggregate historical weather data manually, though that task can be very time-consuming. It is impossible to test the accuracy of a forecast based on hypothetical weather data, unless that same weather data happened to play out in nature and we could compare our hypothetical yields to that, which is highly unlikely to occur. Instead, we assume that since our forecasts for real weather scenarios show promising accuracy, our hypothetical forecasts are also reasonable.

To test PYCS, we fictionalize data in the forecast year, using values that are within reason based on other data featuring realistically higher and lower than normal values. We first isolate each feature, fictionalizing hypothetical high and low values for radiation, then precipitation, then temperature. Table 6.1 shows results where hypothetical solar radiation is higher than normal. Roughly 10,000 to 13,000 $W/m^2$ of solar radiation accumulates in a normal month in this region, so we tested PYCS output when solar radiation accumulates at an increased rate of roughly 15,000 $W/m^2$ per month, or about 2,000 to 5,000 $W/m^2$ more than normal for each cut. The following results use South Charleston, OH as the source and North Baltimore, OH as the target, and it uses SITS with $s = 0.5$ and 100 synthesized samples, as we determined these to be moderate parameters in previous experiments. As Table 6.1 shows, boosting the hypothetical solar radiation this way usually leads to higher yield in the first and last cuts, but lower yield in the middle two cuts, and a lower total yield for the season. We are not strict with our hypothetical input values because we are interested in general trends, and naturally occurring weather would not be uniform.

Table 6.1. Hypothetical higher radiation. Leads to forecasting a mix of higher and lower than actual yields, but overall lower.

| **Actual Rad.** | **Hypothetical Rad.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 297866.6 | 300866.6 | 2.24 | 2.60 |
| 310590.4 | 315590.4 | 1.87 | 1.23 |
| 323710.5 | 330710.5 | 1.16 | 0.99 |
| 333422.2 | 345422.2 | 0.84 | 0.97 |
| - | **Total Yield:** | 6.11 | 5.79 |

Table 6.2 shows the same locations as Table 6.1, but these hypothetical solar radiation totals are about 5,000 $W/m^2$ lower than the actual values for that season. This leads to another mix of higher and lower yields than actual, but the total annual yield is higher.

Table 6.2. Hypothetical lower radiation. Leads to forecasting a mix of higher and lower than actual yields, but overall higher.

| **Actual Rad.** | **Hypothetical Rad.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 297866.6 | 294866.6 | 2.24 | 2.48 |
| 310590.4 | 305590.4 | 1.87 | 1.53 |
| 323710.5 | 318710.5 | 1.16 | 1.17 |
| 333422.2 | 325422.2 | 0.84 | 1.20 |
| - | **Total Yield:** | 6.11 | 6.38 |

Tables 6.3 and 6.4 show more extremes in high and low solar radiation. Since the previous results were somewhat mixed, but they seem to suggest that too much solar radiation leads to low yields and lower amounts lead to higher yields, we thought it would be interesting to try very high and very low solar radiation. As Tables 6.3 and 6.4 confirm, this trend continues more clearly when radiation levels are very high and very low.

Table 6.3. Hypothetical very low radiation. Leads to forecasting overall higher than actual yields.

| **Actual Rad.** | **Hypothetical Rad.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 297866.6 | 287866.6 | 2.24 | 2.46 |
| 310590.4 | 292590.4 | 1.87 | 1.53 |
| 323710.5 | 297710.5 | 1.16 | 1.30 |
| 333422.2 | 302422.2 | 0.84 | 1.25 |
| - | **Total Yield:** | 6.11 | 6.54 |

Table 6.4. Hypothetical very high radiation. Leads to forecasting mostly lower than actual yields.

| **Actual Rad.** | **Hypothetical Rad.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 297866.6 | 310866.6 | 2.24 | 2.42 |
| 310590.4 | 330590.4 | 1.87 | 1.23 |
| 323710.5 | 360710.5 | 1.16 | 0.49 |
| 333422.2 | 390422.2 | 0.84 | 0.40 |
| - | **Total Yield:** | 6.11 | 4.55 |

Tables 6.5, 6.6, 6.7, and 6.8 show hypothetical crop yields resulting from hypothetical high and low precipitation. Tables 6.5 and 6.6 show the results of nominal increases in precipitation, while Tables 6.7 and 6.8 show the results of more extremes in precipitation. As expected, more rain can lead to higher yields, but we were somewhat surprised to see how quickly high precipitation becomes too much precipitation and results in lower yields. Normal precipitation accumulation in this region is roughly 100mm per month. Table 6.5 shows the results of a 20mm increase in precipitation per cut, and this causes most cuts to yield more, and it causes a total annual increase in yield. Table 6.6 shows the results of a 50mm increase in hypothetical precipitation per cut, which leads to higher yields than the 20mm increase. Table 6.7 shows the results of an approximate 100mm increase in hypothetical precipitation per cut, which leads to the highest forecast yields yet. However, Table 6.8 shows forecasts based on a 200mm increase in precipitation per cut, which appears to be too much moisture for the crops and reduces their yield substantially. We do, however, see an increase in the forecast for the first cut, but this is typical and expected, since it is

based on a 100mm increase over the entire winter, which is not a big increase. For these experiments, we generated 200 samples with SITS and $s$ = 0.4, which we determined to be moderate parameters in earlier experiments. Table 6.9 shows the results of a decrease in precipitation of about 80 to 85mm, beginning with about 100mm lower than actual in the first cut, which leads to mostly lower forecasts per cut and a substantially lower annual total.

Table 6.5. Hypothetical +10 to 20mm high precipitation. Leads to forecasting slightly higher overall than actual yields.

| **Actual Precip.** | **Hypothetical Precip.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 2811.76 | 2831.76 | 2.24 | 2.37 |
| 2968.34 | 2988.34 | 1.87 | 1.53 |
| 3057.78 | 3077.78 | 1.16 | 1.16 |
| 3177.75 | 3197.75 | 0.84 | 1.06 |
| - | **Total Yield:** | 6.11 | 6.12 |

Table 6.6. Hypothetical +50mm high precipitation. Leads to forecasting higher than actual yields.

| **Actual Precip.** | **Hypothetical Precip.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 2811.76 | 2861.76 | 2.24 | 2.52 |
| 2968.34 | 3018.34 | 1.87 | 1.86 |
| 3057.78 | 3107.78 | 1.16 | 1.17 |
| 3177.75 | 3227.75 | 0.84 | 0.92 |
| - | **Total Yield:** | 6.11 | 6.45 |

Table 6.7. Hypothetical +100mm high precipitation. Leads to forecasting higher than actual yields.

| **Actual Precip.** | **Hypothetical Precip.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 2811.76 | 2911.76 | 2.24 | 2.72 |
| 2968.34 | 3168.34 | 1.87 | 1.89 |
| 3057.78 | 3357.78 | 1.16 | 1.37 |
| 3177.75 | 3577.75 | 0.84 | 1.07 |
| - | **Total Yield:** | 6.11 | 7.05 |

Table 6.8. Hypothetical +200mm high precipitation. Leads to forecasting lower than actual yields.

| **Actual Precip.** | **Hypothetical Precip.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 2811.76 | 3011.76 | 2.24 | 2.60 |
| 2968.34 | 3368.34 | 1.87 | 1.18 |
| 3057.78 | 3657.78 | 1.16 | 0.58 |
| 3177.75 | 3977.75 | 0.84 | 0.78 |
| - | **Total Yield:** | 6.11 | 5.14 |

Table 6.9. Hypothetical -80mm low precipitation. Leads to forecasting lower than actual yields.

| **Actual Precip.** | **Hypothetical Precip.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 2811.76 | 2711.76 | 2.24 | 2.39 |
| 2968.34 | 2731.34 | 1.87 | 1.34 |
| 3057.78 | 2757.78 | 1.16 | 0.74 |
| 3177.75 | 2777.75 | 0.84 | 0.70 |
| - | **Total Yield:** | 6.11 | 5.17 |

Tables 6.10, 6.11, 6.12, and 6.13 show results from unusually low and high average temperatures, and for simplicity, these examples assume that the average lows and highs are directly correlated. Table 6.10 shows the results of hypothetical average temperatures 5 degrees Celsius (C°) lower than the actual average minimum and maximum temperatures, and this leads to forecasting higher than actual yields. As Table 6.11 shows, when we lower the temperature decrease to 2 C°, we continue to see increased but not as high yield forecasts as with lowering by 5 C°. We see the inverse pattern in Tables 6.12 and 6.13, where we trained using hypothetical yields 5 C° and 2 C° higher than the actual average minimum and maximum temperatures, which forecasted lower than actual yields. The negative total annual effect on yield is more severe with the 5 C° increase than with the 2 C°. This is consistent with alfalfa variety trials in the U.S., where colder northern states usually grow more alfalfa and publish more data than southern states.

Table 6.10. Hypothetical -5 C° low average temperatures. Leads to forecasting higher than actual yields.

| **Actual Avg Min/Max** | **Hypothetical Avg Min/Max** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 5.510158 / 16.04933 | 0.510158 / 11.04933 | 2.24 | 2.76 |
| 5.923658 / 16.46616 | 0.923658 / 11.46616 | 1.87 | 2.06 |
| 6.306201 / 16.86261 | 1.306201 / 11.86261 | 1.16 | 1.68 |
| 6.632579 / 17.15108 | 1.632579 / 12.15108 | 0.84 | 1.43 |
| - | **Total Yield:** | 6.11 | 7.93 |

Table 6.11: Hypothetical -2 C° low average temperatures. Leads to forecasting higher than actual yields.

| **Actual Avg Min/Max** | **Hypothetical Avg Min/Max** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 5.510158 / 16.04933 | 3.510158 / 14.04933 | 2.24 | 2.66 |
| 5.923658 / 16.46616 | 3.923658 / 14.46616 | 1.87 | 2.07 |
| 6.306201 / 16.86261 | 4.306201 / 14.86261 | 1.16 | 1.31 |
| 6.632579 / 17.15108 | 4.632579 / 15.15108 | 0.84 | 1.10 |
| - | **Total Yield:** | 6.11 | 7.14 |

Table 6.12. Hypothetical +5 C° high average temperatures. Leads to forecasting lower than actual yields.

| **Actual Avg Min/Max** | **Hypothetical Avg Min/Max** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|
| 5.510158 / 16.04933 | 10.510158 / 21.04933 | 2.24 | 1.63 |
| 5.923658 / 16.46616 | 10.923658 / 21.46616 | 1.87 | 1.10 |
| 6.306201 / 16.86261 | 11.306201 / 21.86261 | 1.16 | 0.78 |
| 6.632579 / 17.15108 | 11.632579 / 22.15108 | 0.84 | 0.50 |
| - | **Total Yield:** | 6.11 | 4.01 |

Table 6.13. Hypothetical +2 C° high average temperatures. Leads to forecasting lower than actual yields.

| Actual Avg Min/Max | Hypothetical Avg Min/Max | Actual Yields | Forecast Yields |
|---|---|---|---|
| 5.510158 / 16.04933 | 7.510158 / 18.04933 | 2.24 | 2.23 |
| 5.923658 / 16.46616 | 7.923658 / 18.46616 | 1.87 | 1.66 |
| 6.306201 / 16.86261 | 8.306201 / 18.86261 | 1.16 | 0.33 |
| 6.632579 / 17.15108 | 8.632579 / 19.15108 | 0.84 | 0.47 |
| - | **Total Yield:** | 6.11 | 4.69 |

Tables 6.14 and 6.15 show predictions from increasing hypothetical precipitation while decreasing hypothetical temperatures, then decreasing hypothetical precipitation while increasing hypothetical temperatures. First, we trained using precipitation starting the season 100mm higher than actual, then accumulating 150mm between cuts instead of the typical 100mm, and we used temperatures 1 C° lower than actual. As Table 16.14 shows, this moderately higher than normal precipitation combined with moderately lower than normal temperatures leads to forecasting higher than normal yields each cut, which is expected from these moderate changes. Next, we trained using precipitation starting the season 100mm lower than actual, then accumulating only about 5mm between cuts, about 95mm less than normal, and we used temperatures 2 C° higher than actual. As Table 6.15 shows, this drought combined with very high temperatures leads to higher-than-normal yields in the first half of the season followed by much lower-than-normal yields in the second half, and a slightly higher total annual yield. We expect the models to have difficulty understanding the meaning of the precipitation at the first cut, since the precipitation we used, though lower than actual, could be understood to be high if trained on instances where seeding had taken place a year later. Meanwhile, the high average temperature, though detrimental in summer, may indicate to the model that the preceding winter was warm enough for alfalfa to

begin growing early that season. Therefore, it makes sense that the first cuts would be higher than normal, but the drought and high temperatures would eventually take a toll on the yield.

Table 6.14. +50mm precipitation, -1 C° average temperatures. Leads to forecasting higher than actual yields.

| **Actual Avg Min/Max Temp** | **Hypoth. Avg Min/Max Temp** | **Actual Precip.** | **Hypoth. Precip.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|---|---|
| 5.510158 / 16.04933 | 4.510158 / 15.04933 | 2811.76 | 2911.76 | 2.24 | 2.75 |
| 5.923658 / 16.46616 | 4.923658 / 15.46616 | 2968.34 | 3061.34 | 1.87 | 1.96 |
| 6.306201 / 16.86261 | 5.306201 / 15.86261 | 3057.78 | 3217.78 | 1.16 | 1.41 |
| 6.632579 / 17.15108 | 5.632579 / 16.15108 | 3177.75 | 3367.75 | 0.84 | 1.18 |
| - | - | - | **Total Yield:** | 6.11 | 7.30 |

Table 6.15. -95mm precipitation, +2 C° average temperatures. Leads to forecasting mix of higher and lower than actual yields.

| **Actual Avg Min/Max Temp** | **Hypoth. Avg Min/Max Temp** | **Actual Precip.** | **Hypoth. Precip.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|---|---|
| 5.510158 / 16.04933 | 7.510158 / 18.04933 | 2811.76 | 2611.76 | 2.24 | 3.05 |
| 5.923658 / 16.46616 | 7.923658 / 18.46616 | 2968.34 | 2615.34 | 1.87 | 2.49 |
| 6.306201 / 16.86261 | 8.306201 / 18.86261 | 3057.78 | 2617.78 | 1.16 | 0.57 |
| 6.632579 / 17.15108 | 8.632579 / 19.15108 | 3177.75 | 2620.75 | 0.84 | 0.26 |
| - | - | - | **Total Yield:** | 6.11 | 6.37 |

Tables 6.16 and 6.17 show predictions from increasing hypothetical solar radiation while decreasing hypothetical precipitation, then decreasing hypothetical solar radiation while increasing hypothetical precipitation. First, we trained using the same increase in precipitation as in the experiment shown in Table 6.14, and we decreased solar radiation by about 10,000 $W/m^2$ less than

normal per cut. As Table 6.16 shows, this led to forecasting substantially higher than normal yields, which is expected, as moderate precipitation increase and radiation decrease led to higher yields in earlier experiments. Next, we increased solar radiation by about 15,000 W/m$^2$ per cut, beginning the season with a 30,000 W/m$^2$ increase, and we decreased precipitation by about 95mm per cut. As Table 6.17 shows, this led to a substantial decrease in yields, which one would expect when combining a drought with unusually high solar radiation.

Table 6.16. +50mm precipitation, -10,000 W/m$^2$ solar radiation. Leads to forecasting higher than actual yields.

| **Actual Rad.** | **Hypoth. Rad.** | **Actual Precip.** | **Hypoth. Precip.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|---|---|
| 297866.6 | 267866.6 | 2811.76 | 2911.76 | 2.24 | 2.53 |
| 310590.4 | 272590.4 | 2968.34 | 3061.34 | 1.87 | 1.99 |
| 323710.5 | 277710.5 | 3057.78 | 3217.78 | 1.16 | 1.57 |
| 333422.2 | 282422.2 | 3177.75 | 3367.75 | 0.84 | 1.03 |
| - | - | - | **Total Yield:** | 6.11 | 7.12 |

Table 6.17. -95mm precipitation, +15,000 W/m$^2$ solar radiation. Leads to forecasting lower than actual yields.

| **Actual Rad.** | **Hypoth. Rad.** | **Actual Precip.** | **Hypoth. Precip.** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|---|---|
| 297866.6 | 330866.6 | 2811.76 | 2611.76 | 2.24 | 2.18 |
| 310590.4 | 340590.4 | 2968.34 | 2615.34 | 1.87 | 1.38 |
| 323710.5 | 373710.5 | 3057.78 | 2617.78 | 1.16 | 1.01 |
| 333422.2 | 403422.2 | 3177.75 | 2620.75 | 0.84 | 0.81 |
| - | - | - | **Total Yield:** | 6.11 | 5.38 |

Finally, we tried decreasing hypothetical precipitation as in the previous experiment and increasing solar radiation and temperature as before, and this led to the mixed season forecast shown in Table 6.18. As in previous tests, the season starts with high yields which are probably due to the model's lack of understanding the relative meaning of high or low rain at the start of the season, plus the suggestion of a warm and sunny winter that indicates alfalfa could grow early. However, the drought conditions and overly sunny and warm weather take their toll on the yields by the end of the season, when yields become very low.

Table 6.18. -95mm precip., +15,000 W/m$^2$ rad., +2 C° avg. temp. Leads to forecasting a mix of lower and higher than actual yields.

| **Actual Rad.** | **Hypoth. Rad.** | **Actual Precip.** | **Hypoth. Precip.** | **Actual Avg Min/Max Temp** | **Hypoth. Avg Min/Max Temp** | **Actual Yields** | **Forecast Yields** |
|---|---|---|---|---|---|---|---|
| 297866.6 | 330866.6 | 2811.76 | 2611.76 | 5.510158 / 16.04933 | 7.510158 / 18.04933 | 2.24 | 3.63 |
| 310590.4 | 340590.4 | 2968.34 | 2615.34 | 5.923658 / 16.46616 | 7.923658 / 18.46616 | 1.87 | 2.65 |
| 323710.5 | 373710.5 | 3057.78 | 2617.78 | 6.306201 / 16.86261 | 8.306201 / 18.86261 | 1.16 | 0.31 |
| 333422.2 | 403422.2 | 3177.75 | 2620.75 | 6.632579 / 17.15108 | 8.632579 / 19.15108 | 0.84 | 0.23 |
| - | - | - | - | - | **Total Yield:** | 6.11 | 6.82 |

Conclusion

We have demonstrated that PYCS, powered by ForDA proposed in Chapter 5, can be a potentially useful tool in helping to forecast biomass yields for the alfalfa crop. It is interesting to note that, according to PYCS, greater solar radiation may decrease yields, which may be counterintuitive to

the notion that more sun makes plants grow more. Also, it is interesting that PYCS successfully predicts that more water, or precipitation, will only increase yields up to a point, after which more precipitation will decrease yields. As a nice bonus in the precipitation experiments, PYCS exhibited a potential ability to help find the point where plenty of rain becomes too much, though we did not necessarily have that feature in mind when designing the tool. Also, PYCS revealed an interesting relationship between temperature and yields, where alfalfa seems to favor cooler temperatures, which is arguably consistent with reality, as most alfalfa variety trial data come from the northern U.S., indicating it grows more successfully in those colder regions. Overall, PYCS shows potential and warrants further research and development. These experiments are limited, and more training data and more results would help reveal more about the application's potential.

# CHAPTER 7

# CONCLUSION

The most important conclusion of the research described in this document is that domain adaptation (DA) can improve machine learning (ML) results in the task of forecasting biomass yields, and our novel techniques can produce high forecasting accuracies. ML has shown promise in an increasing number of domains over the past decade, and this work shows it also has promise in crop yield forecasting. Furthermore, ML-based forecasting accuracy usually beats the well-established, popular family of statistical models like ARIMA, SARIMA, and SARIMAX. Meanwhile, as the United Nations and other world powers acknowledge and combat climate change, the current research offers a small contribution toward that effort as it shows how ML can help keep agricultural practices efficient as the Earth's climate warms and changes. Chapter 3 describes how we initially used ML models with feature selection and simple weather data to estimate past and current alfalfa biomass yields with greater accuracy than previous related work. Chapter 4 describes our novel contributions to the SNLT pipeline, a simple form of DA, it introduces the PYCS application and shows how data synthesis with deep generative networks can increase dataset sizes and help train better models. Chapter 5 describes our novel ML-based crop yield forecasting technique with DA (ForDA) and shows that its results are competitive with or better than well-established statistical models, and Chapter 5 also proposes the SITS data synthesizer, which is competitive with or better than at least two established tabular synthesizers. Chapter 6 proposes an updated GUI for the PYCS application and presents several examples of its potential use as a what-if tool.

We have shown our novel ForDA technique to be very accurate under ideal alfalfa farm management conditions, and our experiments have produced symmetric mean absolute percent error (sMAPE) scores as low as 9.81%, best correlation coefficient (R) scores above 0.90, and best mean absolute error (MAE) scores as low as 0.14 tons per acre, beating traditional, non-ML-based techniques like ARIMA and SARIMAX. ForDA also produced very promising average validated sMAPE scores around 10% and MAEs of 0.18 tons/acre. Related literature has shown forecasting of biomass yields to be an especially difficult problem, so these scores are very encouraging. This work also showed good results in estimating historical and present-day tabular data using data synthesis. In that phase of the project, we proposed a novel tabular data synthesizer we call Scale Invariant Tabular Synthesis (SITS), which helps boost the performance of our ML models by increasing training dataset sizes. We show that our synthesis algorithm leads to R scores over 100% higher than established synthesizers in tests that produce an average R of 0.63 with SITS versus 0.30 with CTGAN. Training with data from one location to estimate historical and present-day data in another location provides insight into which regions can be effectively used to train models to estimate other target regions, especially when the target region's dataset is too small to train its own model. We call this non-local training, and when we include synthesis in the pipeline, we call this approach Synthetic Non-Local Training (SNLT), and it is essentially a form of DA.

This research also presents many opportunities for future work, and this author hopes to see this work live on after the present phase is complete. The PYCS application is still in its initial stages of development, so the opportunity to enhance its GUI and interactivity is clear. In the future, we would like to see a team of professional developers take our primitive GUI from a simple email form to something much more sophisticated and interactive. We envision an

authenticated system that allows only serious users to run forecasting tests on PYCS, as GPU processing can be expensive. It would be better if users were able to upload their data to PYCS rather than pointing the developer to the data at some third-party's repository. Once PYCS is fully automated, users will not have to wait days for results but should be able to get results in as little as a few seconds at best or several minutes at worst. Also, there is potential to develop an application programmer interface (API) so that PYCS may be used programmatically and expose its functionality for others to code against and use in a backend way or create their own user interface for it. There is also potential future work in pre-training an array of models from which PYCS users can choose, and applying user feedback and experimental results to tune and improve those models is a rich potential research area. The end application also requires infrastructure based on graphics processing units (GPUs), presenting another aspect for the project's future evolution. Determining the best way to distribute such resources among users and jobs presents an interesting area for potential future work.

Early in this research, we emphasized R and $R^2$ scores as indicators of strong estimators. However, the later time series work painted a clear picture of why one cannot rely on correlation coefficients and coefficients of determination alone. Some of our time series experiments produced very high R scores but poor MAEs, and graphing the corresponding time plots revealed that when yields go up or down at the same time the forecast goes up or down, high R scores can result, even if the actual predicted values are significantly far off from the truth. On the other hand, we cannot dismiss R as a metric, because we sometimes see results where sMAPE scores are good but R scores are bad, and the resulting time plot looks bad.

As this work is the first this author knows of that presents ML-based crop yield forecasts as time plots of actual and predicted yields with multiple time points forecast, the author hopes

this will inspire other researchers to perform similar experiments with other crops and present their results in a similar way. To summarize the primary contributions of this work: (1) we propose a novel ML-based forecasting technique, (2) we propose a novel DA technique combining the SITS data synthesizer with pre-training, (3) we combine our DA and forecasting techniques into one enhanced novel forecaster, ForDA, which produces very accurate biomass yield forecasts, and (4) we integrate these techniques into the proposed PYCS what-if tool, we provide a GUI for it, and we show its utility with several examples.

# REFERENCES

Copenhagen Consensus Center. *Background*. n.d. 2023 9 5. <https://www.copenhagenconsensus.com/post-2015-consensus/>.

ArunKumar, K.E., Kalaga, D.V., Kumar, C.M.S., Chilkoor, G., Kawaji, M. and Brenza, T.M. "Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-Regressive Integrated Moving Average (SARIMA)." *Applied soft computing* 103 (2021): 107161.

Ayoubi, S. and Sahrawat, K.L. "Comparing multivariate regression and artificial neural network to predict barley production from soil characteristics in northern Iran." *Archives of Agronomy and Soil Science* 57(5) (2011): 549-565.

B V Vishwas, ASHISH PATEL. *Hands-on Time Series Analysis with Python*. Berkeley, CA: Apress, 2020.

Bak, S., Carr, P. and Lalonde, J.F. "Domain adaptation through synthesis for unsupervised person re-identification." *European conference on computer vision (ECCV)*. 2018. 189-205.

Baral, R., Lollato, R.P., Bhandari, K. and Min, D. "Yield gap analysis of rainfed alfalfa in the United States." *Frontiers in Plant Science* (2022): 2492.

Bashar, M.A., Nayak, R., Luong, K. and Balasubramaniam, T. "Progressive domain adaptation for detecting hate speech on social media with small training set and its application to COVID-19 concerned posts. Social Network Analysis and Mining." 11 (2021): 1.

Bellocchio, E., Costante, G., Cascianelli, S., Fravolini, M.L. and Valigi, P. "Combining domain adaptation and spatial consistency for unseen fruits counting: a quasi-unsupervised approach." *IEEE Robotics and Automation Letters* 5(2) (2020): 1079-1086.

Bose, P., Kasabov, N.K., Bruzzone, L. and Hartono, R.N. "Spiking neural networks for crop yield estimation based on spatiotemporal analysis of image time series." *IEEE Transactions on Geoscience and Remote Sensing* 54.11 (2016): 6563-6573.

Cao, J., Katzir, O., Jiang, P., Lischinski, D., Cohen-Or, D., Tu, C. and Li, Y. "Dida: Disentangled synthesis for domain adaptation." *arXiv preprint* arXiv:1805.08019 (2018).

Chen, T. and Guestrin, C. "Xgboost: A scalable tree boosting system." *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, August. 785-794.

Chlingaryan, A., Sukkarieh, S. and Whelan, B. "Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review." *Computers and electronics in agriculture* 151 (2018): 61-69.

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F. and Sun, J. "Generating multi-label discrete patient records using generative adversarial networks." *Machine learning for healthcare conference*. Boston, MA, USA: PMLR, November, 2017. 286-305.

Choudhary, A., Tong, L., Zhu, Y. and Wang, M.D. "Advancing medical imaging informatics by deep learning-based domain adaptation." *Yearbook of medical informatics* 29.01 (2020): 129-138.

Choudhury, A. and Jones, J. "Crop yield prediction using time series models." *Journal of Economics and Economic Education Research* 15.3 (2014): 53-67.

Cunha, R.L., Silva, B. and Netto, M.A. "A scalable machine learning system for pre-season agriculture yield forecast." *2018 IEEE 14th International Conference on e-Science (e-Science)*. IEEE, 2018, October. 423-430.

Daumé III, H. "Frustratingly easy domain adaptation." arXiv preprint arXiv:0907.1815, 2009.

Daymet. *Daymet*. n.d. 12 October 2022. <https://daymet.ornl.gov/>.

Dhore, A., et al. "Weather prediction using the data mining Techniques." *Int. Res. J. Eng. Technol* (2017).

Distribution, Anaconda Software. *Version 4.6.8 anaconda*. n.d. 11 March 2021. <https://anaconda.com>.

DM., Johnson. "An assessment of pre-and within-season remotely sensed variables for forecasting corn and soybean yields in the United States." *Remote Sensing of Environment* 141 ( 2014): 116-128.

Du Preez, J. and Witt, S.F. "Univariate versus multivariate time series forecasting: an application to international tourism demand. International Journal of Forecasting." 19(3) (2003): 435-451.

Duke, S.O. and S.B. Powles. "Glyphosate: A once-in-a-century herbicide." *Pest Manag. Sci. Former. Pestic. Sci.* 64 (2008): 319–325.

Easterbrook, S.M. and Johns, T.C. "Engineering the software for understanding climate change." *Computing in science & engineering* 11(6) (2009): 65-74.

Eibe Frank, Mark A. Hall, and Ian H. Witten. "The WEKA Workbench." *Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*. Morgan Kaufmann, Fourth Edition, 2016.

El-Ramady, Hassan, et al. "Alfalfa growth under changing environments: An overview." *Environment, Biodiversity and Soil Security 4* 2020.4 (2020): 201-224.

Feleke, H.G. "Assessing weather forecasting needs of smallholder farmers for climate change adaptation in the Central Rift Valley of Ethiopia." *Journal of Earth Science and Climate Change* 6(10) (2015): 1-8.

Foundation, Python Software. *Python language reference v3.68*. 2021. 11 March 2021. <https://python.org>.

Frausto-Solis J, Gonzalez-Sanchez A, Larre M. "A new method for optimal cropping pattern." *MICAI 2009: Advances in Artificial Intelligence: 8th Mexican International Conference on Artificial Intelligence*. Guanajuato, México, November 9-13, 2009.

Freund, Y. and Schapire, R.E. "Experiments with a new boosting algorithm." *icml*. 1996, July. 148-156.

G., Ruß. "Data mining of agricultural yield data: A comparison of regression models." *InAdvances in Data Mining. Applications and Theoretical Aspects: 9th Industrial Conference, ICDM*. Leipzig, Germany: Springer Berlin Heidelberg, July 20-22, 2009. 24-37.

Gonzalez-Sanchez, A., Frausto-Solis, J. and Ojeda-Bustamante, W. " Predictive ability of machine learning methods for massive crop yield prediction." *Spanish Journal of Agricultural Research* 12(2) (2014): 313-328.

Goodfellow, I., 2016. "Nips 2016 tutorial: Generative adversarial networks." *arXiv* (2016).

Goodwin, P. and Lawton, R. "On the asymmetry of the symmetric MAPE." *International journal of forecasting* 15(4) (1999): 405-408.

Gopal, P.M. and Bhargavi, R. "A novel approach for efficient crop yield prediction." *Computers and Electronics in Agriculture* 165 (2019): p.104968.

Hendrycks, D., Lee, K. and Mazeika, M. "Using pre-training can improve model robustness and uncertainty." *International Conference on Machine Learning*. PMLR, 2019, May. 2712-2721.

Hunter, J.D. "Matplotlib: A 2D graphics environment." *Computing in science & engineering* 9(03) (2007): 90-95.

Javeri, I.Y., Toutiaee, M., Arpinar, I.B., Miller, J.A. and Miller, T.W. "Improving neural networks for time-series forecasting using data augmentation and AutoML." *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*. 2021, August.

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. and Kim, S.H. "Random forests for global and regional crop yield predictions." *PloS one* 11(6) (2016): e0156571.

Jeong, J.W., et al. "Negative effect of abnormal climate on the fruits productivity-focusing on the special weather report." *Korean Journal of Agricultural and Forest Meteorology* 20(4) (2018): 305-312.

Jiang, J. "A literature survey on domain adaptation of statistical classifiers." 2008. 9 May 2023. <http://www.mysmu.edu/faculty/jingjiang/papers/da_survey.pdf>.

Jin, X., Park, Y., Maddix, D., Wang, H. and Wang, Y. "Domain adaptation for time series forecasting via attention sharing." *In International Conference on Machine Learning (pp. 10280-10297). PMLR.* 2022, June.

Kastens, J.H., Kastens, T.L., Kastens, D.L., Price, K.P., Martinko, E.A. and Lee, R.Y. "Image masking for crop yield forecasting using AVHRR NDVI time series imagery." *Remote Sensing of Environment* 99.3 (2005): 341-356.

Király, Franz. *SKTime*. n.d. 15 6 2023. <https://github.com/sktime/sktime>.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B.E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J.B., Grout, J., Corlay, S. " Jupyter Notebooks-a publishing format for reproducible computational workflows." *Positioning and Power in Academic Publishing: Players, Agents and Agendas; Loizides, F., Schmidt, B., Eds, IOS Press*. Amsterdam, 2016.

Kouw, W.M. and Loog, M. "An introduction to domain adaptation and transfer learning." 2018.

Kulkarni, Krishnanand P., et al. "Harnessing the potential of forage legumes, alfalfa, soybean, and cowpea for sustainable agriculture and global food security." *Frontiers in plant science* 1314.9 (2018).

Li, C. and Lee, G.H. "From synthetic to real: Unsupervised domain adaptation for animal pose estimation." *IEEE/CVF conference on computer vision and pattern recognition*. 2021. 1482-1491.

Lomborg, B. *The Nobel Laureates' Guide to the Smartest Targets for the World: 2016-2030.* Copenhagen Consensus Center USA, 2015.

Lütkepohl, Helmut. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I. and Frey, B.,. "Adversarial autoencoders." *arXiv* (2015).

Makridakis, S., Spiliotis, E. and Assimakopoulos, V. "M5 accuracy competition: Results, findings, and conclusions." *International Journal of Forecasting* 38(4) (2022): 1346-1364.

Mark W Rosegrant, Eduardo Magalhaes, Rowena A Valmonte-Santos, Daniel Mason-D'Croz. *Returns to investment in reducing postharvest food losses and increasing agricultural productivity growth.* Cambridge University Press, 2018.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, L. "TensorFlow: Large-scale machine learning on heterogeneous systems." 2015. 15 6 2013. <tensorflow.org>.

Matouq, M., El-Hasan, T., Al-Bilbisi, H., Abdelhadi, M., Hindiyeh, M., Eslamian, S. and Duheisat, S. "The climate change implication on Jordan: A case study using GIS and Artificial Neural Networks for weather forecasting." *Journal of Taibah University for Science* 7(2) (2013): 44-55.

McKinney, Wes et al. "Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference." Austin, TX, 2010. 51-56.

Mishra, S., Saenko, K. and Saligrama, V. "Surprisingly simple semi-supervised domain adaptation with pretraining and consistency." preprint (2021).

Myerson, R.B. "Nash equilibrium and the history of economic theory." *Journal of Economic Literature* 37(3) (1999): 1067-1082.

National Agricultural Statistics Center. Crop production. 8 November 2018. 2023 9 5. <https://www.nass.usda.gov/Publications/Todays_Reports/reports/crop1118.pdf>.

Ohio State University. *OSU Crop Performance Trials*. n.d. 11 November 2022. <https://u.osu.edu/perf/archive/>.

Oliphant, Travis E. *A guide to NumPy, volume 1*. USA: Trelgol Publishing USA, 2006.

Panda, S.S., Ames, D.P. and Panigrahi, S. "Application of vegetation indices for agricultural crop yield prediction using neural network techniques." *Remote sensing* 2(3) (2010): 673-696.

Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H. and Kim, Y. "Data synthesis based on generative adversarial networks." *arXiv* preprint (2018).

Pavlyshenko, B.M. "Machine-learning models for sales time series forecasting." *Data* 4.1 (2019): 15.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J. "Scikit-learn: Machine learning in Python." *the Journal of machine Learning research* 12 (2011): 282.

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D. and Saenko, K. "Visda: The visual domain adaptation challenge." *arXiv preprint* arXiv:1710.06924 (2017).

—. "Visda: The visual domain adaptation challenge." *arXiv preprint* arXiv:1710.06924 (2017).

Poerner, N., Waltinger, U. and Schütze, H. "Inexpensive domain adaptation of pretrained language models: Case studies on biomedical NER and covid-19 QA." preprint (2020).

Scher, S. and Messori, G., 2019. "How global warming changes the difficulty of synoptic weather forecasting." *Geophysical Research Letters* 46(5) (2019): 2931-2939.

Schlenker, W., Lobell, D.B. "Robust negative impacts of climate change on African agriculture." *Environ. Res. Lett.* 5, 014010 (2010).

Seabold, Skipper and Perktold, Josef. "statsmodels: Econometric and statistical modeling with python." *9th Python in Science Conference*. 2010.

Sharma, S.K., Sharma, D.P. and Verma, J.K. "Study on Machine-Learning Algorithms in Crop Yield Predictions specific to Indian Agricultural Contexts." *2021 International Conference on Computational Performance Evaluation (ComPE)*. IEEE, 2021, December. 155-166.

Shen, K., Jones, R.M., Kumar, A., Xie, S.M., HaoChen, J.Z., Ma, T. and Liang, P. "Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation." *In International Conference on Machine Learning PML*. 2022, June. 19847-19878.

Sheshadri, A., Borrus, M., Yoder, M. and Robinson, T. "Midlatitude error growth in atmospheric GCMs: The role of eddy growth rate." *Geophysical Research Letters* 48(23) (2021): p.e2021GL096126.

South Dakota State University. *SDSU Extension Publications Archive*. n.d. 11 November 2022. <https://openprairie.sdstate.edu/extension_pubs/4/>.

Taieb SB, Hyndman RJ. "Recursive and direct multi-step forecasting: the best of both worlds." 2012.

Torrey, L. and Shavlik, J. "Transfer learning." *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (2010): 242-264.

Toutiaee, M., Li, X., Chaudhari, Y., Sivaraja, S., Venkataraj, A., Javeri, I., Ke, Y., Arpinar, I., Lazar, N. and Miller, J. "Improving COVID-19 Forecasting using eXogenous Variables." *arXiv preprint* arXiv:2107.10397 (2021).

United Nations. *The Sustainable Development Agenda 2015*. 2015. United Nations. 9 5 2023. <https://www.un.org/sustainabledevelopment/development-agenda/>.

University of Florida. *HiPerGator 3.0*. 2023. 16 May 2023. <https://www.rc.ufl.edu/get-started/hipergator/>.

University of Georgia. *UGA Variety Trials*. n.d. 29 November 2022. <https://georgiaforages.caes.uga.edu/content/dam/caes-subsite/forages/docs/species/alfalfa-variety-trials-2008-2010.pdf>.

University of Kentucky. *UK Forage Variety Trials*. n.d. 29 November 2022. <https://forages.ca.uky.edu/variety_trials>.

Van Der Walt, S., Colbert, S.C. and Varoquaux, G. "The NumPy array: a structure for efficient numerical computation." *Computing in science & engineering* 13(2) (2011): 22-30.

Vance, J., Rasheed, K., Missaoui, A., Maier, F., Adkins, C. and Whitmire, C. "Comparing Machine Learning Techniques for Alfalfa Biomass Yield Prediction." *arXiv* preprint (2022).

Vance, Jonathan. *PYCS*. 2022. 10 October 2022. <www.jonathanvance.online/pycs>.

Vance, Jonathan, et al. "Comparing Machine Learning Techniques for Alfalfa Biomass Yield Prediction." *arXiv preprint* arXiv:2210.11226 (2022).

—. "Data Synthesis for Alfalfa Biomass Yield Estimation." *AI* 4.1 (2022): 1-15.

Waskom, Michael et al. *Seaborn*. n.d. 11 March 2021. <https://doi.org/10.5281/zenodo.592845>.

Whitmire, C.D., Vance, J.M., Rasheed, H.K., Missaoui, A., Rasheed, K.M. and Maier, F.W. "Using machine learning and feature selection for alfalfa yield prediction." *AI* 2(1) (2021): 71-88.

Whitmire, Christopher Duncan. "Machine learning and feature selection for biomass yield prediction using weather and planting data, Master's Thesis." University of Georgia, 2019.

Xie, L., Lin, K., Wang, S., Wang, F. and Zhou, J. "Differentially private generative adversarial network." *arXiv* preprint (2018).

Xu, L., Skoularidou, M., Cuesta-Infante, A. and Veeramachaneni, K. "Modeling tabular data using conditional gan." *Advances in Neural Information Processing Systems* (2019): 32.

Xue J, Su B. "Significant remote sensing vegetation indices: A review of developments and applications." *Journal of sensors* (May 23, 2017).

Xue, J. and Su, B.,. " Significant remote sensing vegetation indices: A review of developments and applications." *Journal of sensors* (2017).

Yahya, B.M., Seker, D.Z. "Designing Weather Forecasting Model Using Computational Intelligence Tools." *Appl. Artif. Intell.* (2018).

You J, Li X, Low M, Lobell D, Ermon S. "Deep gaussian process for crop yield prediction based on remote sensing data." *Proceedings of the AAAI conference on artificial intelligence*. 2017, n.d.

Zolghadr-Asli, B., Enayati, M., Pourghasemi, H.R., Jahromi, M.N. and Tiefenbacher, J.P. "A linear/non-linear hybrid time-series model to investigate the depletion of inland water bodies." *Environment, Development and Sustainability* 23 (2021): 10727-1074.

# APPENDICES

## A. CODE AND DATA ACCESSIBILITY

- The source code for this project is available at https://github.com/chriswhitmire/alfalfa-yield-prediction and https://github.com/thejonathanvancetrance/Alfalfa
- The University of Georgia alfalfa yield data can be found here: https://georgiaforages.caes.uga.edu/species-and-varieties/cool-season/alfalfa.html
- The University of Kentucky alfalfa yield data can be found as progress reports on this page: http://dept.ca.uky.edu/agc/pub_prefix.asp?series=PR
- Note that the only data that was used from the University of Kentucky was the non-roundup ready alfalfa varieties that were first harvested in the year 2013 or later. The daily weather data for Kentucky, Pennsylvania, Mississippi, and Wisconsin was found on the National Oceanic and Atmospheric Administration website: https://www.ncdc.noaa.gov/crn/qcdatasets.html
- The daily weather data for Georgia was given to us by the Georgia Automated Environmental Monitoring Network.
- The day length was found from the United States Naval Observatory's website: https://aa.usno.navy.mil/data/docs/Dur_OneYear.php
- Variety Trial Reports for PA can be found at https://extension.psu.edu/forage-variety-trials-reports
- Variety Trial Reports for WI can be found at https://fyi.extension.wisc.edu/forage/category/trial-results/

- Variety Trial Reports for MS can be found at https://www.mafes.msstate.edu/variety-trials/includes/forage/about.asp#perennial

## B. HYPERPARAMETER GRID VALUES REGRESSION

The grid for the hyperparameters of each model is as follows:

Regression Tree:

- ‘criterion’: [’mae’],
- ‘max_depth’: [5,10,25,50,100]

Random forest:

- ‘n_estimators’: [5, 10, 25, 50, 100],
- ‘max_depth’: [5, 10, 15, 20],
- ‘criterion’: ["mae"]

K-nearest neighbors:

- ‘n_neighbors’: [2,5,10],
- ‘weights’: [’uniform’, ’distance’],
- ‘leaf_size’: [5, 10, 30, 50]

Support vector machine:

- ‘kernel’: [’linear’, ’poly’, ’rbf’, ’sigmoid’],
- ‘C’: [0.1, 1.0, 5.0, 10.0],
- ‘gamma’: ["scale", "auto"], ’degree’: [2,3,4,5]

Neural Network:

- ‘hidden_layer_sizes’:[(3), (5), (10), (3,3), (5,5), (10,10)],
- ‘solver’: [’sgd’, ’adam’],

- ‘learning_rate’ : [‘constant’, ‘invscaling’, ‘adaptive’], ‘learning_rate_init’: [0.1, 0.01, 0.001]

Bayesian ridge regression:

- ‘n_iter’:[100,300,500],
- ‘lambda_1’: [1.e-6, 1.e-4, 1.e-2, 1, 10],
- ‘lambda_1’: [1.e-6, 1.e-4, 1.e-2, 1, 10]

Linear Regression: no hyperparameters

## C. HYPERPARAMETER GRID VALUES CLASSIFICATION

The grid for the hyperparameters of each model is as follows:

Decision Tree

- ‘criterion’: [‘gini’];
- ‘max_depth’: [5, 10, 25, 50, 100]

Random forest

- ‘n_estimators’: [5, 10, 25, 50, 100];
- ‘max_depth’: [5, 10, 15, 20];
- ‘criterion’: [‘gini’]

K-nearest neighbors

- ‘n_neighbors’: [2, 5, 10];
- ‘weights’: [‘uniform’, ‘distance’];
- ‘leaf_size’: [5, 10, 30, 50]

Support vector classifier

- ‘kernel’: [‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’];
- ‘C’: [0.1, 1.0, 5.0, 10.0];
- ‘gamma’: [‘scale’, ‘auto’];
- ‘degree’: [2, 3, 4, 5]

Neural Network

- ‘hidden_layer_sizes’: [(3), (5), (10), (3,3), (5,5), (7,7)];
- ‘’solver’: [‘sgd’, ‘adam’];

• ‘learning_rate’: [‘constant’, ‘invscaling’, ‘adaptive’]; • ‘learning_rate_init’: [0.1, 0.01, 0.001]

Logistic Regression—default parameters; XGBoost—default parameters

## D. ADDITIONAL TIME PLOTS

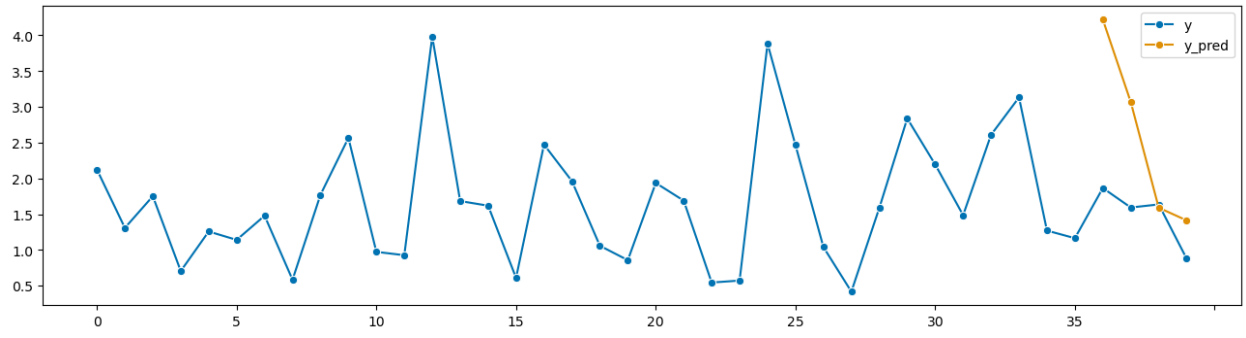


Figure D.1. Time plot, Table 5.5 ARIMA(30,1,15). Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 47.52%, R = 0.73, MAE = 1.10 tons/acre.

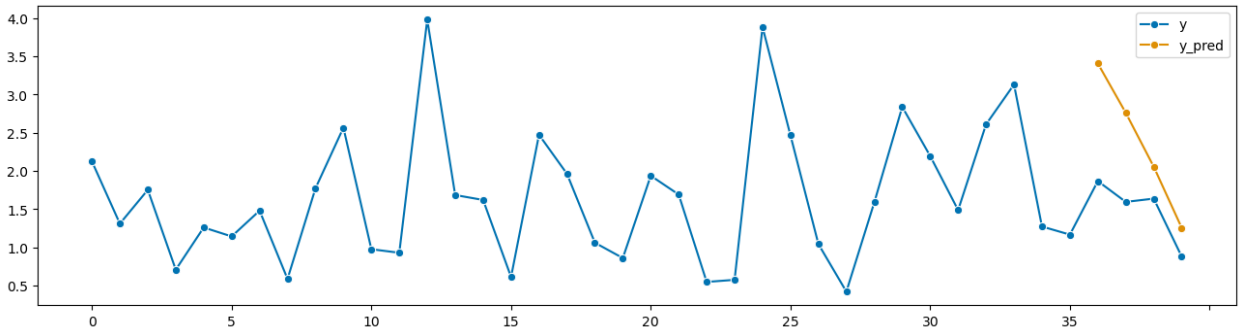


Figure D.2. Time plot, Table 5.5 SARIMA(1,0,0)(5,0,0,4). Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 34.67%, R = 0.83, MAE = 0.71 tons/acre.

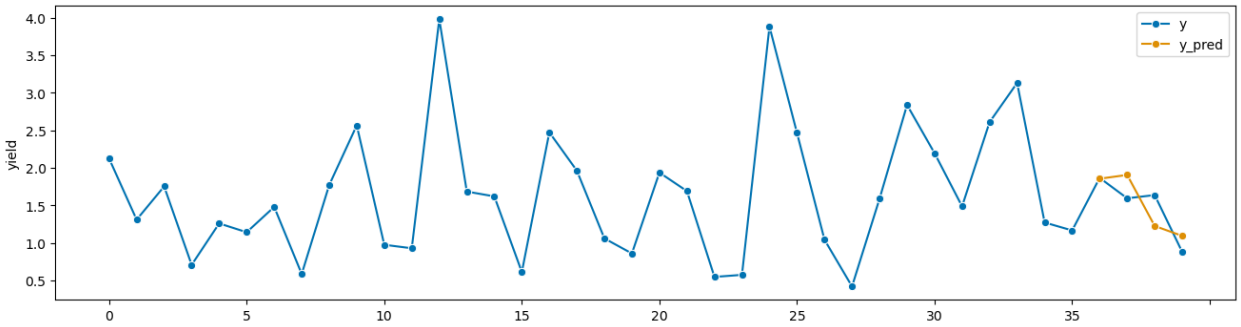


Figure D.3. Time plot, Table 5.5 RF. Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 17.24%, R = 0.71, MAE = 0.24 tons/acre, best model.

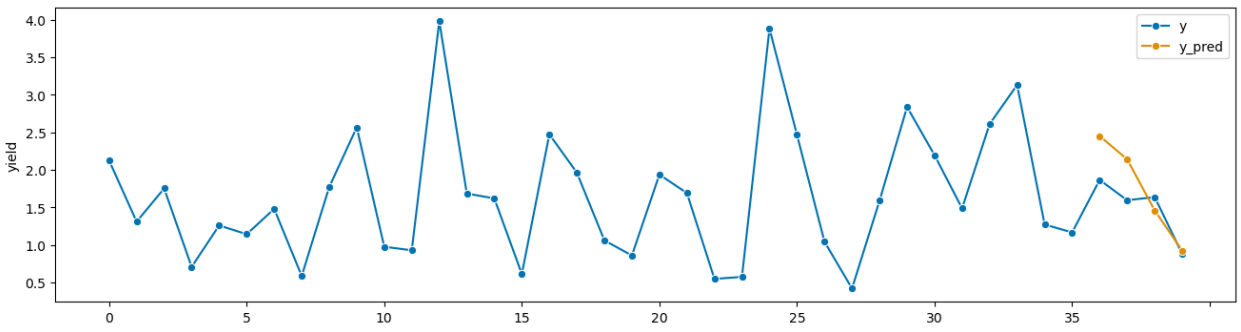

Figure D.4. Time plot, Table 5.5 KNN. Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 18.18%, R = 0.87, MAE = 0.34 tons/acre.

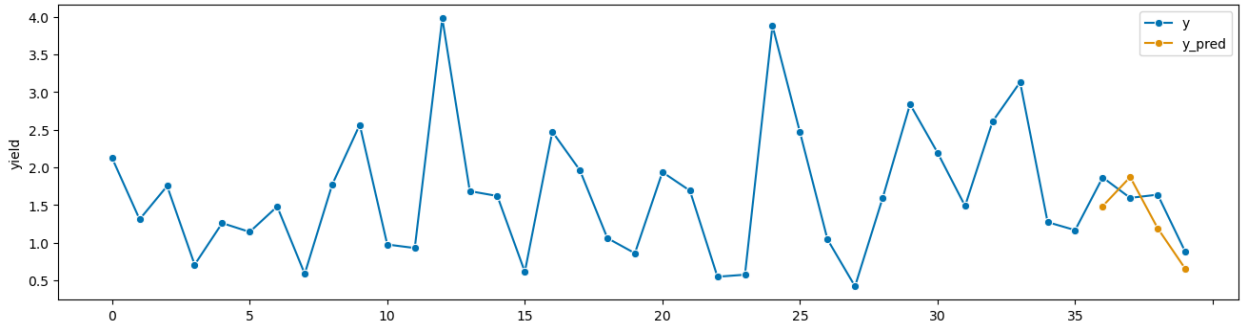


Figure D.5. Time plot, Table 5.5 DT. Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 24.97%, R = 0.77, MAE = 0.33 tons/acre.

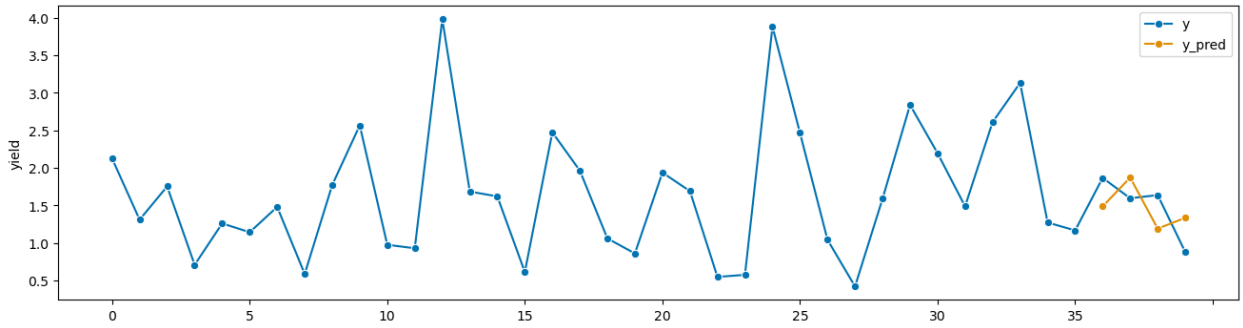


Figure D.6. Time plot, Table 5.5 XGB. Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 27.86%, R = 0.24, MAE = 0.39 tons/acre.

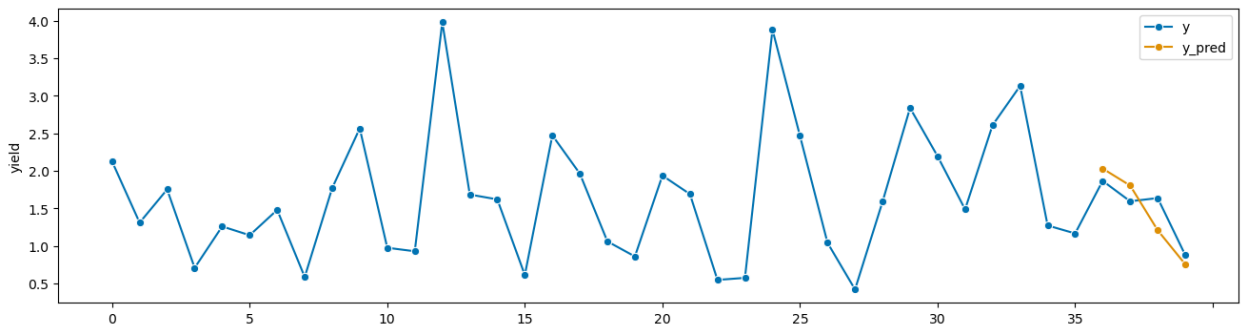

Figure D.7. Time plot, Table 5.5 MLP. Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 16.54%, R = 0.87, MAE = 0.23 tons/acre, best model.

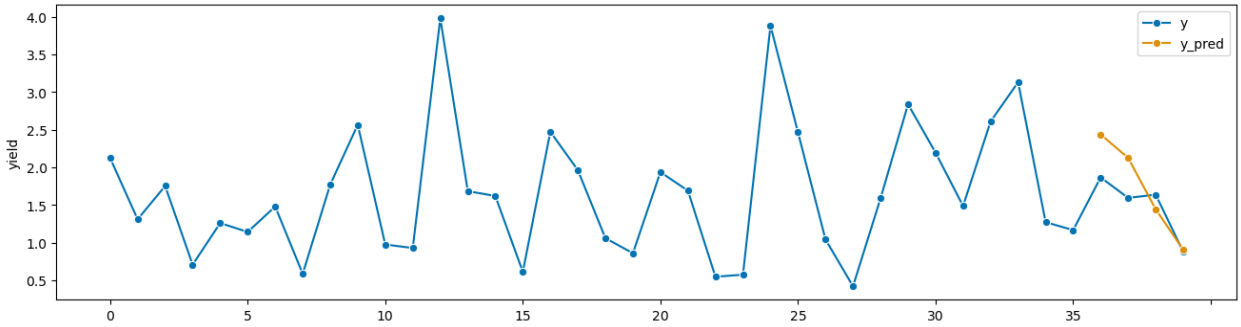


Figure D.8. Time plot, Table 5.5 LR. Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 17.68%, R = 0.87, MAE = 0.33 tons/acre.

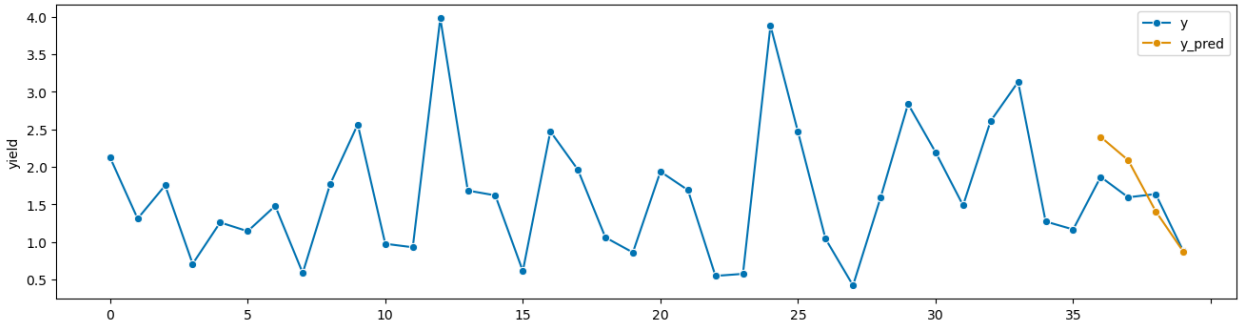


Figure D.9. Time plot, Table 5.5 BRR. Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 16.94%, R = 0.87, MAE = 0.32 tons/acre.

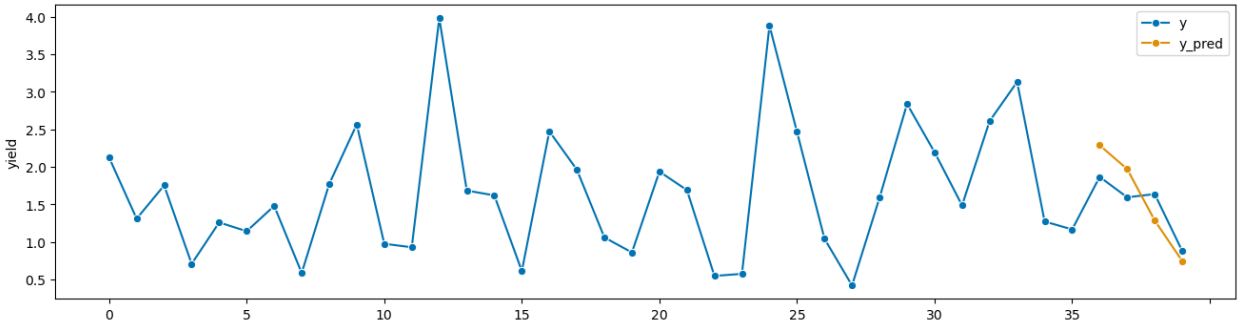


Figure D.10. Time plot, Table 5.5 SVR. Univariate time series, Beresford, SD, 1999 to 2010, FH = 1 (2011), sMAPE = 20.51%, R = 0.87, MAE = 0.32. tons/acre.

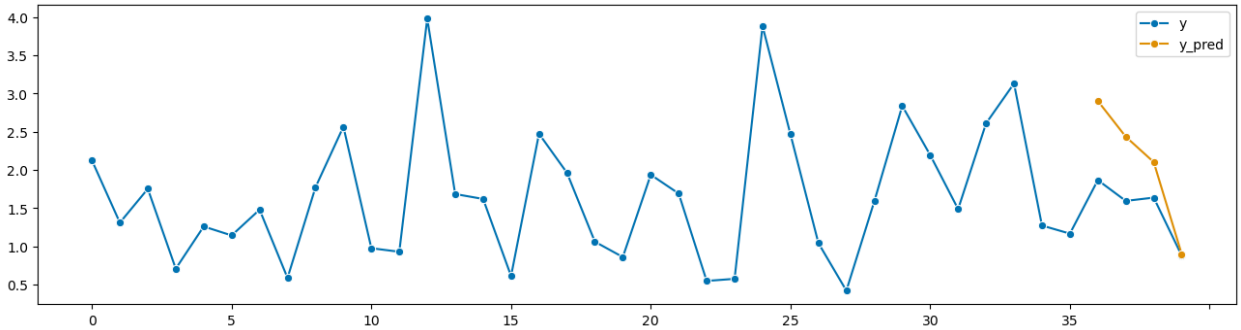


Figure D.11. Time plot, Table 5.6 SARIMAX(1,0,0)(5,0,1,4). Multivariate time series, Beresford, SD, 1999 to 2010, FH=1 (2011), sMAPE = 28.08%, R = 0.98, MAE = 0.59 tons/acre.

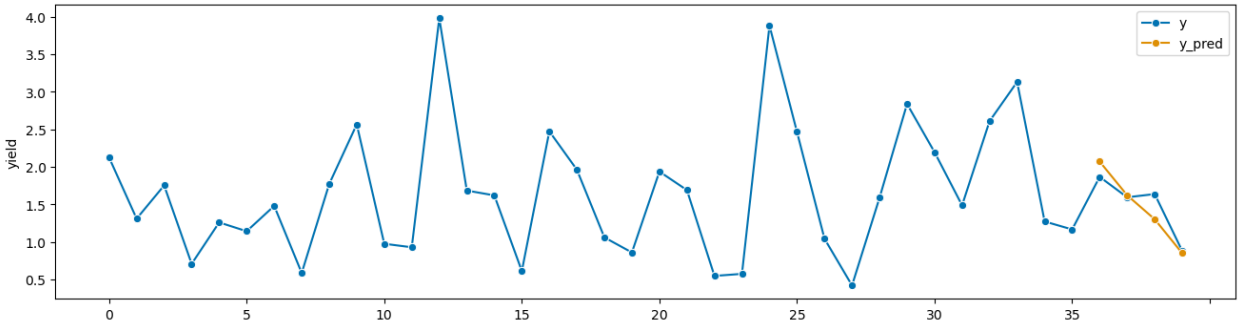


Figure D.12. Time plot, Table 5.6 RF. Multivariate time series, Beresford, SD, 1999 to 2010, FH=1 (2011), sMAPE = 9.61%, R = 0.90, MAE = 0.15 tons/acre, best model.

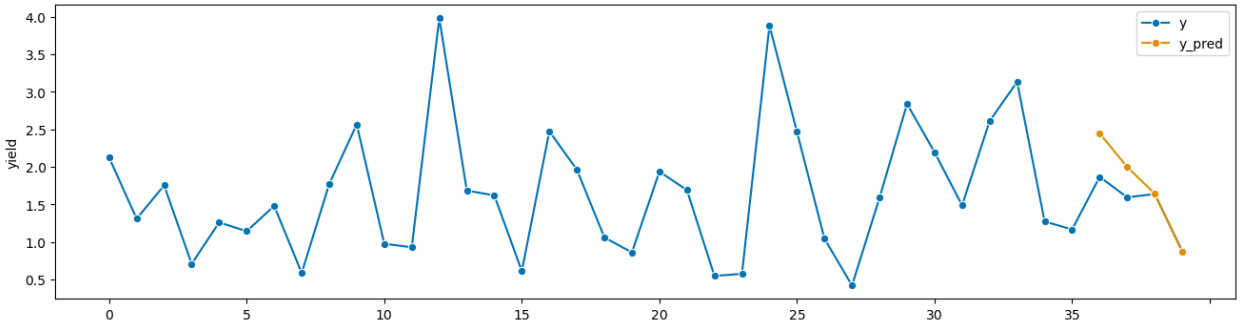


Figure D.13. Time plot, Table 5.6 KNN. Multivariate time series, Beresford, SD, 1999 to 2010, FH=1 (2011), sMAPE = 12.81%, R = 0.95, MAE = 0.25 tons/acre.

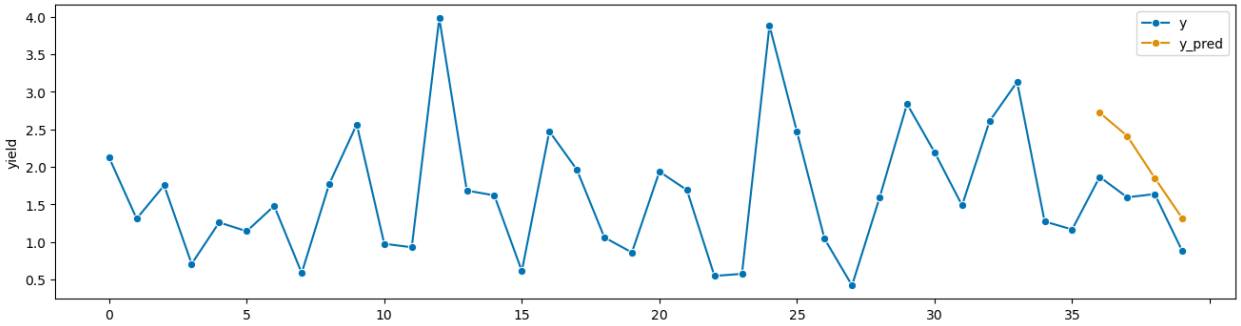


Figure D.14. Time plot, Table 5.6 DT. Multivariate time series, Beresford, SD, 1999 to 2010, FH=1 (2011), sMAPE = 32.60%, R = 0.89, MAE = 0.58 tons/acre, best model.

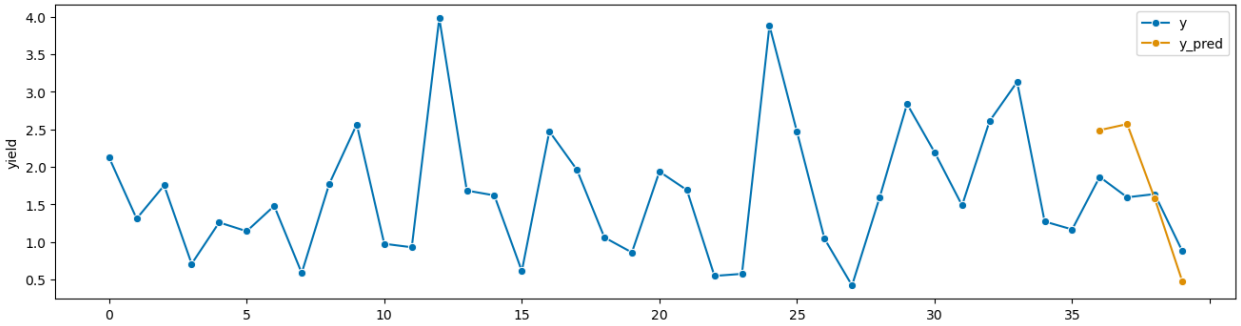


Figure D.15. Time plot, Table 5.6 XGB. Multivariate time series, Beresford, SD, 1999 to 2010, FH=1 (2011), sMAPE = 34.59%, R = 0.89, MAE = 0.51 tons/acre, best model.

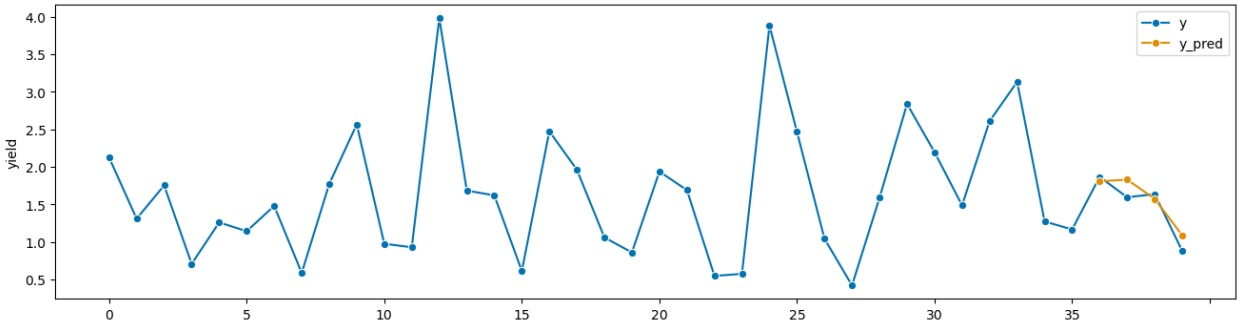


Figure D.16. Time plot, Table 5.6 MLP. Multivariate time series, Beresford, SD, 1999 to 2010, FH=1 (2011), sMAPE = 10.54%, R = 0.93, MAE = 0.14 tons/acre, best model.

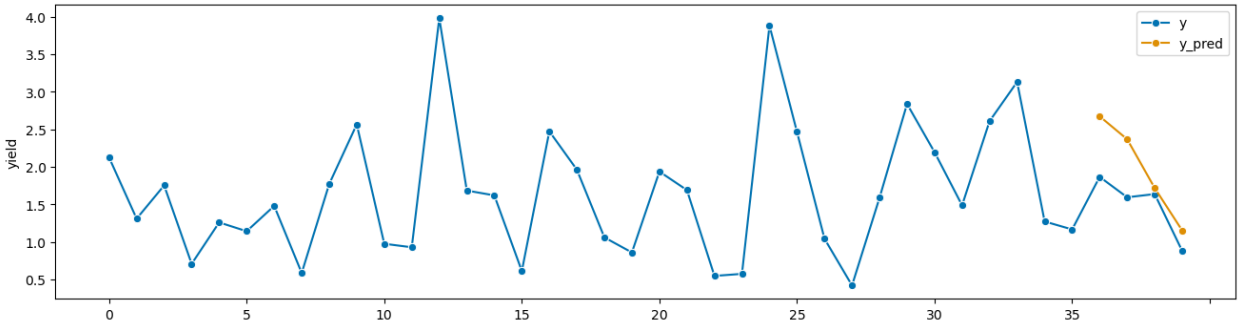


Figure D.17. Time plot, Table 5.6 LR. Multivariate time series, Beresford, SD, 1999 to 2010, FH=1 (2011), sMAPE = 26.57%, R = 0.88, MAE = 0.48 tons/acre.

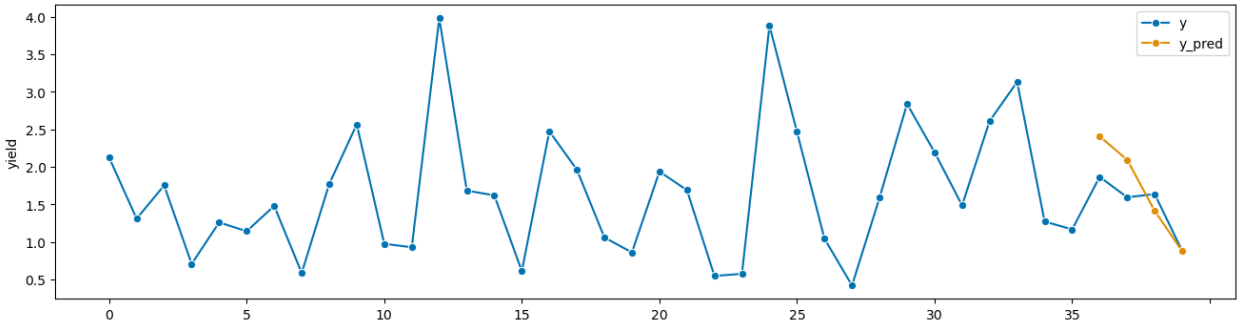


Figure D.18. Time plot, Table 5.6 BRR. Multivariate time series, Beresford, SD, 1999 to 2010, FH=1 (2011), sMAPE = 16.87%, R = 0.87, MAE = 0.32 tons/acre.

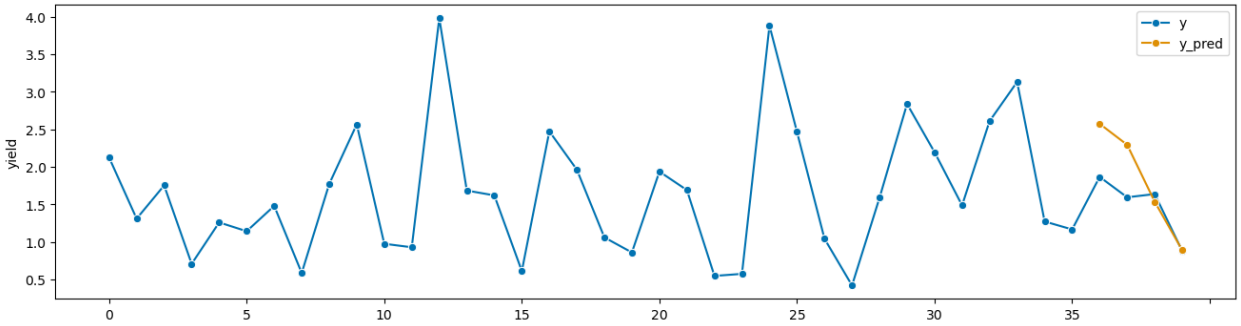


Figure D.19. Time plot, Table 5.6 SVR. Multivariate time series, Beresford, SD, 1999 to 2010, FH=1 (2011), sMAPE = 18.88%, R = 0.88, MAE = 0.38 tons/acre.

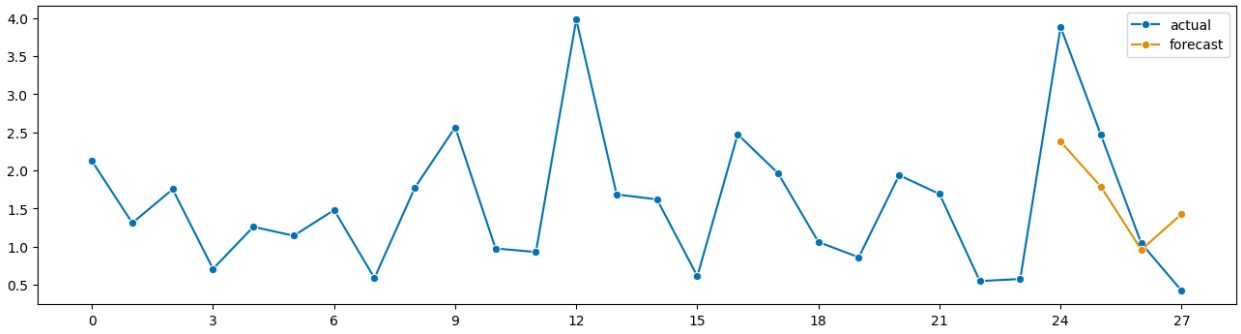


Figure D.20. Time plot, Table 5.7 ARIMA(10,0,0). Univariate sliding window validation results, target year 2008, Beresford, SD, trained on previous 6 years, sMAPE = 49.47%, R = 0.88, MAE = 0.82 tons/acre.

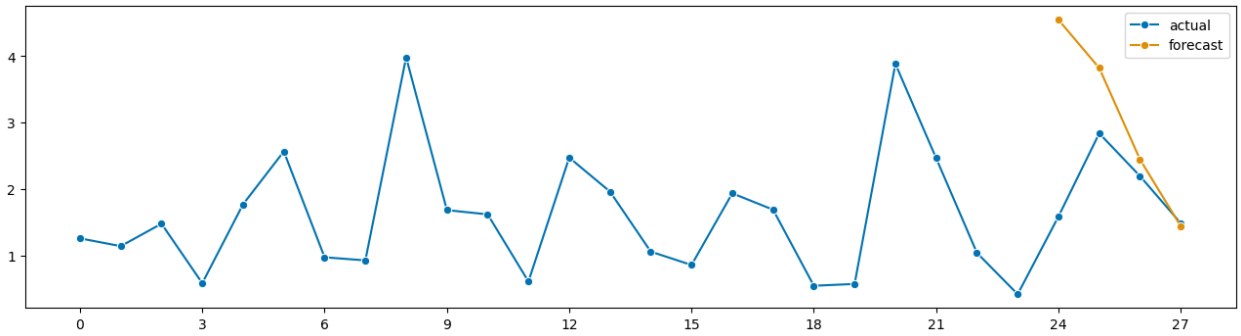


Figure D.21. Time plot, Table 5.7 ARIMA(24,1,10). Univariate sliding window validation results, target year 2009, Beresford, SD, trained on previous 6 years, sMAPE = 34.96%, R = 0.28, MAE = 1.06 tons/acre.

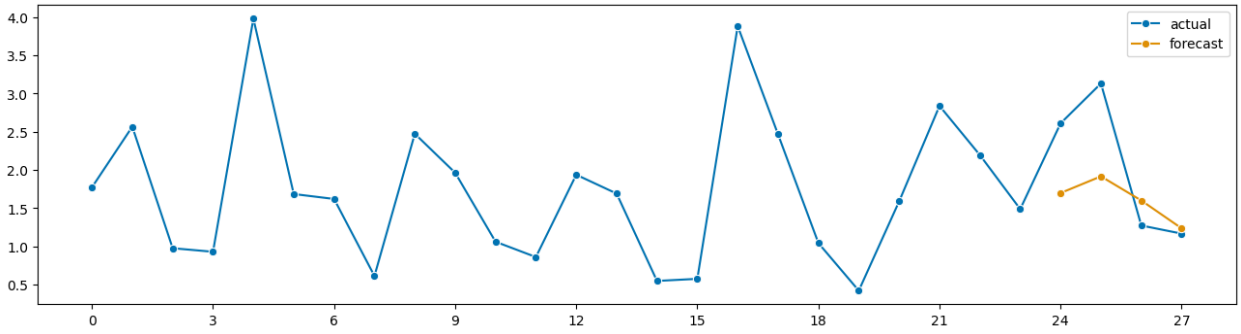


Figure D.22. Time plot, Table 5.7 ARIMA(8,0, 0). Univariate sliding window validation results, target year 2010, Beresford, SD, trained on previous 6 years, sMAPE = 29.80%, R = 0.86, MAE = .63 tons/acre.

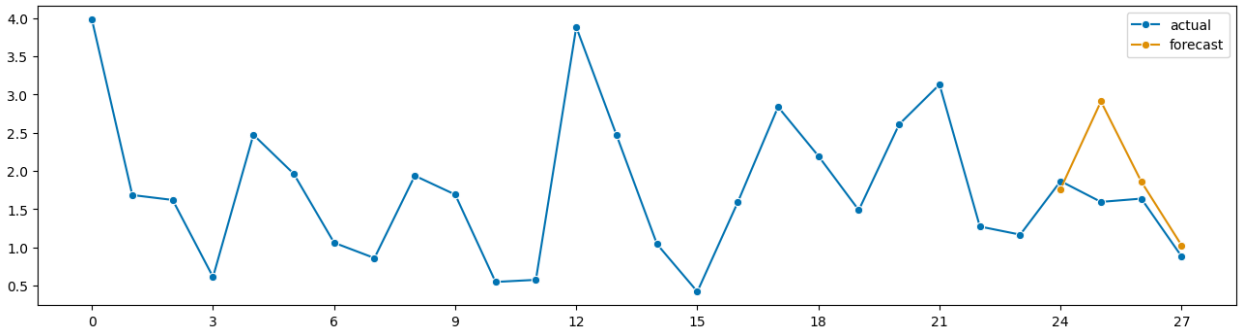


Figure D.23. Time plot, Table 5.7 ARIMA(9,0,0). Univariate sliding window validation results, target year 2011, Beresford, SD, trained on previous 6 years, sMAPE = 22.99%, R = 0.59, MAE = .45 tons/acre.

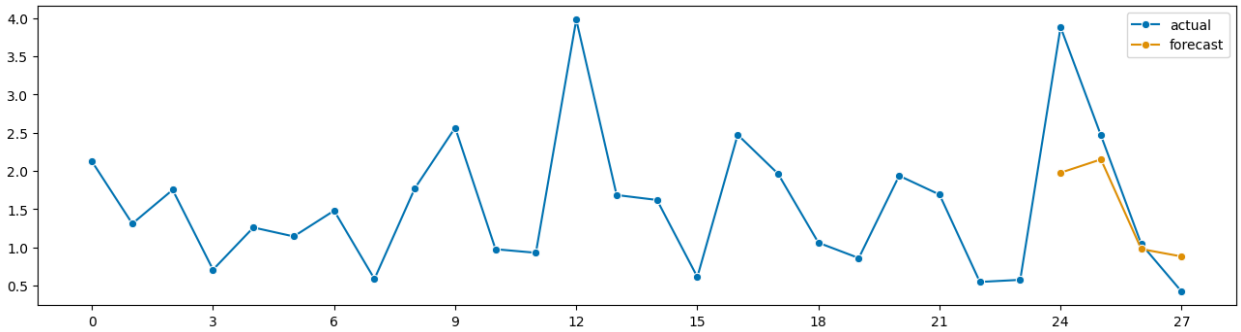


Figure D.24. Time plot, Table 5.7 SARIMA(4,0,0,4). Univariate sliding window validation results, target year 2008, Beresford, SD, trained on previous 6 years, sMAPE = 39.05%, R = 0.88, MAE = .69 tons/acre.

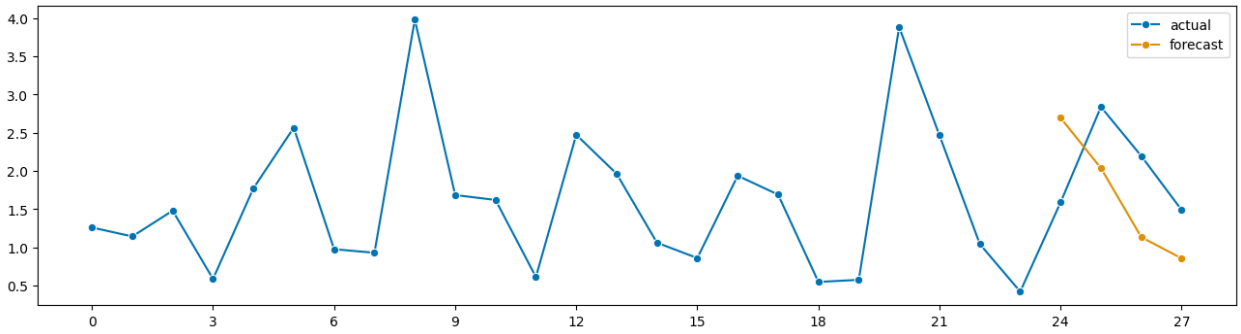


Figure D.25. Time plot, Table 5.7 SARIMA(3,0,0,4). Univariate sliding window validation results, target year 2009, Beresford, SD, trained on previous 6 years, sMAPE = 50.64%, R = 0.13, MAE = .90 tons/acre.

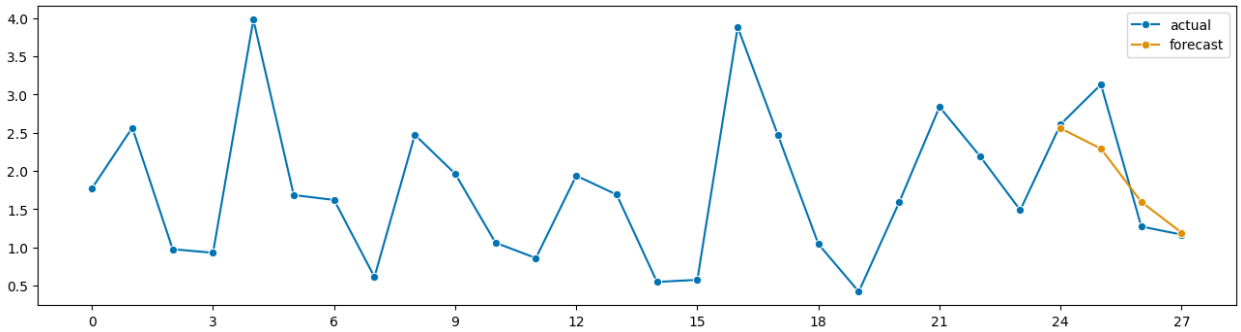


Figure D.26. Time plot, Table 5.7 SARIMA(2,0,0,4). Univariate sliding window validation results, target year 2010, Beresford, SD, trained on previous 6 years, sMAPE = 14.43%, R = 0.90, MAE = .31 tons/acre.

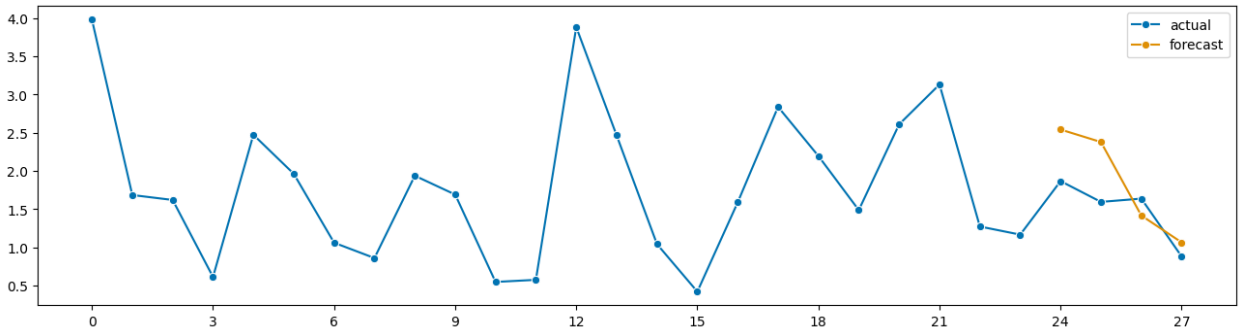


Figure D.26. Time plot, Table 5.7 SARIMA(2,0,1,4). Univariate sliding window validation results, target year 2011, Beresford, SD, trained on previous 6 years, sMAPE = 25.91%, R = 0.79, MAE = .47 tons/acre.

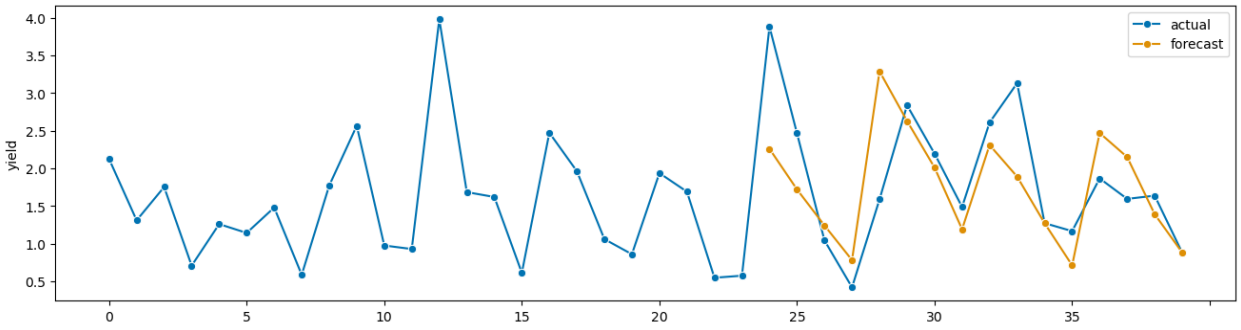


Figure D.27. Time plot, Table 5.7 KNN. Univariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, best models, sMAPE = 41.55%, 27.20%, 27.71%, 18.50%, R = 0.99, 0.23, 0.85, 0.86 for 2008-2011 respectively.

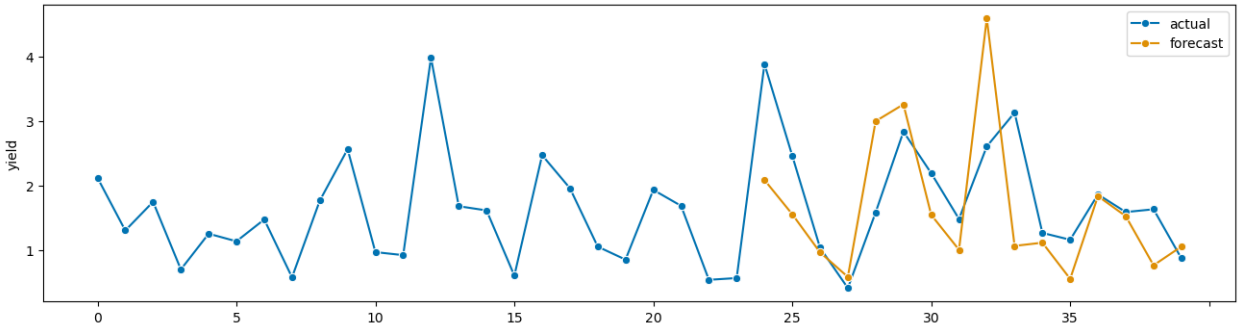


Figure D.28. Time plot, Table 5.7 DT. Univariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, best models, sMAPE = 36.62%, 36.92%, 59.04%, 24.24%, R = 0.99, 0.51, 0.44, 0.48 for 2008-2011 respectively.

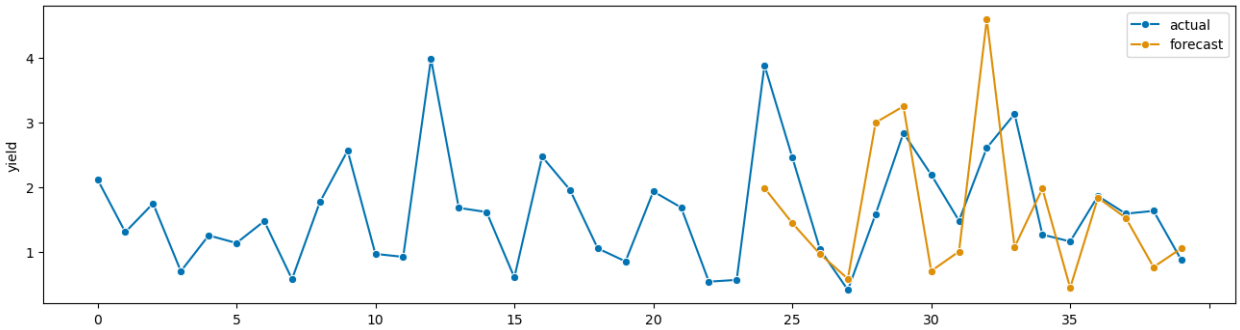


Figure D.29. Time plot, Table 5.7 XGB. Univariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, sMAPE = 39.38%, 54.01%, 71.04%, 24.14%, R = 0.99, 0.36, 0.34, 0.48 for 2008-2011 respectively.

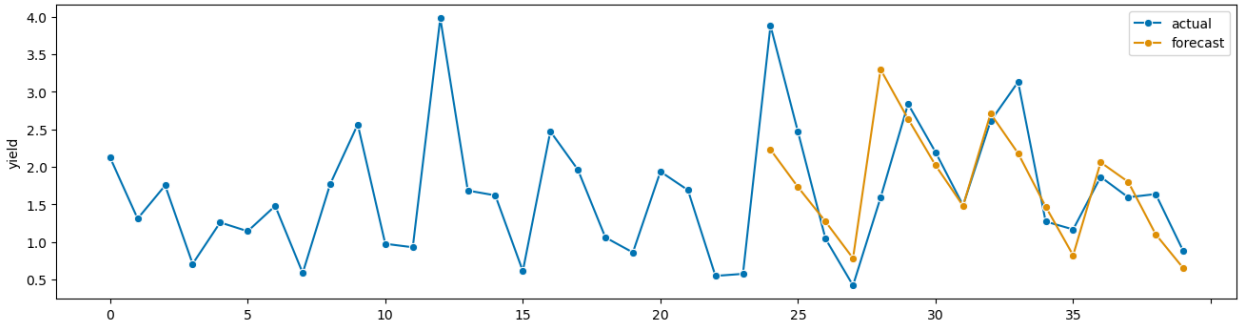


Figure D.30. Time plot, Table 5.7 MLP. Univariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, best models, sMAPE = 42.47%, 21.50%, 22.73%, 22.73%, R = 0.99, 0.16, 0.84, 0.85 for 2008-2011 respectively.

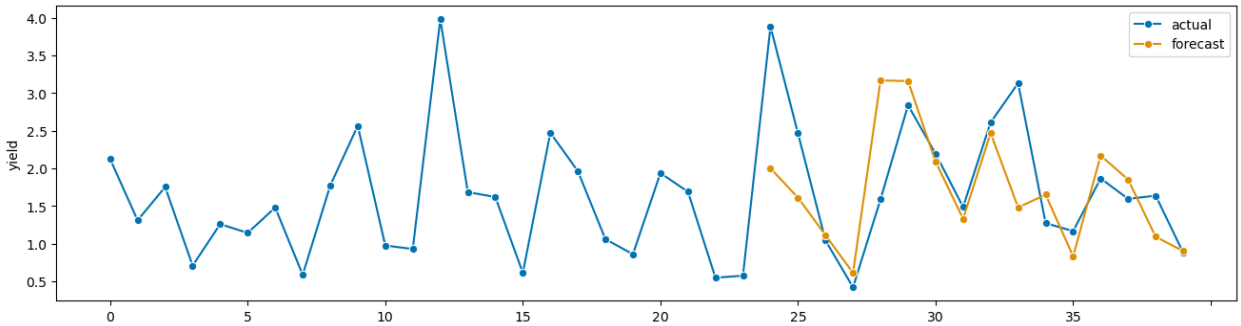


Figure D.31. Time plot, Table 5.7 RF. Univariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, best models, sMAPE = 37.36%, 22.49%, 34.26%, 18.11%, R = 0.98, 0.44, 0.50, 0.77 for 2008-2011 respectively.

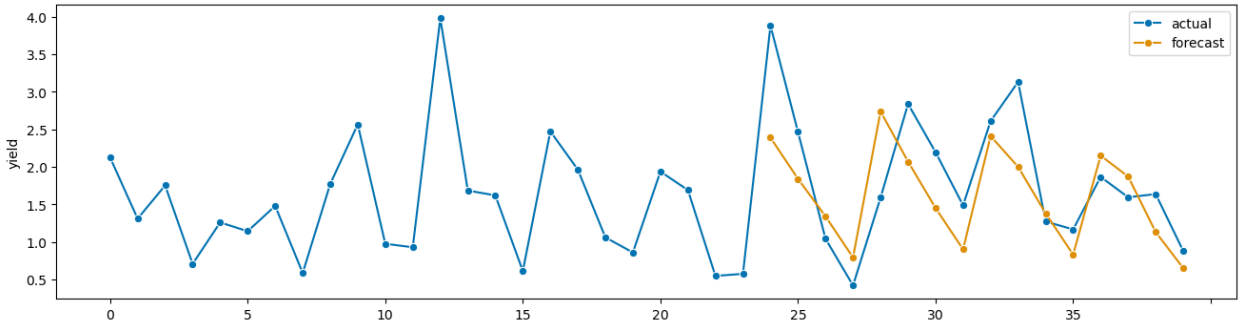


Figure D.32. Time plot, Table 5.7 LR. Univariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, sMAPE = 41.01%, 43.81%, 23.65%, 24.10%, R = 0.99, 0.16, 0.85, 0.85 for 2008-2011 respectively.

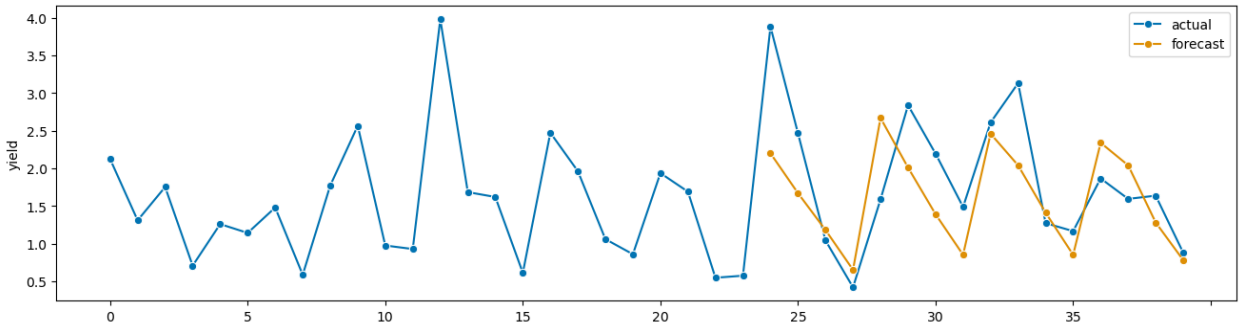


Figure D.33. Time plot, Table 5.7 BRR. Univariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, sMAPE = 37.46%, 46.20%, 22.56%, 20.83%, R = 0.99, 0.16, 0.85, 0.86 for 2008-2011 respectively.

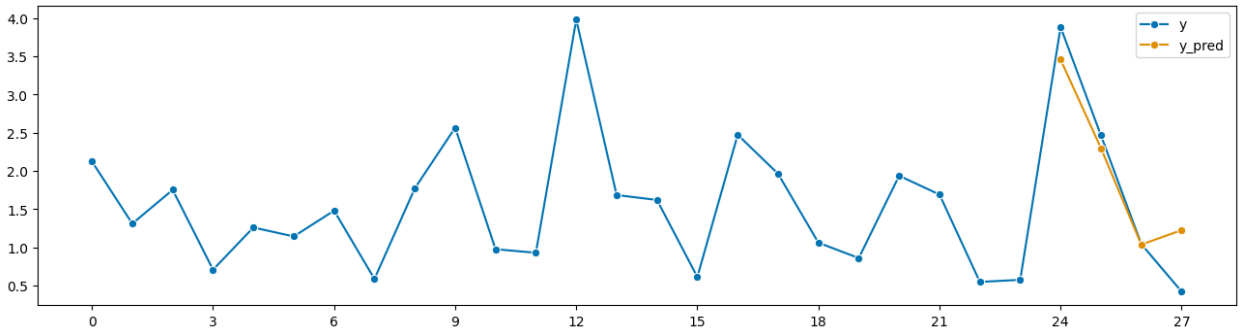


Figure D.34. Time plot, Table 5.8 SARIMAX(1,0,0)(5,0,0,4). Multivariate sliding window validation results, target year 2008, Beresford, SD, trained on previous 6 years, sMAPE = 29.33%, R = 0.97, MAE = 0.35 tons/acre.

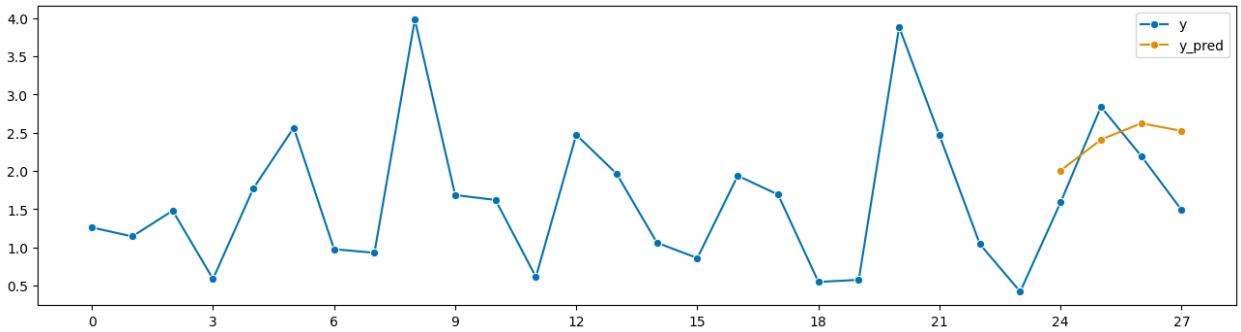


Figure D.35. Time plot, Table 5.8 SARIMAX(1,0,0)(2,0,1,4). Multivariate sliding window validation results, target year 2009, Beresford, SD, trained on previous 6 years, sMAPE = 27.29%, R = 0.30, MAE = 0.58 tons/acre.

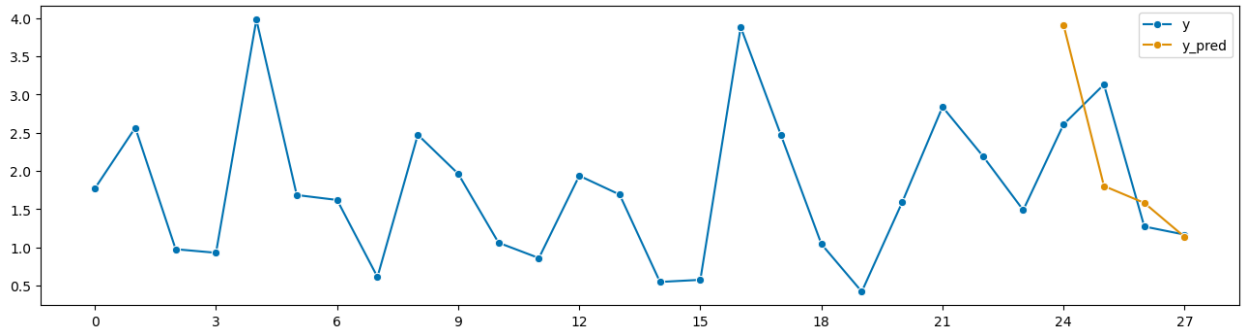


Figure D.36. Time plot, Table 5.8 SARIMAX(1,0,0)(5,1,2,4). Multivariate sliding window validation results, target year 2010, Beresford, SD, trained on previous 6 years, sMAPE = 29.54%, R = 0.54, MAE = 0.74 tons/acre.

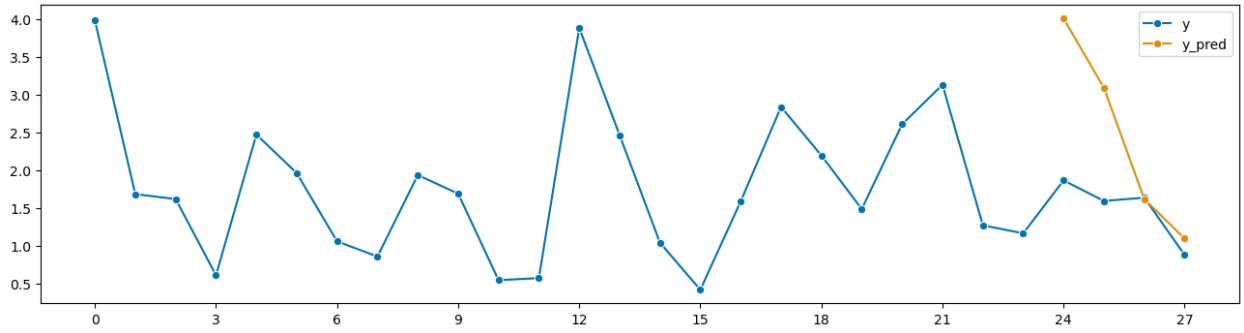


Figure D.37. Time plot, Table 5.8 SARIMAX(1,0,0)(3,0,0,4). Multivariate sliding window validation results, target year 2011, Beresford, SD, trained on previous 6 years, sMAPE = 40.00%, R = 0.79, MAE = 0.97 tons/acre.

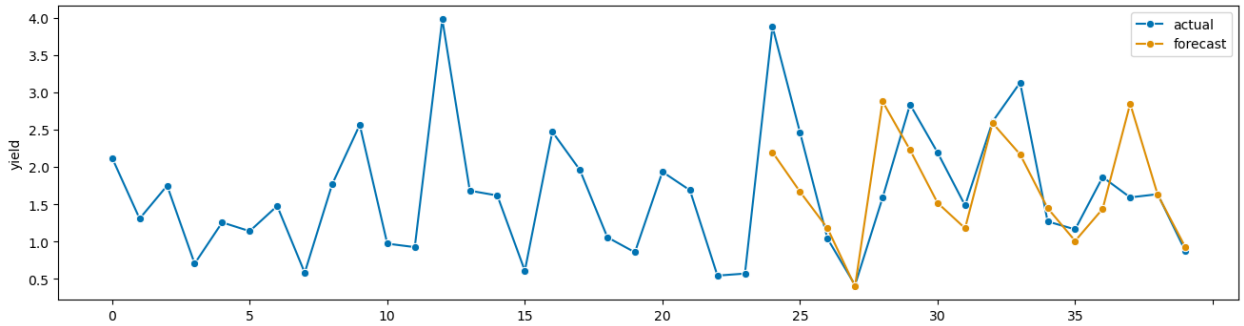

Figure D.38. Time plot, Table 5.8 KNN. Multivariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, sMAPE = 27.47%, 35.35%, 16.06%, 22.11%, R = 0.96, 0.11, 0.87, 0.46 for 2008-2011 respectively.

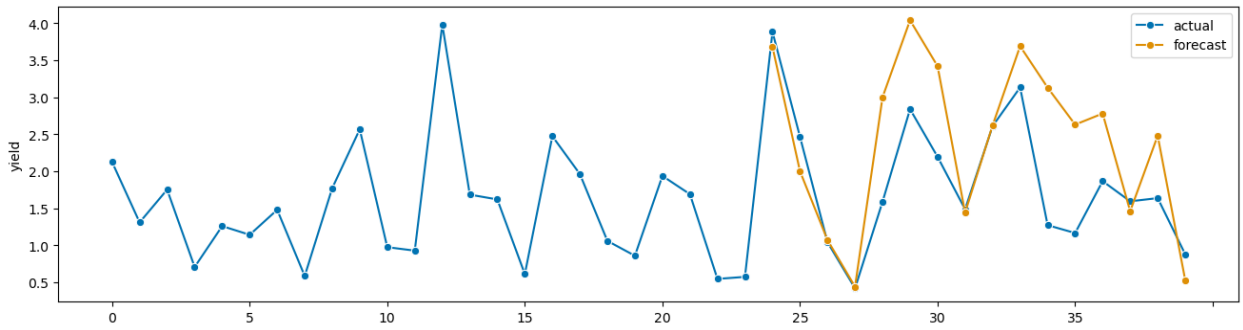


Figure D.39. Time plot, Table 5.8 DT. Multivariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, best models, sMAPE = 8.30%, 35.81%, 44.66%, 43.17%, R = 0.99, 0.85, 0.51, 0.99 for 2008-2011 respectively.

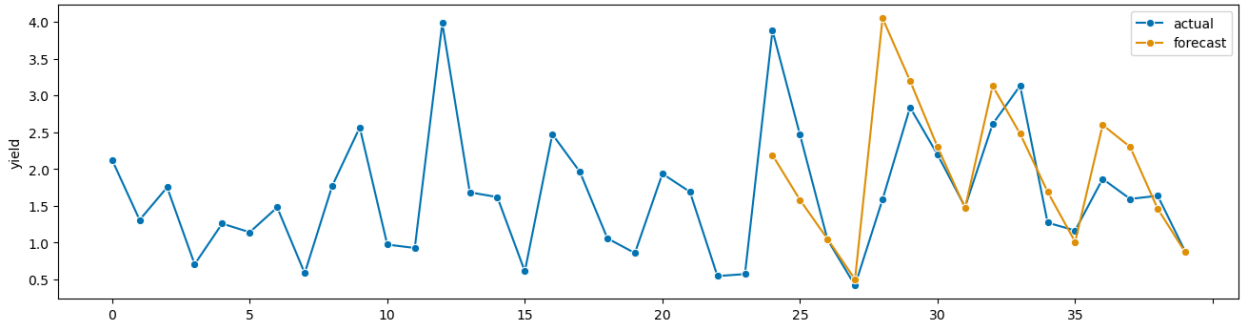


Figure D.40. Time plot, Table 5.8 SVR. Multivariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, best models, sMAPE = 29.61%, 26.24%, 20.91%, 20.34%, R = 0.99, 0.20, 0.84, 0.86 for 2008-2011 respectively.

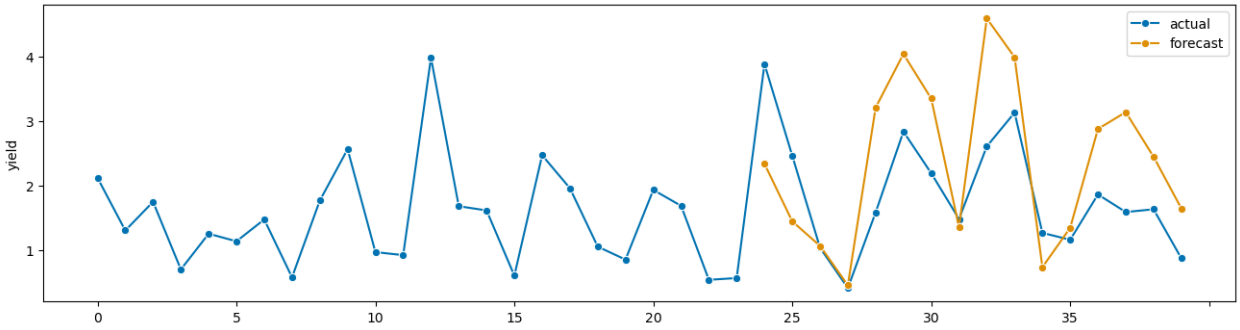


Figure D.41. Time plot, Table 5.8 XGB. Multivariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, best models, sMAPE = 28.24%, 38.27%, 36.61%, 52.26%, R = 0.98, 0.79, 0.92, 0.86 for 2008-2011 respectively.

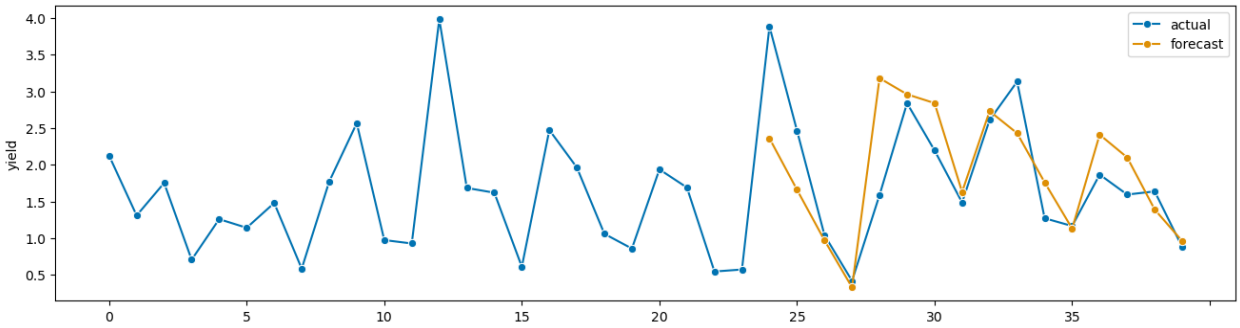


Figure D.42. Time plot, Table 5.8 MLP. Multivariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, best models, sMAPE = 29.98%, 26.47%, 16.27%, 19.60%, R = 0.99, 0.46, 0.87, 0.84 for 2008-2011 respectively.

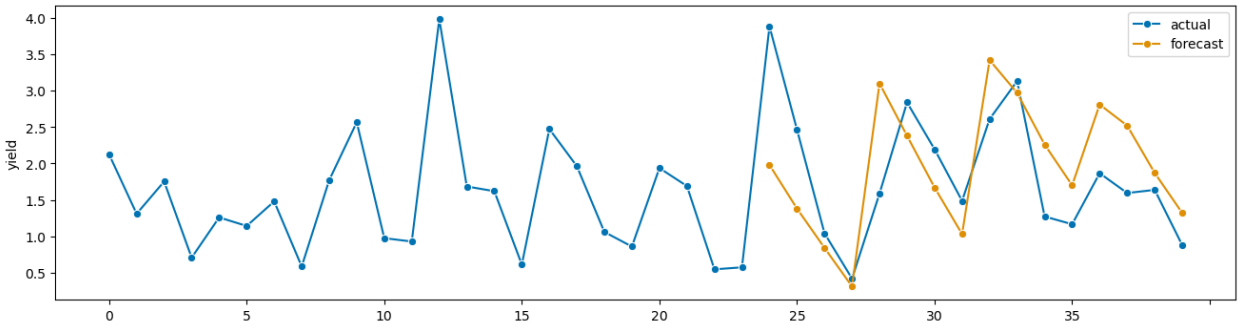

Figure D.43. Time plot, Table 5.8 LR. Multivariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, sMAPE = 43.79%, 36.33%, 31.36%, 34.68%, R = 0.99, 0.17, 0.86, 0.88 for 2008-2011 respectively.

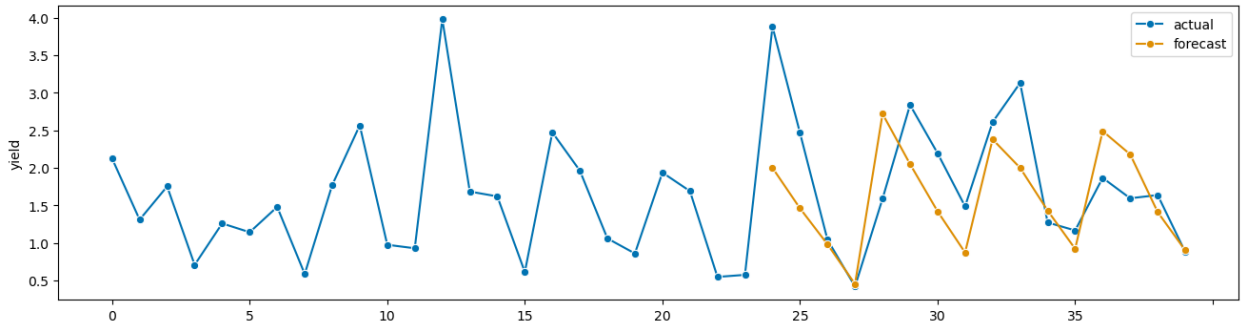


Figure D.43. Time plot, Table 5.8 BRR. Multivariate sliding window validation results, target years 2008 to 2011, Beresford, SD, trained on previous 6 years, sMAPE = 32.13%, 45.11%, 22.16%, 19.29%, R = 0.99, 0.16, 0.85, 0.86 for 2008-2011 respectively.

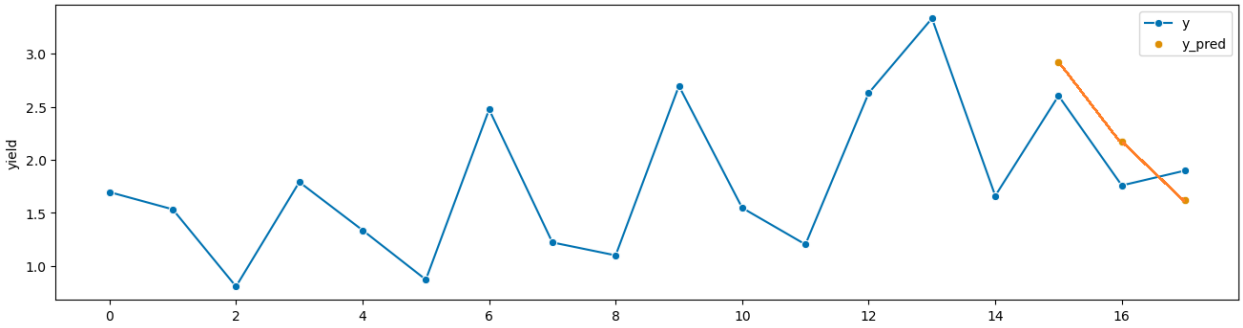


Figure D.44. Time plot, Table 5.9, ForDA w/XGB. Source: Watertown, SD; Target: Highmore, SD. TVAE synth, 20k samples, best run, sMAPE = 16.01%, R = 0.83, MAE = 0.33.

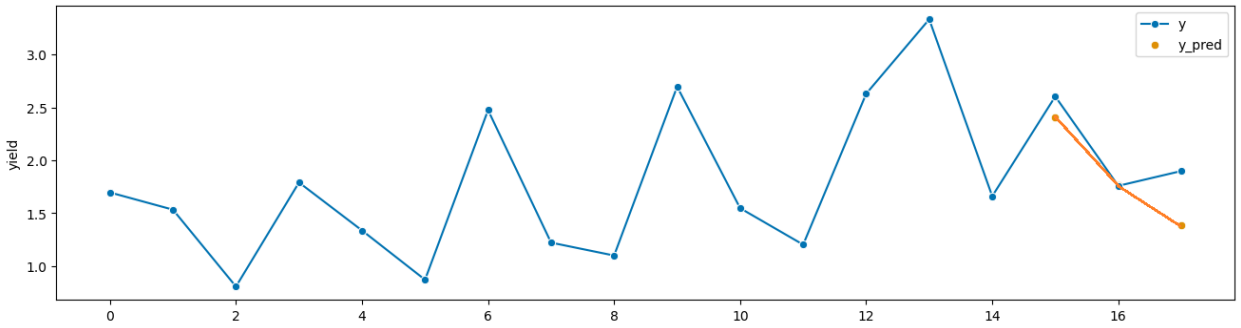


Figure D.45. Time plot, Table 5.9, ForDA w/XGB. Source: Watertown, SD; Target: Highmore, SD. CTG synth, 20k samples, best run, sMAPE = 13.18%, R = 0.86, MAE = 0.24.

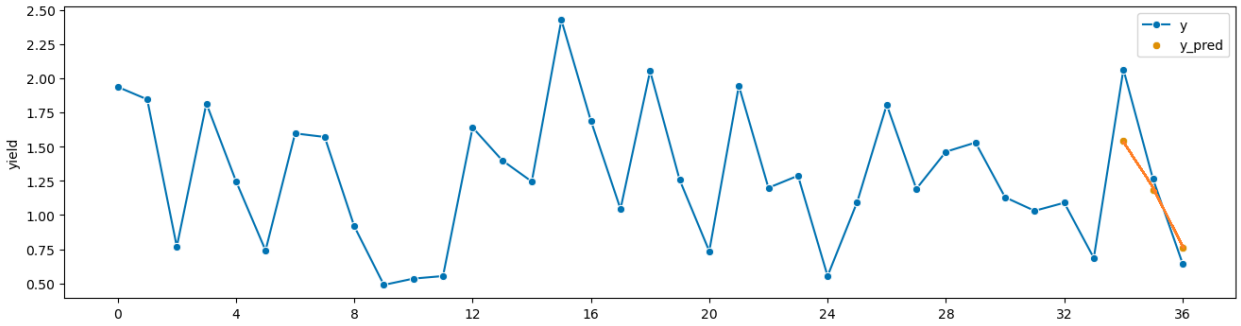


Figure D.46. Time plot, Table 5.10, ForDA w/XGB. Source: Highmore, SD; Target: Watertown, SD. TVAE synth, 20k samples, best run, sMAPE = 17.25%, R = 0.99, MAE = 0.24.

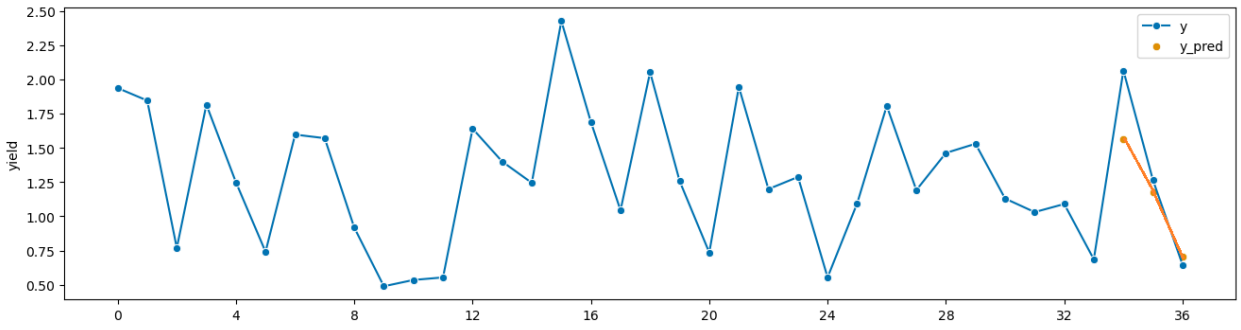


Figure D.47. Time plot, Table 5.10, ForDA w/XGB. Source: Highmore, SD; Target: Watertown, SD. CTG synth, 20k samples, best run, sMAPE = 14.48%, R = 0.99, MAE = 0.21.

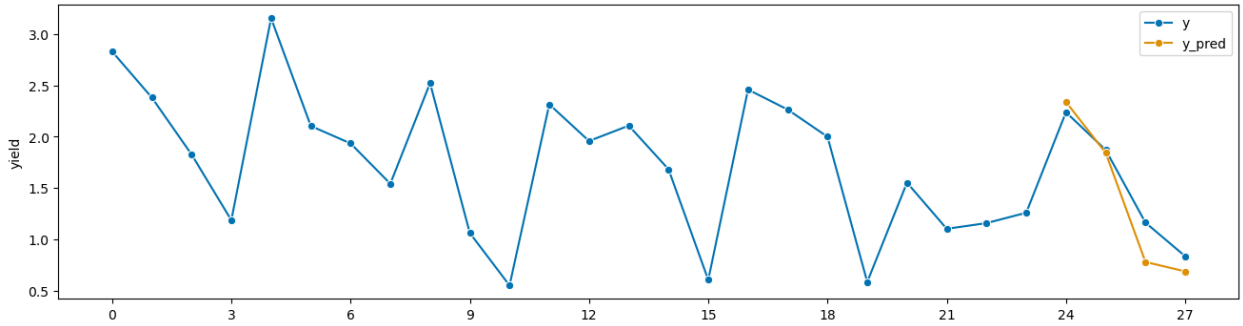


Figure D.48. Time plot, Table 5.12, ForDA w/XGB & SITS. OH round-robin results. Source: Wooster, OH; Target: North Baltimore, OH, best run, sMAPE = 16.13%, R = 0.99, MAE = 0.16.

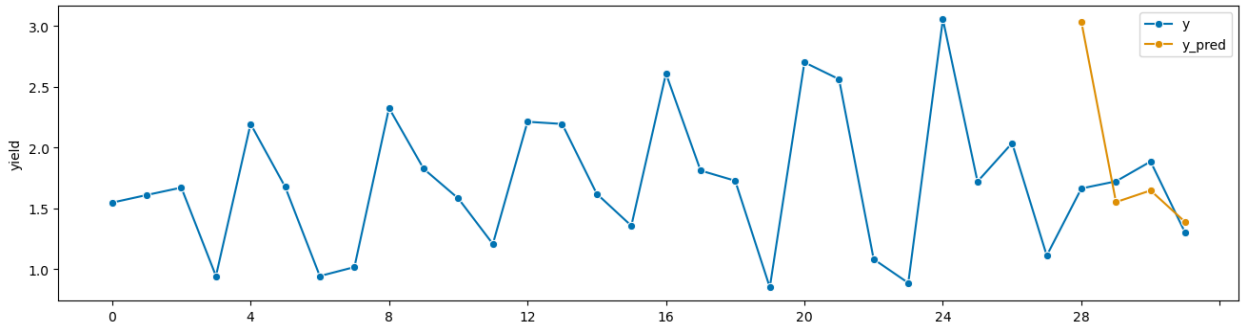


Figure D.49. Time plot, Table 5.12, ForDA w/XGB & SITS. OH round-robin results. Source: North Baltimore, OH; Target: Wooster, OH, best run, sMAPE = 22.20%, R = 0.20, MAE = 0.47.

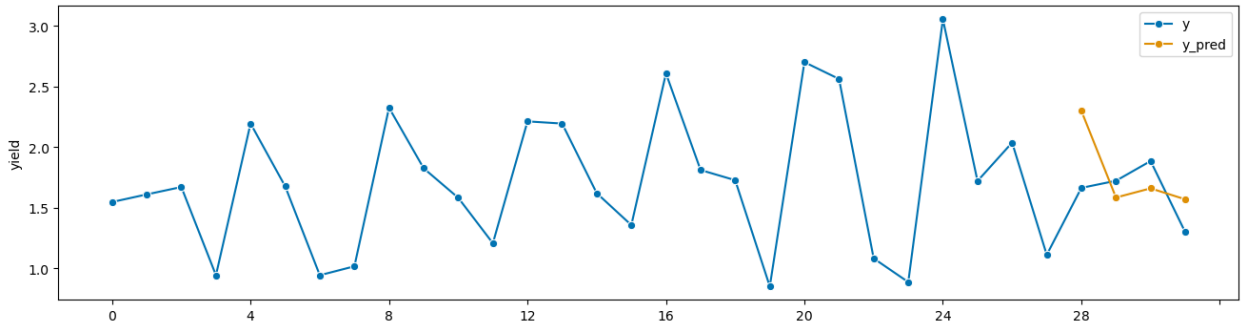

Figure D.50. Time plot, Table 5.12, ForDA w/XGB & SITS. OH round-robin results. Source: South Charleston, OH; Target: Wooster, OH, best run, sMAPE = 17.99%, R = 0.15, MAE = 0.32.

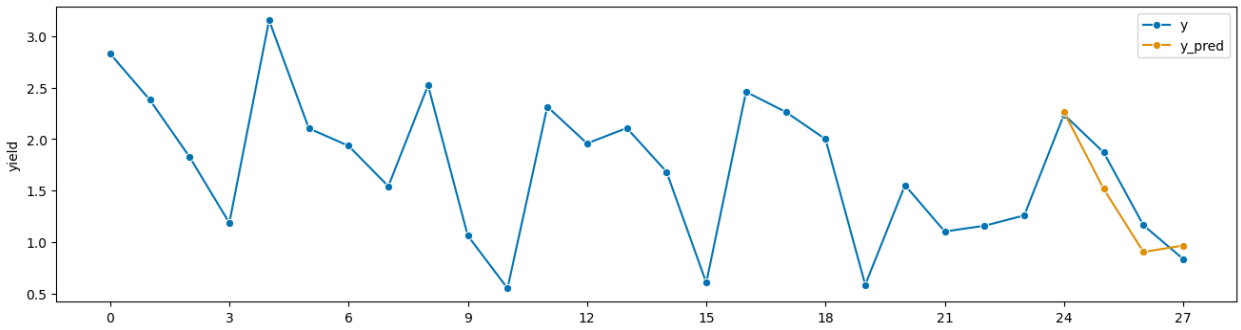


Figure D.51. Time plot, Table 5.12, ForDA w/XGB & SITS. OH round-robin results. Source: South Charleston, OH; Target: North Baltimore, OH, best run, sMAPE = 15.37%, R = 0.93, MAE = 0.19.

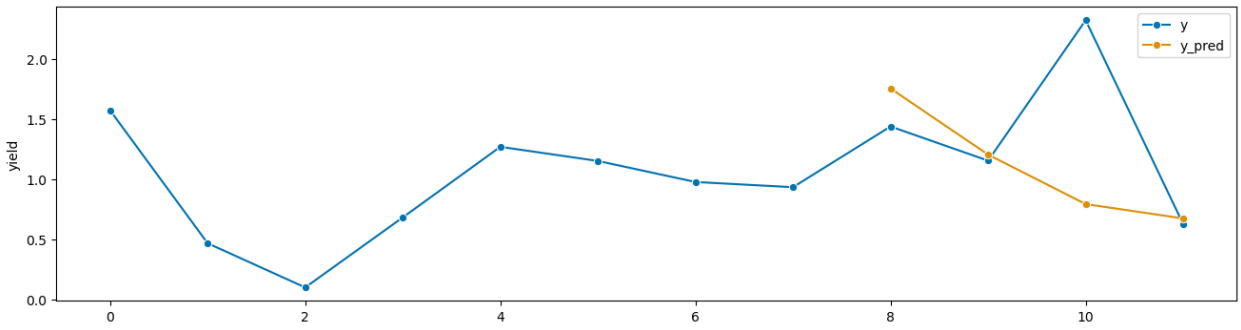


Figure D.52. Time plot, Table 5.13, ForDA, no synthesis. Source: MI, OH, SD, KY; Target: Athens, GA, sMAPE = 32.37%, R = 0.04, MAE = 0.49.

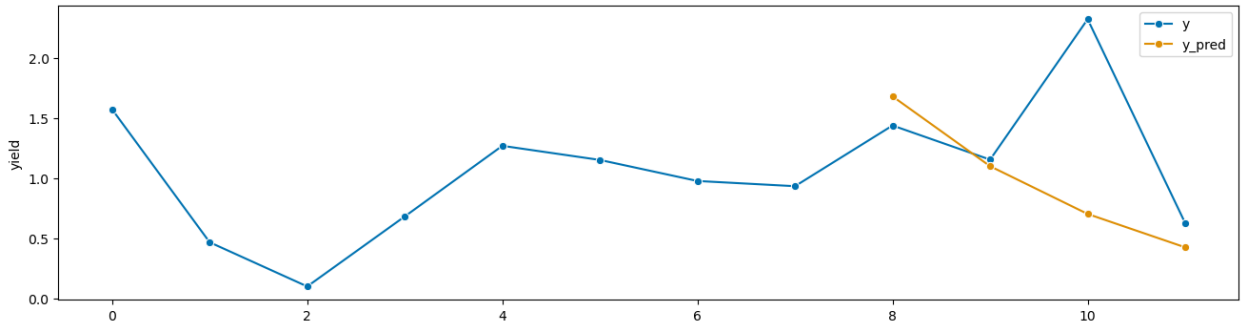


Figure D.53. Time plot, Table 5.13, ForDA w/SITS. Source: MI, OH, SD, KY; Target: Athens, GA, sMAPE = 41.43%, R = 0.15, MAE = 0.53.

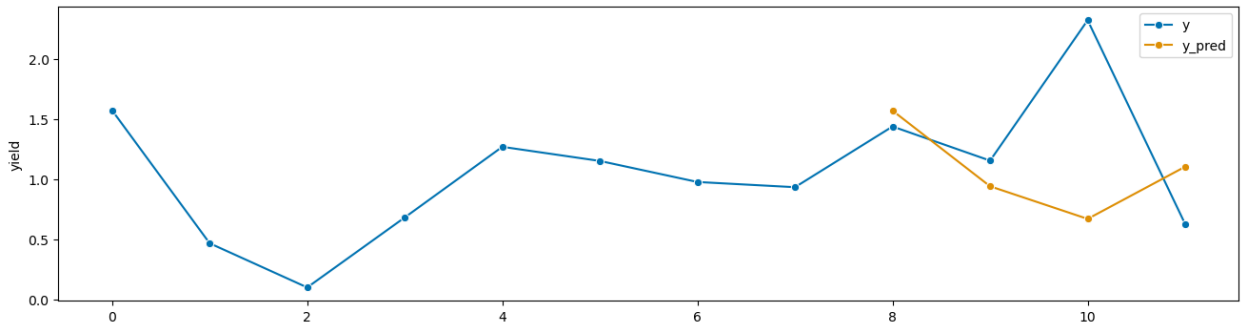


Figure D.54. Time plot, Table 5.13, ForDA, no synthesis. Source: East Lansing, MI; Target: Athens, GA, sMAPE = 48.70%, R = -0.10, MAE = 0.67.

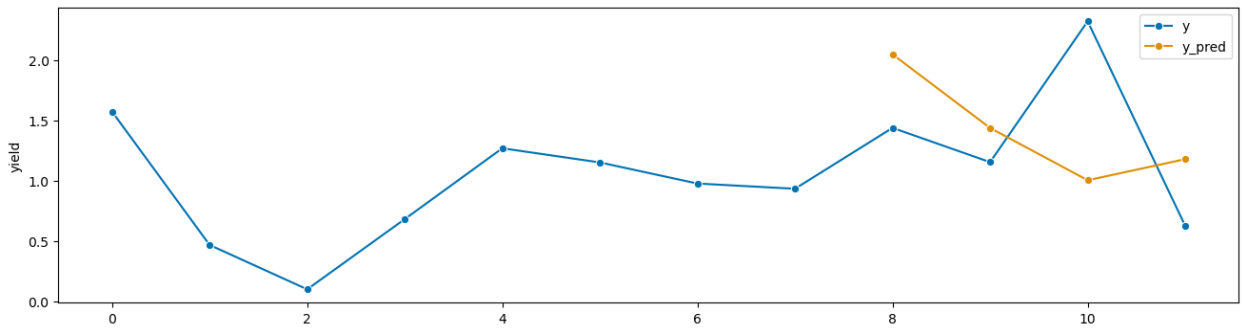

Figure D.55. Time plot, Table 5.13, ForDA, w/SITS. Source: East Lansing, MI; Target: Athens, GA, sMAPE = 49.32%, R = -0.18, MAE = 0.69.

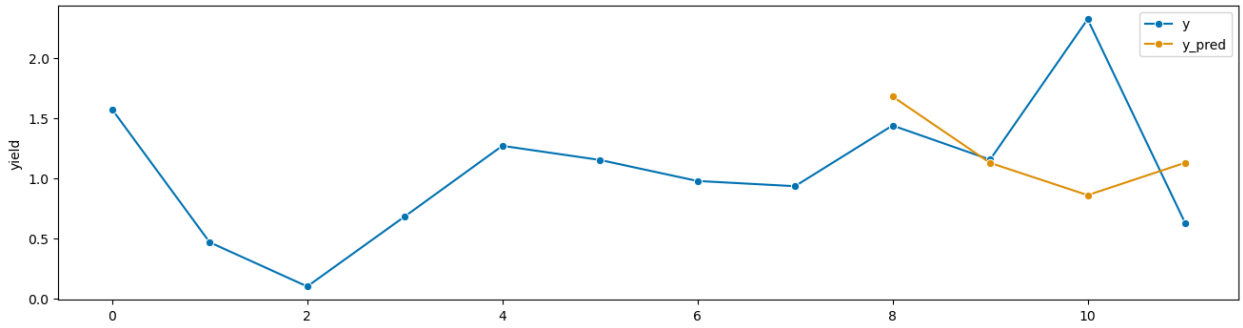


Figure D.56. Time plot, Table 5.13, ForDA, no synthesis. Source: Lexington, KY; Target: Athens, GA, sMAPE = 41.76%, R = -0.30, MAE = 0.56.

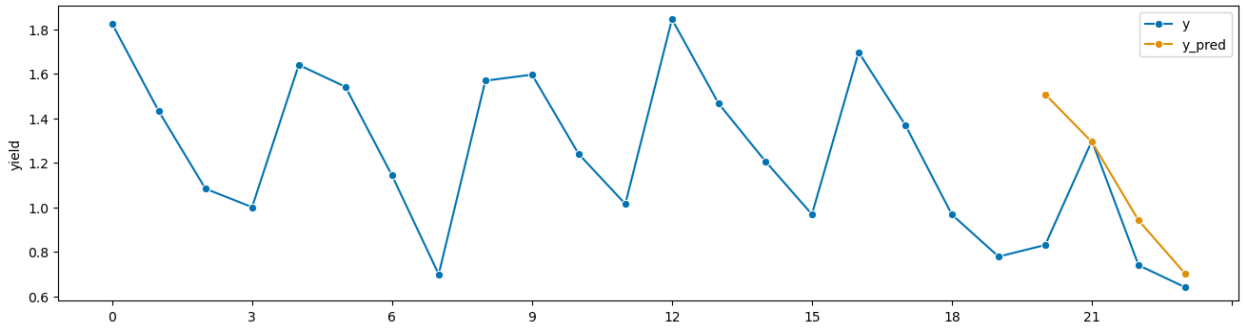


Figure D.57. Time plot, Table 5.14, ForDA w/SITS. Source: MI, OH, SD; Target: Lexington, KY, best model, sMAPE = 22.83%, R = 0.57, MAE = 0.24.

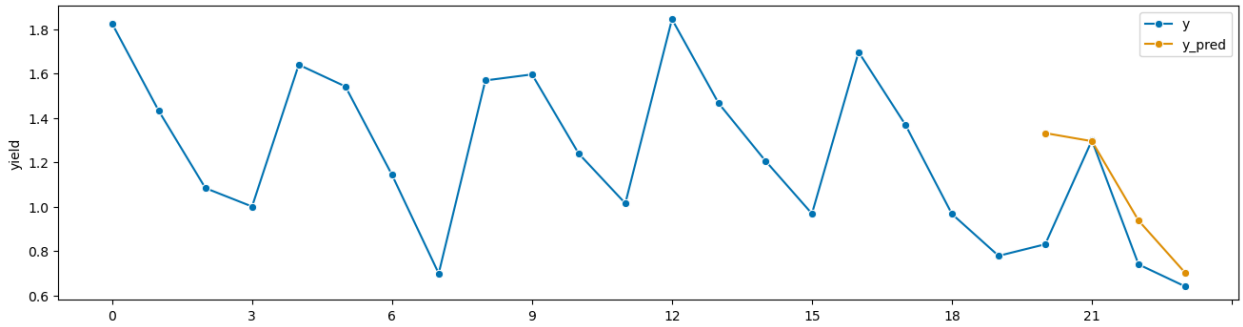

Figure D.58. Time plot, Table 5.14, ForDA, no synthesis. Source: MI, OH, SD; Target: Lexington, KY, sMAPE = 19.82%, R = 0.71, MAE = 0.19.

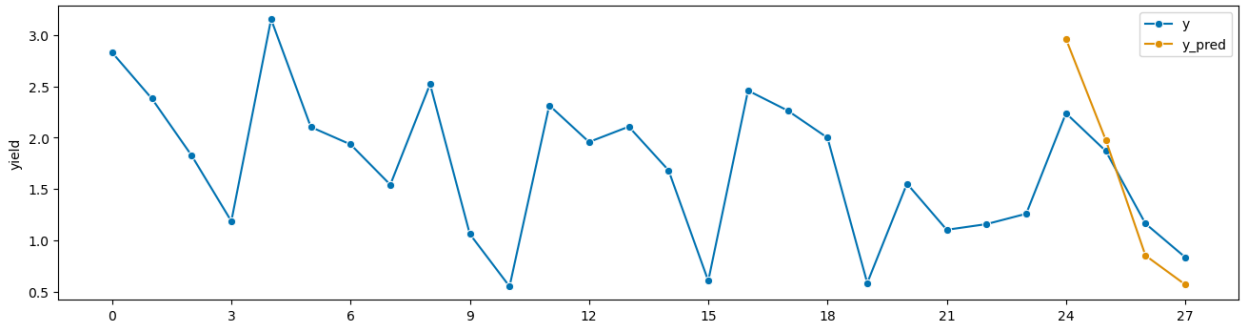


Figure D.59. Time plot, Table 5.14, ForDA, no synthesis. Source: MI, SD, KY; Target: North Baltimore, OH, sMAPE = 25.47%, R = 0.99, MAE = 0.35.

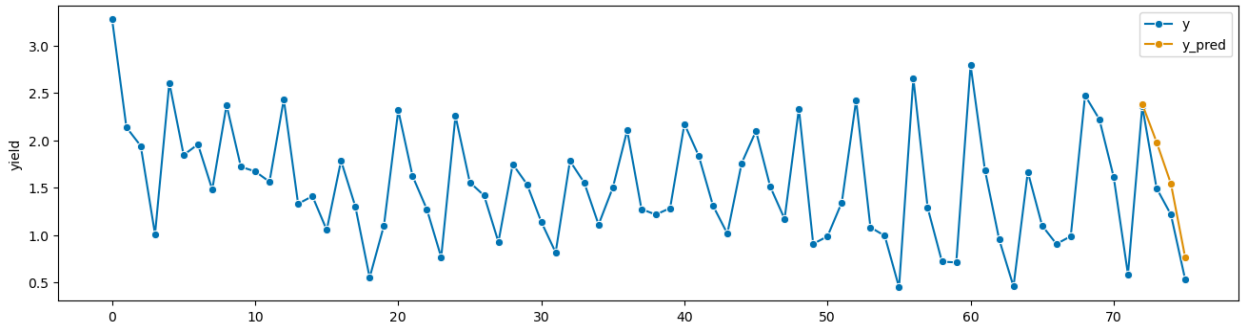


Figure D.60. Time plot, Table 5.14, ForDA w/SITS. Source: OH, SD, KY; Target: East Lansing, MI, best model, sMAPE = 22.15%, R = 0.97, MAE = 0.27.

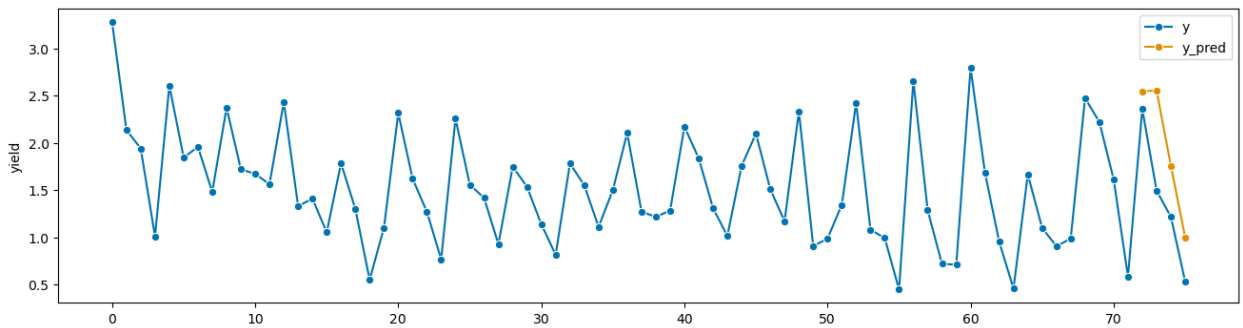

Figure D.61. Time plot, Table 5.14, ForDA, no synthesis. Source: OH, SD, KY; Target: East Lansing, MI, sMAPE = 39.39%, R = 0.88, MAE = 0.56.

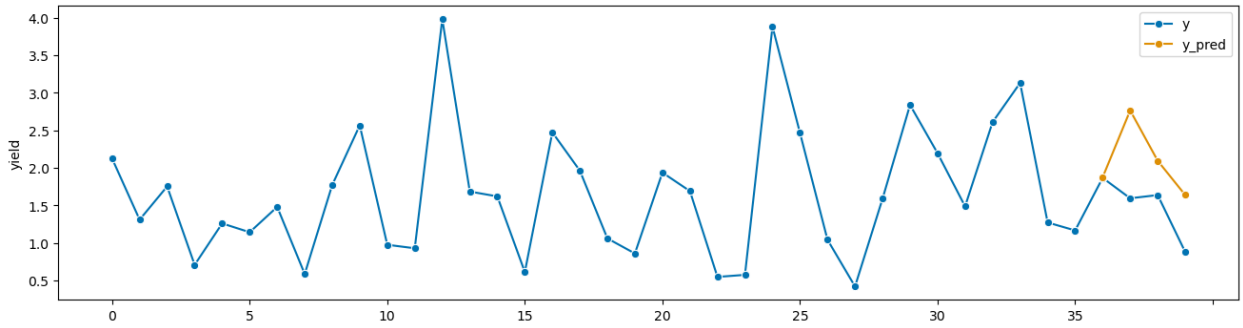


Figure D.62. Time plot, Table 5.14, ForDA w/SITS. Source: MI, OH, KY; Target: Beresford, SD, best model, sMAPE = 34.76%, R = 0.43, MAE = 0.60.

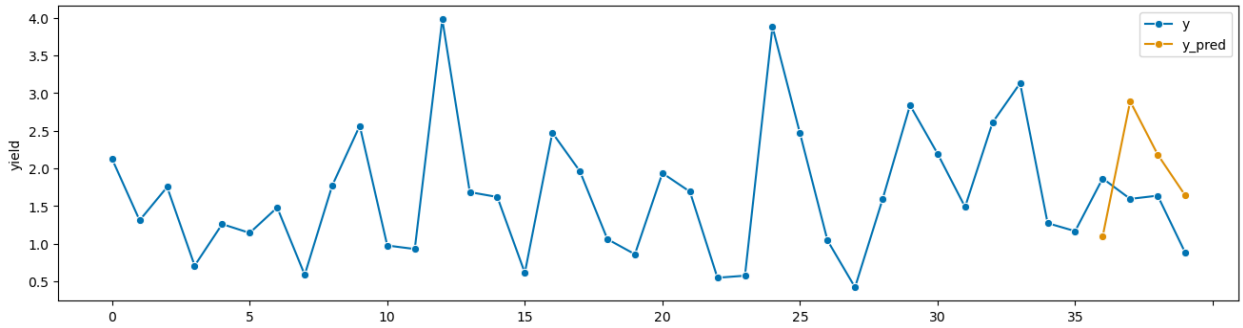


Figure D.63. Time plot, Table 5.14, ForDA, no synthesis. Source: MI, OH, KY; Target: Beresford, SD, best model, sMAPE = 49.79%, R = 0.00, MAE = 0.36.

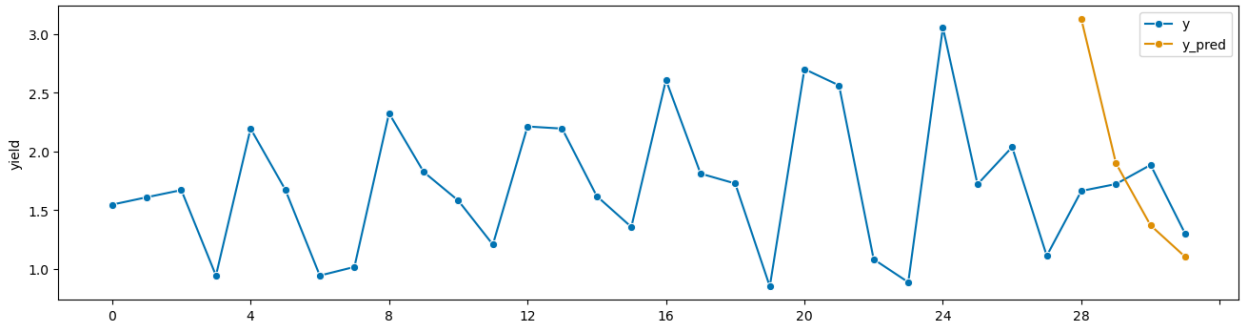

Figure D.64. Time plot, Table 5.15, ForDA, no synthesis. Source: MI; Target: Wooster, OH, sMAPE = 29.65%, R = 0.26, MAE = 0.59.

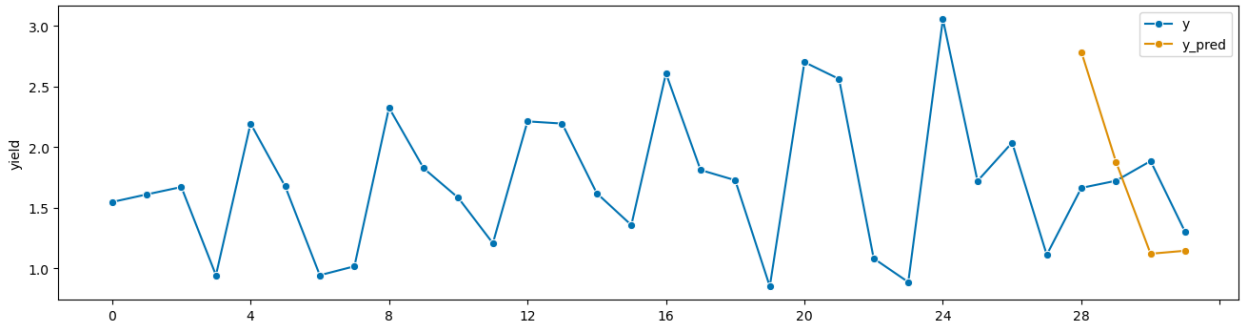


Figure D.65. Time plot, Table 5.15, ForDA w/SITS. Source: MI; Target: Wooster, OH, best model, sMAPE = 30.61%, R = 0.15, MAE = 0.55.

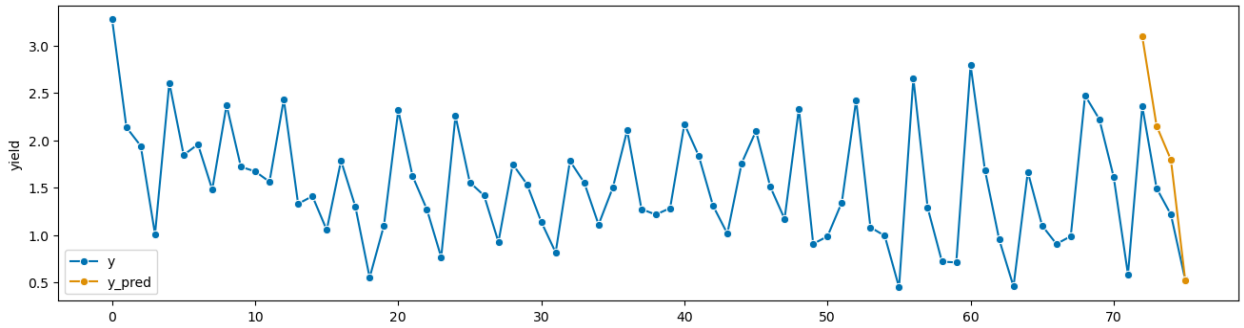


Figure D.66. Time plot, Table 5.15, ForDA, no synthesis. Source: OH; Target: East Lansing, MI, sMAPE = 32.30%, R = 0.99, MAE = 0.49.

# E. FORECASTING RESULTS WITH SYNTHESIS ONLY

Table E.1. Average results for ML-based forecasting. Training data synthesized from local data with SITS, no pre-training, mean results from three runs each.

| Location | Model | sMAPE | MAE | R |
|---|---|---|---|---|
| Wooster | KNN | 23.08 | 0.40 | 0.54 |
| North Baltimore | KNN | 24.27 | 0.31 | 0.90 |
| South Charleston | KNN | 25.65 | 0.37 | 0.94 |
| Wooster | DT | 31.51 | 0.54 | 0.41 |
| North Baltimore | DT | 34.67 | 0.45 | 0.89 |
| South Charleston | DT | 21.46 | 0.30 | 0.90 |
| Wooster | SVR | 25.03 | 0.43 | 0.45 |
| North Baltimore | SVR | 29.88 | 0.34 | 0.92 |
| South Charleston | SVR | 19.79 | 0.30 | 0.94 |
| Wooster | MLP | 19.79 | 0.36 | 0.47 |
| North Baltimore | MLP | 25.80 | 0.29 | 0.97 |
| South Charleston | MLP | 22.60 | 0.33 | 0.97 |
| Wooster | RF | 30.36 | 0.48 | 0.54 |
| North Baltimore | RF | 32.86 | 0.40 | 0.91 |
| South Charleston | RF | 31.60 | 0.42 | 0.94 |
| Wooster | LR | 22.25 | 0.40 | 0.50 |
| North Baltimore | LR | 20.19 | 0.26 | 0.97 |
| South Charleston | LR | 27.57 | 0.42 | 0.96 |
| Wooster | BRR | 20.33 | 0.38 | 0.50 |
| North Baltimore | BRR | 17.04 | 0.22 | 0.97 |
| South Charleston | BRR | 33.92 | 0.50 | 0.96 |
| **Average** | **-** | 25.70 | 0.38 | 0.79 |

Table E.2. Averages by model from Table E.1.

| Model | sMAPE | MAE | R |
|---|---|---|---|
| KNN | 24.33 | 0.36 | 0.79 |
| DT | 29.19 | 0.43 | 0.73 |
| SVR | 24.90 | 0.36 | 0.73 |
| MLP | 22.73 | 0.33 | 0.77 |
| RF | 31.61 | 0.43 | 0.73 |
| LR | 23.34 | 0.36 | 0.81 |
| BRR | 23.76 | 0.37 | 0.81 |

Table E.3. Averages by location of all models in Table E.1.

| Location | sMAPE | MAE | R |
|---|---|---|---|
| Wooster | 24.62 | 0.43 | 0.49 |
| North Baltimore | 26.39 | 0.32 | 0.93 |
| South Charleston | 26.08 | 0.38 | 0.94 |

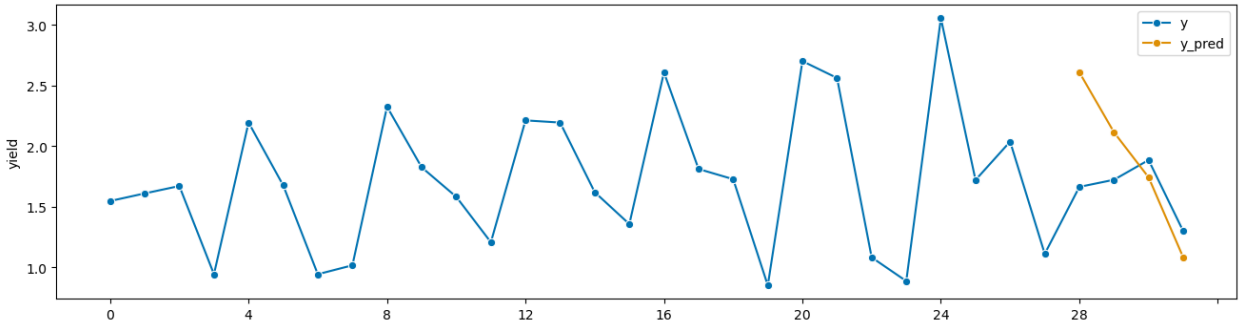


Figure E.1. Time plot for KNN, Wooster, OH. Best model, sMAPE = 22.66%, R = 0.58, MAE = 0.42.

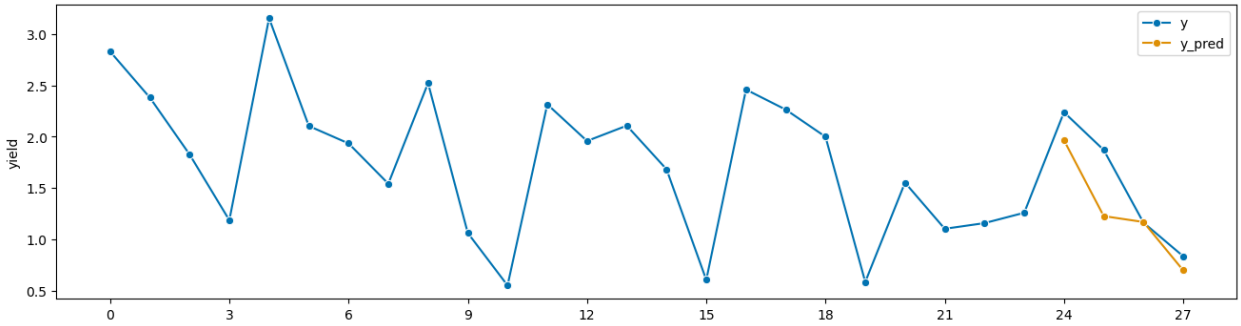


Figure E.2. Time plot for KNN, North Baltimore, OH. Best model, sMAPE = 18.21%, R = 0.90, MAE = 0.26.

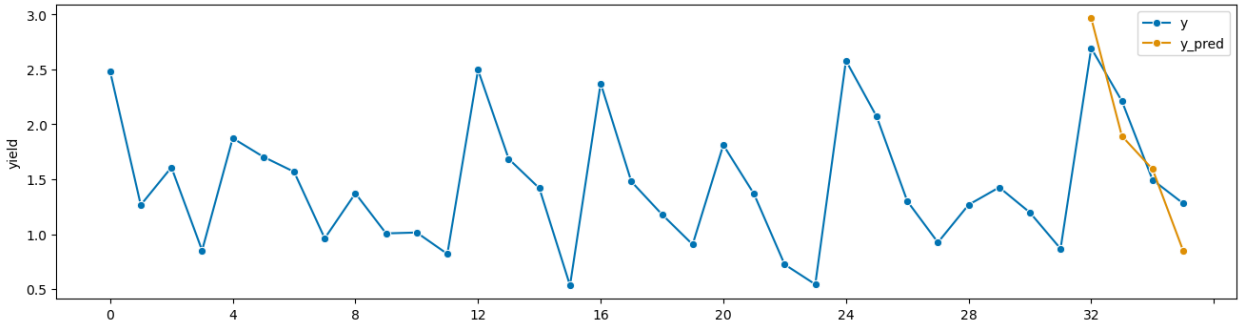


Figure E.3. Time plot for KNN, South Charleston, OH. Best model, sMAPE = 18.00%, R = 0.95, MAE = 0.28.

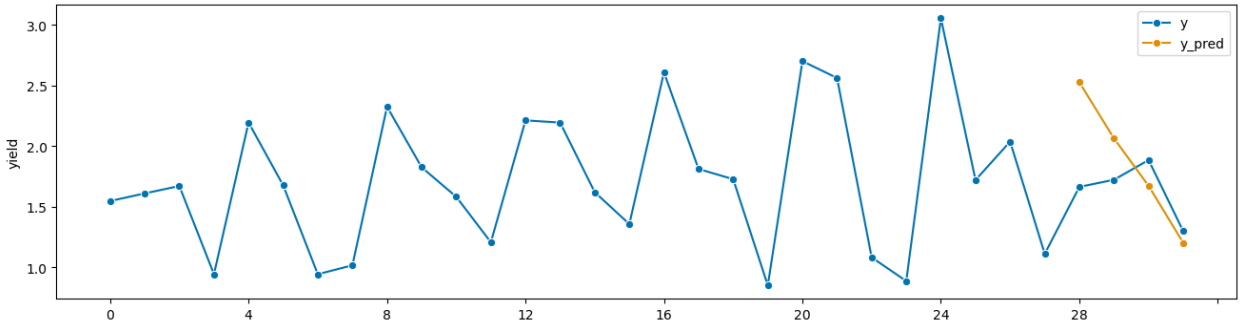


Figure E.4. Time plot for DT, Wooster, OH. Best model, sMAPE = 19.70%, R = 0.50, MAE = 0.38.

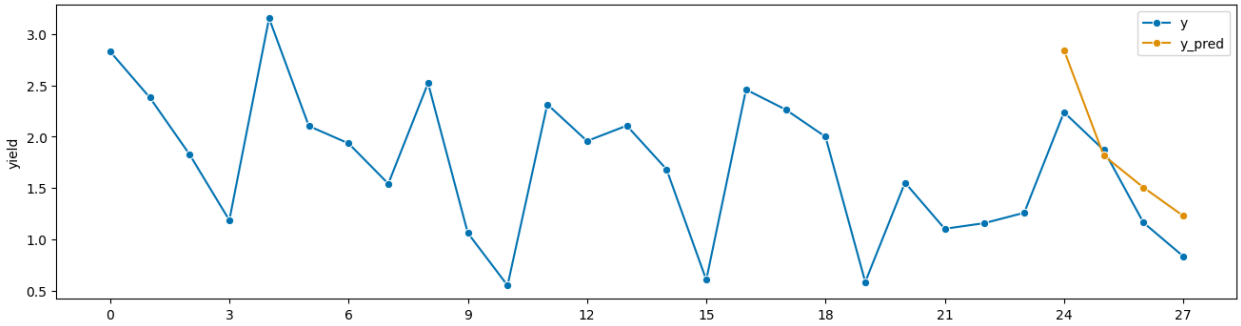


Figure E.5. Time plot for DT, North Baltimore, OH. Best model, sMAPE = 22.57%, R = 0.92, MAE = 0.35.

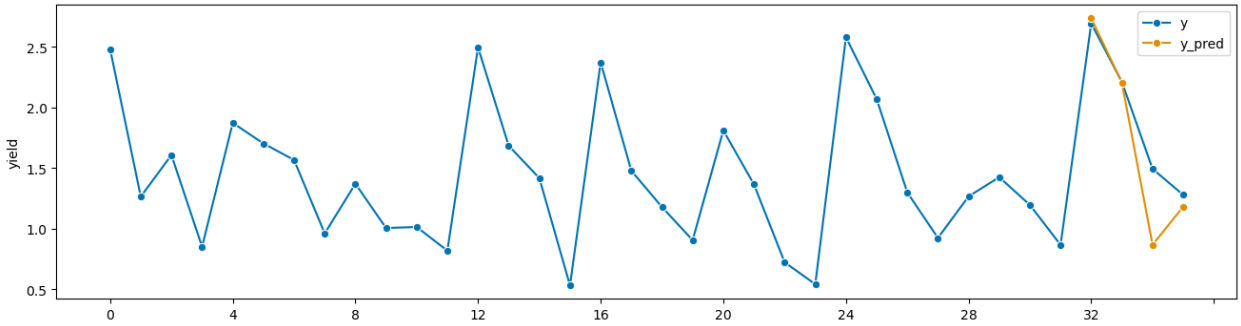


Figure E.6. Time plot for DT, South Charleston, OH. Best model, sMAPE = 15.70%, R = 0.96, MAE = 0.19.

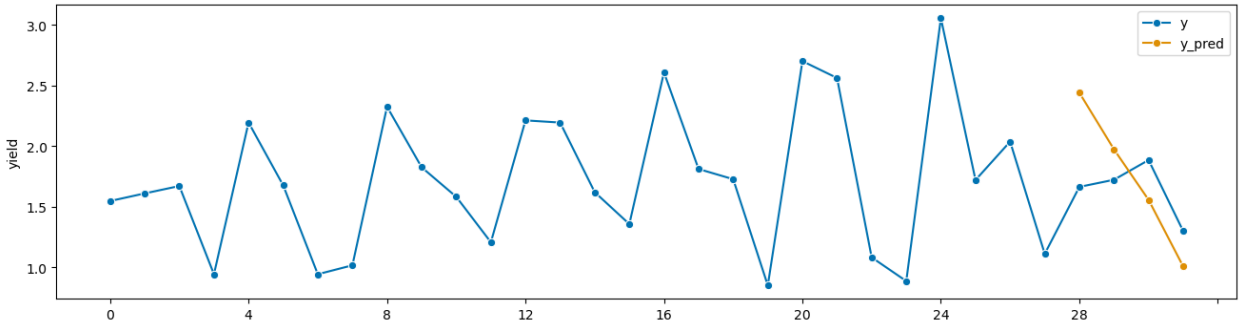


Figure E.7. Time plot for SVR, Wooster, OH. Best model, sMAPE = 23.93%, R = 0.53, MAE = 0.41.

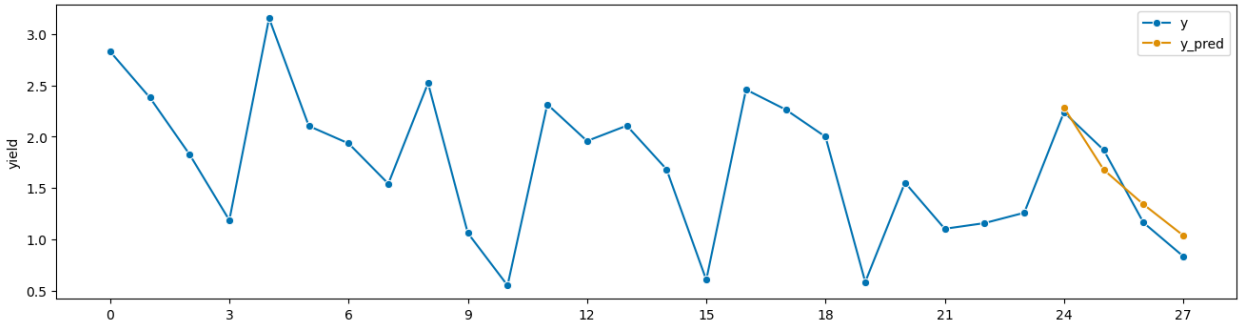


Figure E.8. Time plot for SVR, North Baltimore, OH. Best model, sMAPE = 12.14%, R = 0.97, MAE = 0.15.

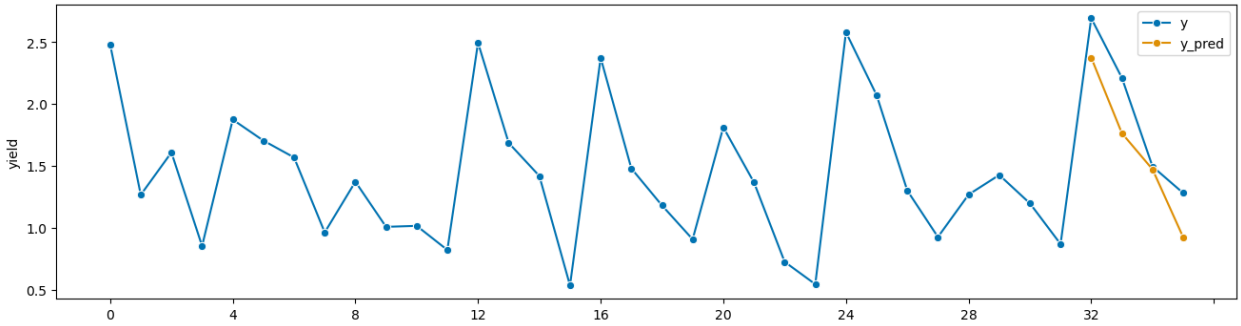


Figure E.9. Time plot for SVR, South Charleston, OH. Best model, sMAPE = 17.28%, R = 0.96, MAE = 0.29.

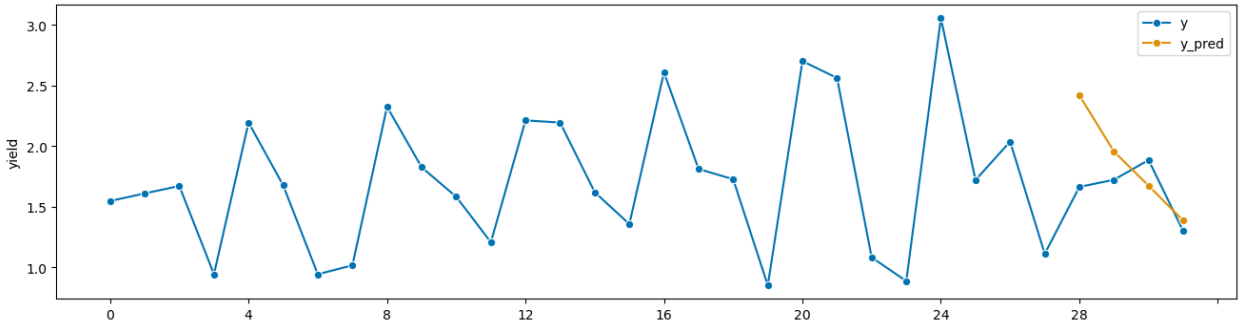


Figure E.10. Time plot for MLP, Wooster, OH. Best model, sMAPE = 16.99%, R = 0.42, MAE = 0.32.

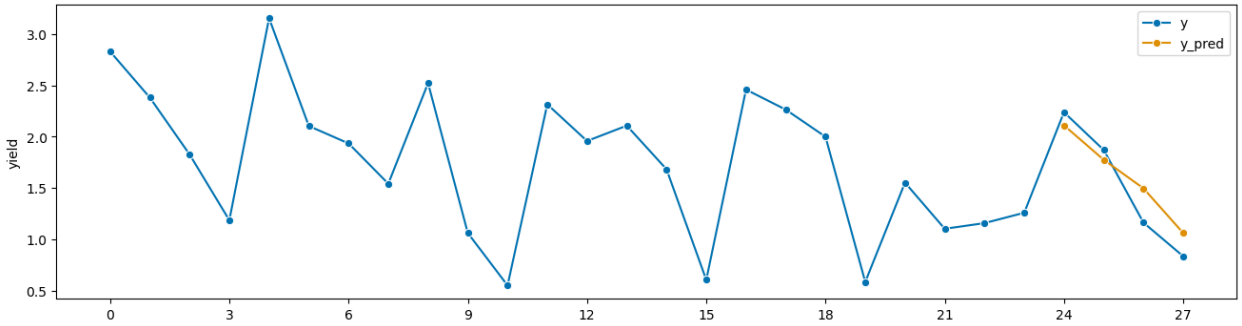


Figure E.11. Time plot for MLP, North Baltimore, OH. Best model, sMAPE = 14.99%, R = 0.98, MAE = 0.20.

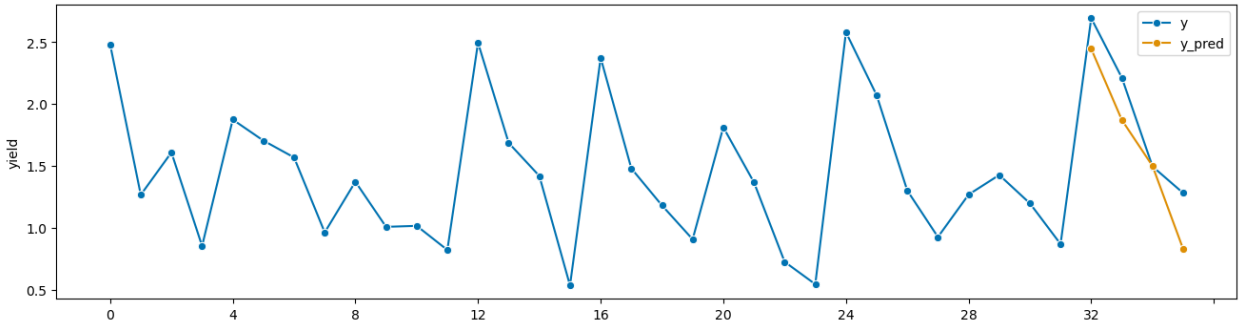


Figure E.12. Time plot for MLP, South Charleston, OH. Best model, sMAPE = 17.43%, R = 0.96, MAE = 0.26.

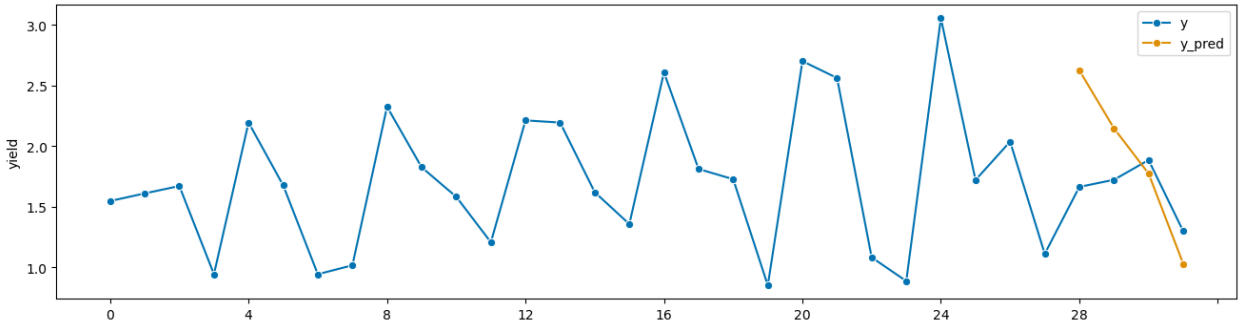


Figure E.13. Time plot for RF, Wooster, OH. Best model, sMAPE = 23.97%, R = 0.61, MAE = 0.44.

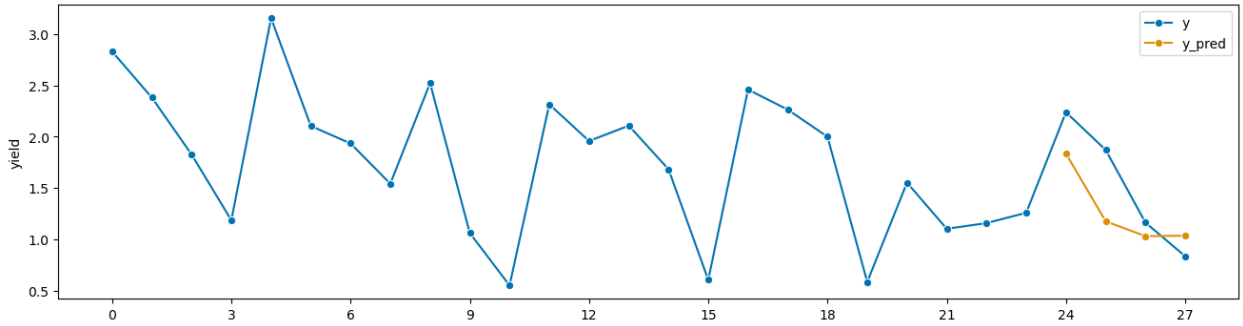


Figure E.14. Time plot for RF, North Baltimore, OH. Best model, sMAPE = 24.66%, R = 0.84, MAE = 0.36.

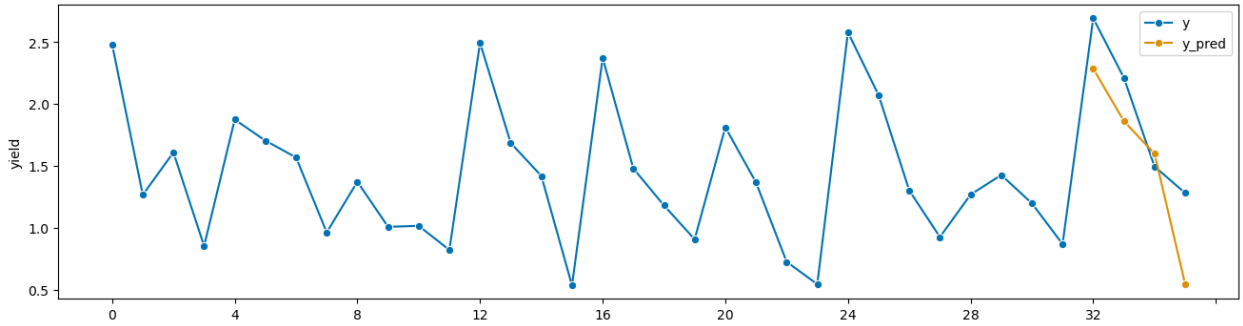


Figure E.15. Time plot for RF, South Charleston, OH. Best model, sMAPE = 30.35%, R = 0.88, MAE = 0.40.

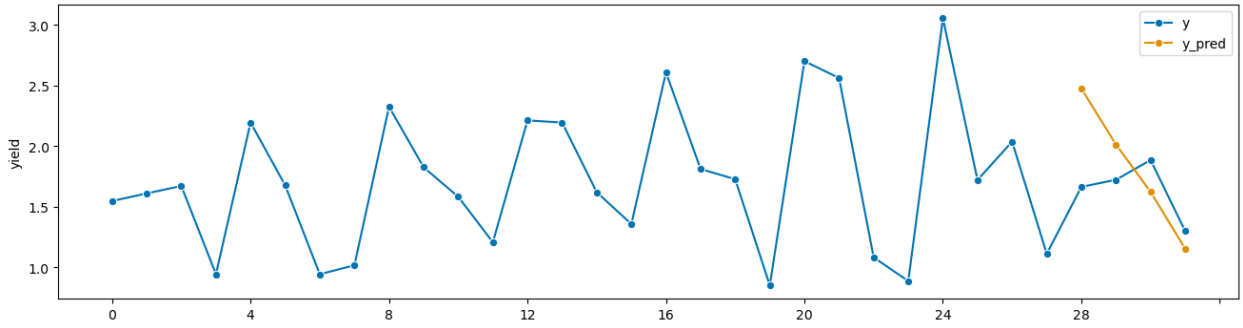


Figure E.16. Time plot for LR, Wooster, OH. Best model, sMAPE = 20.38%, R = 0.50, MAE = 0.38.

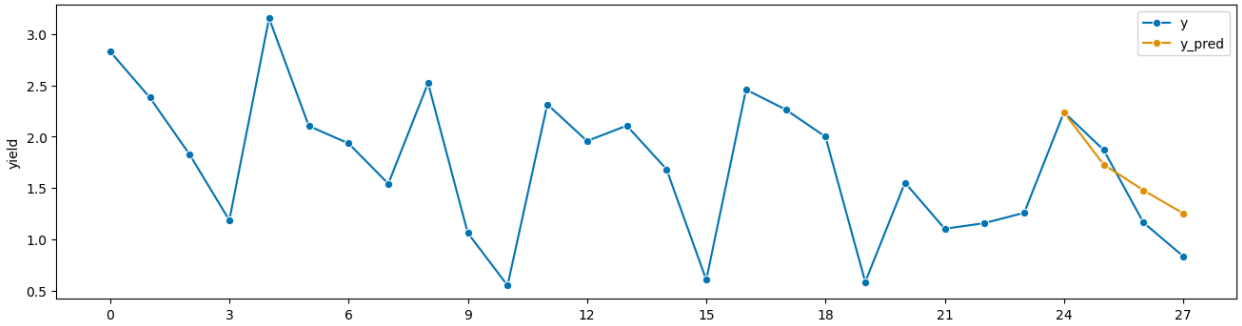


Figure E.17. Time plot for LR, North Baltimore, OH. Best model, sMAPE = 17.91%, R = 0.96, MAE = 0.22.

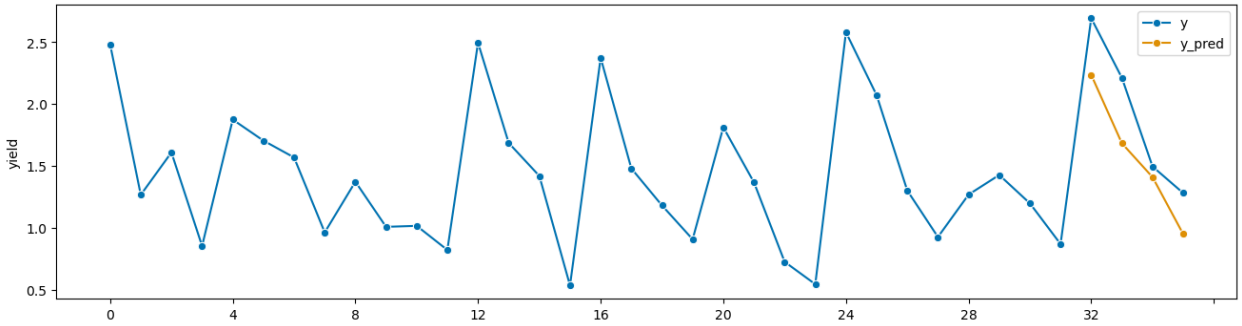


Figure E.18. Time plot for LR, South Charleston, OH. Best model, sMAPE = 20.37%, R = 0.96, MAE = 0.35.

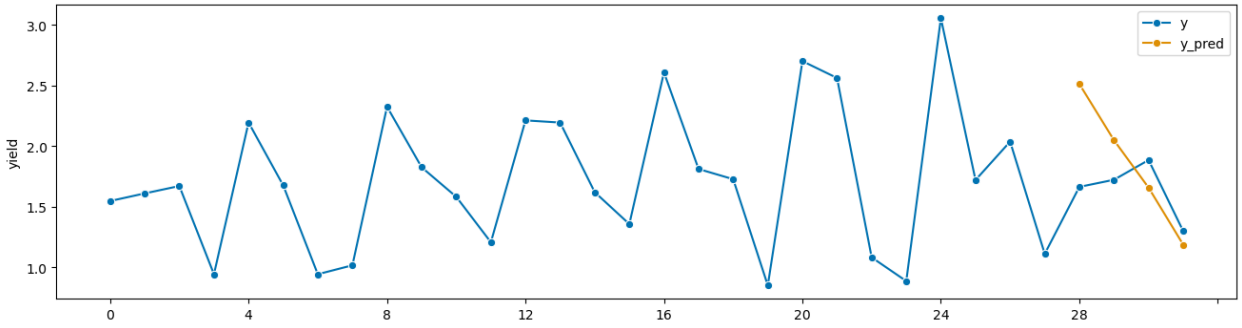


Figure E.19. Time plot for BRR, Wooster, OH. Best model, sMAPE = 19.93%, R = 0.50, MAE = 0.38.

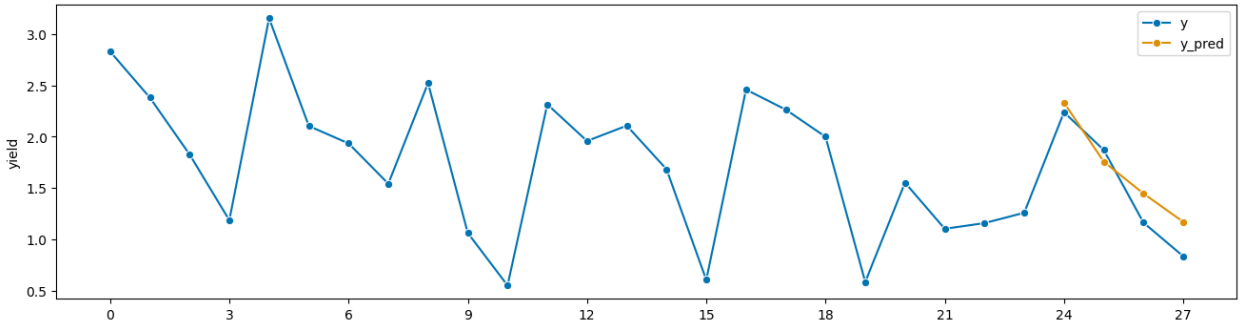


Figure E.20. Time plot for BRR, North Baltimore, OH. Best model, sMAPE = 16.33%, R = 0.97, MAE = 0.21.

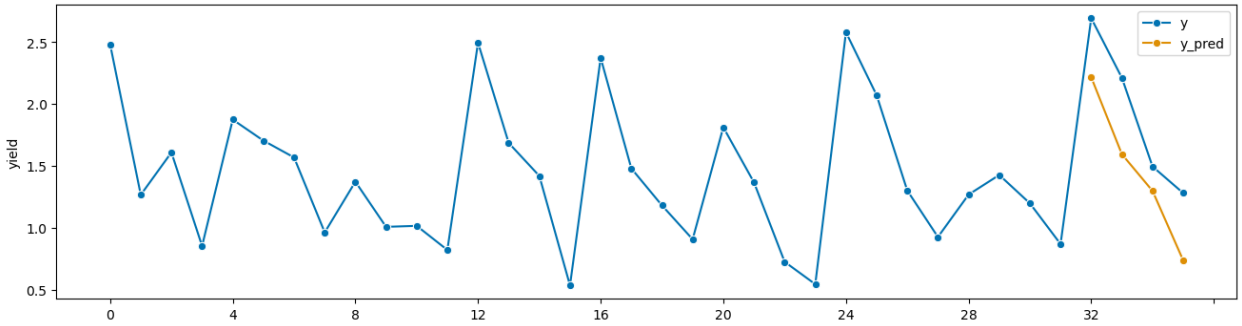


Figure E.21. Time plot for BRR, North Baltimore, OH. Best model, sMAPE = 30.02%, R = 0.96, MAE = 0.46.