Predicting Corn Yields in the United States: A Machine Learning Approach

Amy Steward (asteward@berkeley.edu), Michael White (msteckwhite@berkeley.edu),
Yoko Morishita (yoko_morishita@berkeley.edu), Chad Adelman (chad_adelman@berkeley.edu)

Abstract

Climate change and increasingly unpredictable weather patterns have led to greater volatility in year-to-year corn yields across the United States. Accurate yield prediction helps farmers allocate resources more effectively and supports agricultural policy and food security planning. Many existing models rely on outdated data or fail to capture the complexities in agricultural systems. In order to address this, we created several models which aim to predict future corn yield in the midwestern United States, including a baseline model, a feedforward neural network, and a transformer model. These models incorporate data from several climate and environmental data sources. All models were evaluated using root mean squared error, with data from 2022–2023 used for validation and data from 2024 held out for testing. The transformer model achieved the strongest performance, especially in handling temporal weather patterns. Genetic modification features were consistently among the most predictive inputs.

Introduction

Agricultural yield quantities have benefited or stymied humanity's progress since the invention of plant domestication. Despite the knowledge farmers have collected throughout the millennia, the yield from one year to the next has always been mercurial. This is particularly true in today's changing climate. In this project, we aim to use machine learning to model the relationship between county characteristics in a year and the corn yield expected for relevant regions of the United States. We will focus on predicting corn yield in bushels per acre (bu/ac). By forecasting farm output and, by extension, the profitability of a growing season, farmers can better allocate resources and plan for financial burdens such as upgrading farm equipment. This also informs policymakers about overall production trends, which is crucial for setting agricultural policy, and managing food security initiatives.

To predict corn yield (bu/ac) in seven midwestern US states, our machine learning algorithms take in monthly precipitation and temperature data, percentage of planted corn that is genetically engineered, and location information for each corn yield data point. Predictions come from three models: a baseline model (mean yield), a feedforward neural network, and a transformer model.

Related work

In their paper, Kim, N., et al. (2020) aim to develop a model for predicting corn yield in the midwestern US under drought and heatwave conditions, highlighting corn's sensitivity to heat stress, and poor yield's negative effects on the global food supply. The data includes July and August yields from 2006 to 2015 in five US states: North Dakota, South Dakota, Minnesota, Iowa, and Illinois. Predictors come from satellite images and meteorological data, provided by the National Aeronautics and Space Administration. Their use of multiple data sources (satellite imagery and meteorological data) facilitates a nuanced and comprehensive understanding of the factors influencing corn yield by allowing a deep neural network (DNN) to capture non-linear, high-dimensional interactions. To ensure optimal results, the researchers tested multiple machine learning algorithms (with the DNN performing best), providing a broader evaluation of different modeling strategies for yield prediction. As a result, the model generated is robust to extreme cases, specifically extreme weather such as heatwaves and droughts.

There are several areas in which our work aims to improve on the methods in this paper. The scope of data collection for this analysis is meaningfully constrained in both the time interval and geographical focus; we have access to more recent data, and incorporating 8 years of more recent information will allow our model to adjust to recent changes in corn yield patterns. In line with Kim et al, we try to replicate this comprehensive feature incorporation and model robustness through our feature engineering and modeling testing.

Iniyan, S., et al. (2023) describe similar models to those we are interested in, but focus on crop yields in Maharashtra, a large state in western India. The stated importance of this research is to create a performant model to help farmers decide which crops to grow, in addition to increasing yield by predicting crop yields using climate and soil data and the relationship between these variables. The study's input features were chosen through feature engineering based high-likelihood yield predictors: precipitation, humidity, temperature, planted area, soil type, and crop type. The authors collected 18 years of data from the Indian agricultural website at the district level. They built and evaluated seven models, using R^2, mean absolute error (MAE), and root mean squared error (RMSE) as metrics. For all three metrics, the long short-term memory (LSTM) model, a type of recurrent neural network, performed best.

One strength of the study is the inclusion of derived features through feature engineering, in addition to the raw variables collected. This advanced feature engineering allows the model to pick up on underlying patterns that raw features alone might miss. However, this research could use clearer metrics to determine the success of the model. It labels R^2 as "accuracy", which could lead to misinterpretations of the success metric, since R^2 says more about the explained variance of the collected data than the accuracy of the predictions. This may cause overestimation of the real-world predictive performance of the model. In order to ensure interpretability, our model uses RMSE error and MAE, which are designed to assess continuous numerical predictions compared to actual values.

Dataset

Our primary data for this corn yield project comes from three sources: the National Agricultural Statistics Service (NASS) for yield amounts, the National Oceanic and Atmospheric Administration (NOAA) for temperature and precipitation records, and the Economic Research Service (ERS) within the USDA for data relating to the adoption of genetically engineered (GE) crops.

Initially, the county-level corn data downloaded from NASS included yield information spanning from 1919 to 2024, consisting of 189,249 records. From NOAA, we had 410,222 observations of monthly temperature and precipitation, covering 1985 to 2025. Finally, there were 3500 rows on the topic of GE crops between 2000 and 2024, showing the total percent of all corn planted that was genetically engineered. These GE observations were at the state and year level. To incorporate these into our dataset, we applied the values to each county within the respective state. This simplification was necessary given the time constraints of the project.

We noted that the dataset only included GE data for the years beginning with 2000 and based on our interest in including the effect of growing genetically engineered corn, we elected to subset only the corn yield data from 2000 onwards. Furthermore, we selected the ten states with the most observations, each of which had over 8,000 records. Additional preprocessing included removing redundant features like zip code, any records that were missing the dependent variable (yield in bu/ac), and outlier investigations (none were found). Finally, we removed observations that had no GE information to avoid any issues with missing values, which further winnowed the total number of states to seven. We added one-hot encoding columns for each state-county pair, generated from the `state` and `county` columns, which were useful during experimentation but were ultimately omitted in our final models.

To split into training, validation and test sets, we used the years 2000-2021 for training, 2022-2023 for validation, and 2024 for testing. In addition to aligning with the availability of GE data, this split makes sense considering that the distribution of corn yields in the 1920s is no doubt quite different from modern yields, and so would not prove as useful for training a machine learning model to predict near-future yields. The data splitting yielded the data sizes as shown in Table 1.

Table 1. Data Splitting

Data Subset	Years Included	Size
Training	2000 - 2021	12,418 x 1000
Validation	2022 - 2023	1,091 x 1000
Testing	2024	446 x 1000

EDA indicated that bu/ac of corn are fairly normally distributed. Some yield-by-state amounts show localized minimums, but generally, the values are approximately normal. Looking at yield by year (Fig. 1), there is year-to-year variability in distribution, but the mean remains fairly constant in the ~120-180 bu/ac window. From the heat map of the various GE features and corn yield (Fig. 2), we were able to select just the `all_ge_var` and `stacked_genes` variables as the most predictive of the target variable.

Figures 3 and 4 demonstrate the correlation between corn yield and climate variables. At a high level, the correlation between corn yield and climate data is limited. However, Figure 3 indicates that precipitation in January has the most negative correlation with corn yield, and June has the most positive correlation. Figure 4 reveals that temperatures in any month have a negative correlation with corn yield, but July and August are the strongest.

Based on heatmaps of the correlation between the climate metrics and the corn yield, it was noted that summer rain and spring and fall temperature are most correlated with yields. This observation led us to experiment with various derived features such as mean temperature and total sum of precipitation during a given period of interest. Experimentation details follow later in this paper.

Methods

In order to incorporate the complex dynamics of corn yield in the United States, two machine learning models and a baseline model were built. These attempt to model the relationship between corn yield and the features: location, genetic modifications and weather. The model needed to ensure the data was properly normalized. Feature engineering was leveraged to optimize data for parametric neural networks. RMSE and MAE were used to evaluate these models. Since all models except the baseline used scaling transformations on the outcome variable, evaluation metrics lost their physical meaning (e.g., bu/ac). Before reporting metrics, we first applied the inverse transformation to regain the original scale.

The baseline model provides a benchmark for accuracy which will be compared to more complex models to illustrate relative performance. If these models were performing at similar levels to baseline, that indicates that the features used are not significantly relevant in predicting yield outcome. Focusing on the simplest model that one might use to forecast corn yield, we used the mean value of yields across the entire training set as the baseline.

The first machine learning model we built leveraged the functional API using a feedforward neural network (FFNN) with embeddings and 3 dense layers. An FFNN is a parametric model which processes input data through a series of connected layers, where each layer applies learned weights and activation functions to transform the data. Information flows in one direction without any memory of previous inputs, making them well-suited for structured, non-sequential data. FFNN models excel in approximating non-linear relationships, while still providing weights which allows for limited human interpretability in understanding the importance of each feature in making predictions. The TensorFlow functional API allowed us to include 3 dynamic embedding layers for location inputs and numerical, static inputs, allowing for easy experimentation of features. We began by using heat maps to assess feature correlations with the target (corn yield). The most informative genetic features were `stacked_genes` and `all_ge_var`, which showed the strongest correlations. We normalized all the numerical inputs to reduce scaling and gradient issues. Encoding and embedding state and county helped define relationships between the location data better than a one-hot encoding method would have. Finally, the yield output was normalized using the mean yield from the train dataset.

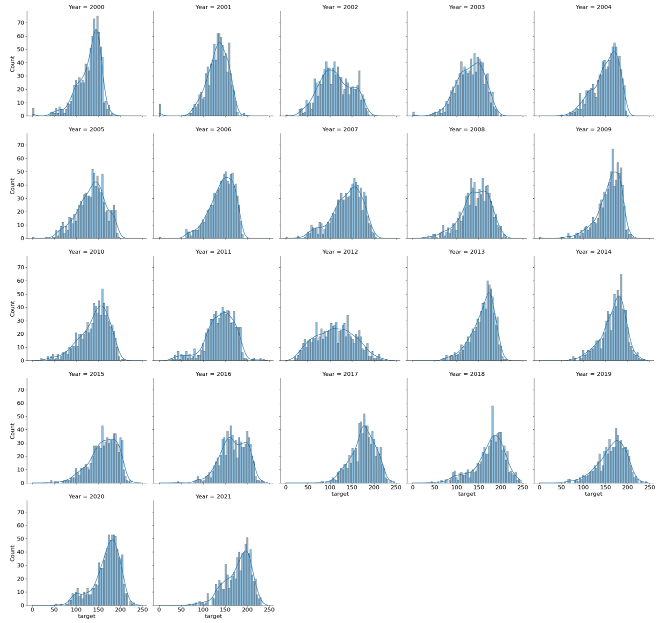


Figure 1. Distribution of yield (bu/acre) by year.

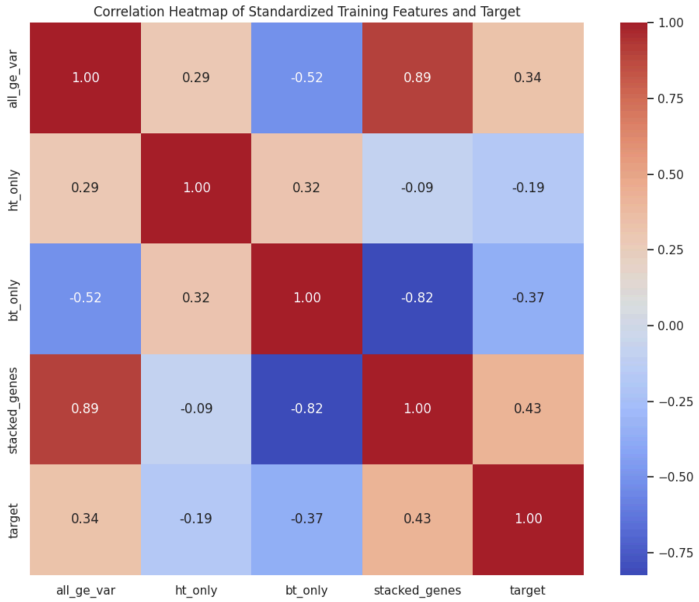


Figure 2. Correlation heatmap of yield and genetically engineered information features.

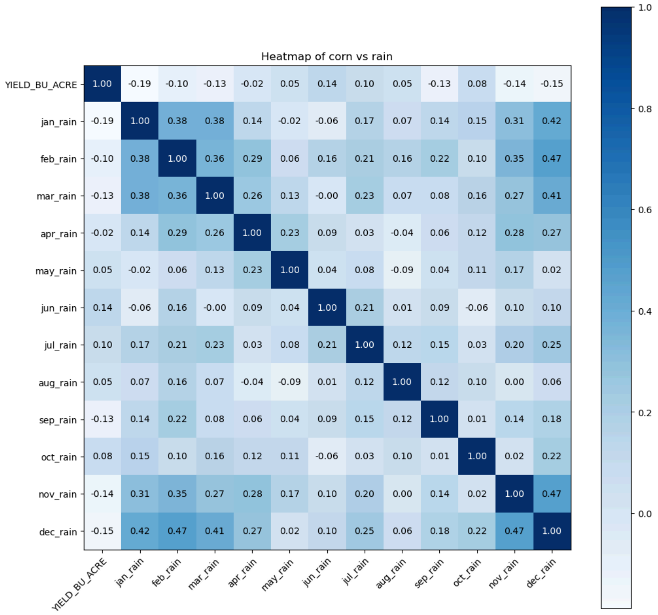


Figure 3. Correlation heatmap of yield and average precipitation by month.

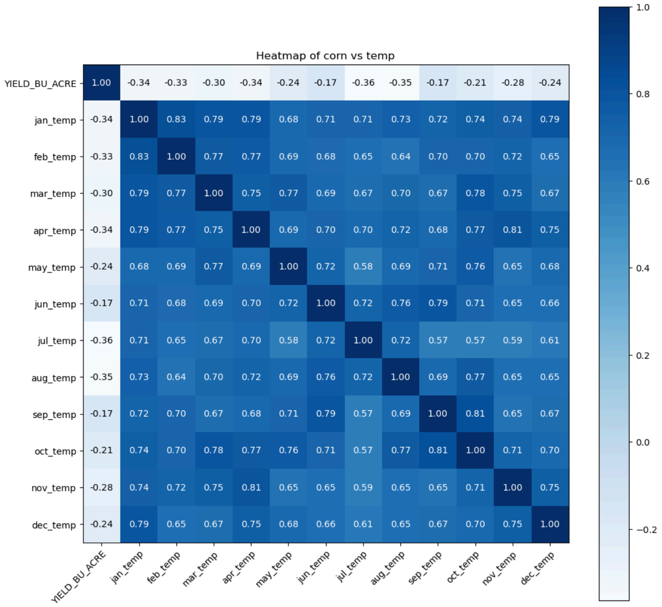


Figure 4. Correlation heatmap of yield and average temperature by month.

The second machine learning model created was a transformer model. This model type was chosen because transformer models use self-attention to weigh the importance of different parts of the input sequence, allowing the model to capture complex relationships across time steps or positions. Unlike recurrent models, transformers process all inputs in parallel, using positional encoding to retain sequence order. Positional encoding was added to build relationships between months, temperature and rain. Additionally, we applied two self-attention heads and allowed the weather data to learn monthly sequences.

Experiments, Results and Discussion

This section presents the experiments conducted with, and results of, the baseline, feedforward neural network (FFNN) and transformer models. Experiments for each model followed a similar pattern and raw and rough experimentation results can be found in the Appendix. For both the FFNN and the Transformer models, we looked at three factors: (1) identifying impactful inputs and normalization, (2) feature engineering and inclusion, and (3) neural network hyperparameter tuning.

Baseline Model: The simplicity of the baseline model precludes experimentation beyond calculating the mean of test set yield values, the result of which is 156.7 bu/ac. Predicting this mean value for all observations in the test set returned an RMSE of 39.57 bu/ac.

Feedforward Neural Network Model: Model experimentation included three phases: (1) initial model with two dense layers and all inputs; (2) multi-feature testing to identify which features provided the best model; and (3) tuning the dense and dropout layers to find the best combination. An initial FFNN model with two hidden layers used only the temperature and precipitation features, ignoring all location and GE information, but this exhibited poor generalization (validation MAE to train MAE ratio of 2.33). Using all available features including embeddings for state county and GE data improved the model performance, but still yielded a validation-to-training generalization ratio of 1.75. Removing the climate data entirely as well as redundant features further improved the results, and we thus used the state/ county information, agricultural district codes, and the two most strongly correlated GE variables (`stacked_genes` and `all_ge_var`) for the final FFNN model. Hyperparameter tuning primarily consisted of searching across the number of hidden layers and number of neurons in each layer. The final model had the architecture outlined in Table 2 (excluding input and output layers). This model resulted in a test RMSE of 20.79 bu/ac.

Table 2. FFNN model architecture.

Layer Type	Specification	Activation
Dense	256 units	Rectified Linear Unit (ReLU)
Dense	128 units	ReLU
Dropout	50%	-
Dense	64 units	ReLU
Dropout	30%	-

Transformer Model: Despite good results from the FFNN model described above, we believed there might be sequential information in the climate data that feedforward architecture could not capture, but that we might extract with a transformer architecture. The baseline transformer model took as inputs twelve months of precipitation and temperature data, in addition to the features used in the FFNN including `all_ge_var` and `stacked_genes` variables. It employed one multihead attention (MHA) block with 4 heads and 64 keys and one feedforward layer after the MHA in order to learn patterns among months. This was followed by a standard three-layer FFNN architecture. Tuning and evaluation revealed improved performance from adding positional encodings to keep track of the sequence of months in the climate data. Including the `year` feature also reduced overall error. To support corn harvest predictions for August through October, we restricted the climate data to January through July. A second layer in the feedforward portion of the self-attention mechanism also reduced the RMSE, and the overall addition of neuron counts in all layers contributed to model improvement. The incorporation of a second MHA block improved the generalization performance on the validation and test sets, as well as resulted in a 2% improvement in RMSE on the test data. The final selected transformer model has the architecture seen in Table 3 (excluding input and output layers). The associated test RMSE was 18.02 bu/ac.

Table 3. Transformer model architecture.

Layer Type	Specification	Activation
Dense	32 units	Linear
MHA 1 w/ Dropout, Add & Normalization	4 heads → 15% dropout → Add to positional encoding → Normalization	-
MHA 2 w/ Dropout, Add & Normalization	4 heads → 15% dropout → Add to MHA 1 output → Normalization	-
Dense x2	128 units → 32 units	Linear x2
Add	Combine outputs from both MHA blocks	-
Normalization	-	-
Dense	1024 units	ReLU
Dropout	50%	-
Dense	512 units	ReLU
Dropout	20%	-
Dense	64 units	ReLU
Dropout	30%	-

Performance of Final Models: The performance of each of the final models, including the baseline, is shown in Table 4. Choosing several metrics to evaluate the models allows for a broader understanding of different aspects of how each of the models performed. The RMSE measures error in the same unit as the target variable and emphasizes larger errors, while MAE is a linear metric for error and is more robust to outliers.

To address overfitting concerns, we implemented several mitigation strategies throughout model development. First, we removed inputs that appeared to be irrelevant or redundant—such as certain weather or location features—to reduce noise and simplify the input space. Second, we limited the number of hidden layers in our neural networks to avoid excessive model complexity that could lead to overfitting. Finally, we split the data into training, validation, and testing data sets. We used the training set for training the model, the validation set for tuning the models and the test set only once all models had been built. Having this split ensured that the model was evaluated on unseen data, decreasing the likelihood a model was overfit to the data. This resulted in RMSE and MAE values that were similar across the training, validation, and testing datasets for all three of our models.

While both models saw a significant improvement over the baseline, the transformer model ultimately outperformed the FFNN model in all of the metrics measured for the test dataset. As such, we used the transformer model as our final model. The success of this model was likely driven by the aforementioned strength of the transformer architecture in learning patterns from sequential data.

Table 4. Results of the baseline and two improvement models.

Metric:	**RMSE**			**MAE**		
Model	Train	Val	Test	Train	Val	Test
Baseline (Mean Yield)	32.60	39.57	41.45	25.43	34.50	35.22
Feedforward NN	23.34	20.24	20.79	18.03	15.71	16.91
Transformer w/ MHA	13.84	17.03	18.02	10.72	12.96	14.37

Conclusion

The key finding from this study is that the transformer model consistently achieved the lowest RMSE on all data sets. While the FFNN demonstrated reasonable generalization, outperforming its training RMSE on the validation set, it still did not surpass the performance of the transformer. This superior performance can be attributed to the transformer's self-attention mechanism, which explicitly models dependencies among all time steps or features, regardless of their relative positions in the sequence. Consequently, transformers are capable of jointly capturing short-term influences (e.g., monthly climate changes) and long-term trends (e.g., seasonal patterns), making them particularly well-suited to the sequential and temporal dynamics frequently found in agricultural datasets. This advantage also enabled the inclusion of a broader range of features, such as climate variables and year, which tended to introduce noise when used with the FFNN. Notably, among the transformer architectures explored, the 2-layer transformer achieved an effective balance, offering sufficient capacity to model data complexity without overfitting. By contrast, increasing the number of layers without adequate regularization or additional data led to rapid overfitting and declined predictive performance.

This project faced several limitations that could be addressed with greater resources or extended timelines. Using state-level rather than county-level GE data constrained the spatial resolution of predictions; incorporating finer-grained data would likely improve performance. Important soil variables—including pH and irrigation—were excluded due to insufficient or unavailable data, and previous-year rainfall, which might influence current yield, was also omitted. The analysis was limited to data from only seven states, further restricting the generalizability of the model. Exploring alternative data partitioning strategies (e.g., splitting by state) and implementing leave-one-out cross-validation could also help enhance the robustness and generalizability of the results. Finally, when developing models to predict yields at finer geographic levels (such as state or county), performance can be evaluated separately for each level. In the appendix (Figure 5), we included a heatmap that illustrates the variance between the transformer model's predictions and actual corn yields at the county level for the test dataset. This highlights that there is room to improve the balance of performance across sub-groups, which could ultimately lead to better overall model optimization.

Contributions

(**GitHub Repository**: https://github.com/mwhite25/207-corn)

Amy Steward: Cleaned data from NASS, joined together in a table, and began EDA. Initial write-ups for the following sections: Dataset, Experiments, Methods. Contributed to and expanded the following sections: Abstract, Introduction. Managed the following .ipynb files: EDA, raw_data_processing, 2fx_api_FFNN_AStew_model_2 (2), TUNED_transformer_arch_AStew_model_3, Baseline_transformer_arch_AStew_model_3, 7_mos_only_TUNED_transformer_arch_AStew_model_3.

Michael White: Created a linear regression, a FFNN with one-hot encoding and a FFNN with embeddings models for predicting the outcome variable. Wrote the Dataset section and contributed to/ edited the following sections: Introduction, Methods, Experiments. Reviewed and edited Abstract. Managed the following .ipynb files: replicate_amys_data, replicate_yokos_data, reformat_data, ml_models.ipynb and map_visualization.

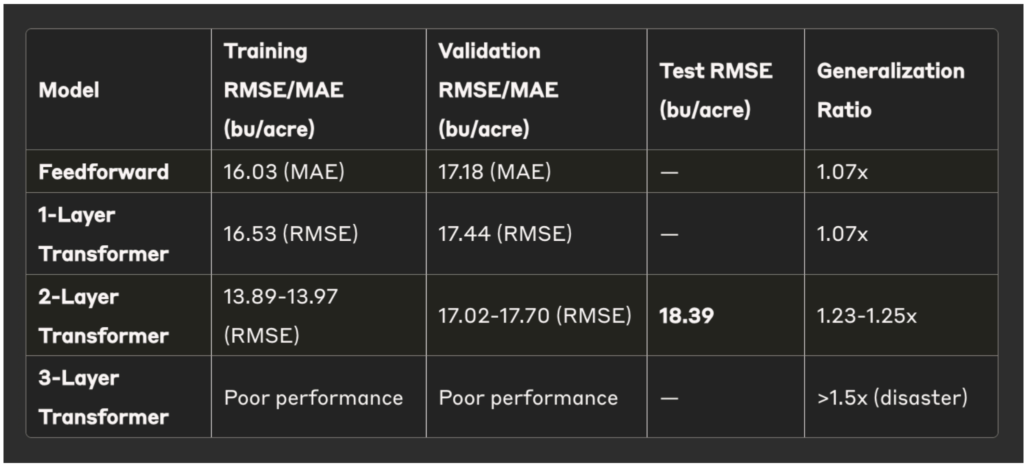
Yoko Morishita: Cleaned climate data and combined with corn data. Conducted initial EDR for climate data. Created Linear Regression, Random Forest, Gradient Boosting, and FFNN models.. Wrote Conclusion and contributed to and expanded the following sections: Related Work, Datasets, Experiments. Reviewed and edited Methods and Experiments. Managed the following .ipynb files: 207_CornYield_rain_wrangling_county, 207_CornYield_ClimateEDA, 207_CornYield_ML

Chad Adelman: Wrote code to split data into training, testing, val. Created a Feedforward Neural Network model for predicting the outcome variable. Initial write-ups for the following sections: Abstract, Introduction, Related Work, Experiments. Managed the following .ipynb files: splitting, yield_model.

References

1. Kim, N., Na, S.I., Park, C.W., Huh, M., Oh, J., Ha, K.J., Cho, J., & Lee, Y.W. (2020). An artificial intelligence approach to prediction of corn yields under extreme weather conditions using satellite and meteorological data. *Applied Sciences*, 10(11). https://www.mdpi.com/2076-3417/10/11/3785?
2. Iniyan, S., Varma, V.A., & Naidu, C.T. (2023). Crop yield prediction using machine learning techniques. *Advances in Engineering Software*, 175. https://doi.org/10.1016/j.advengsoft.2022.103326

Appendix



Model	Training RMSE/MAE (bu/acre)	Validation RMSE/MAE (bu/acre)	Test RMSE (bu/acre)	Generalization Ratio
Feedforward	16.03 (MAE)	17.18 (MAE)	—	1.07x
1-Layer Transformer	16.53 (RMSE)	17.44 (RMSE)	—	1.07x
2-Layer Transformer	13.89-13.97 (RMSE)	17.02-17.70 (RMSE)	**18.39**	1.23-1.25x
3-Layer Transformer	Poor performance	Poor performance	—	>1.5x (disaster)

Table 5. Performance of transformer model by number of transformer layers with FFNN performance

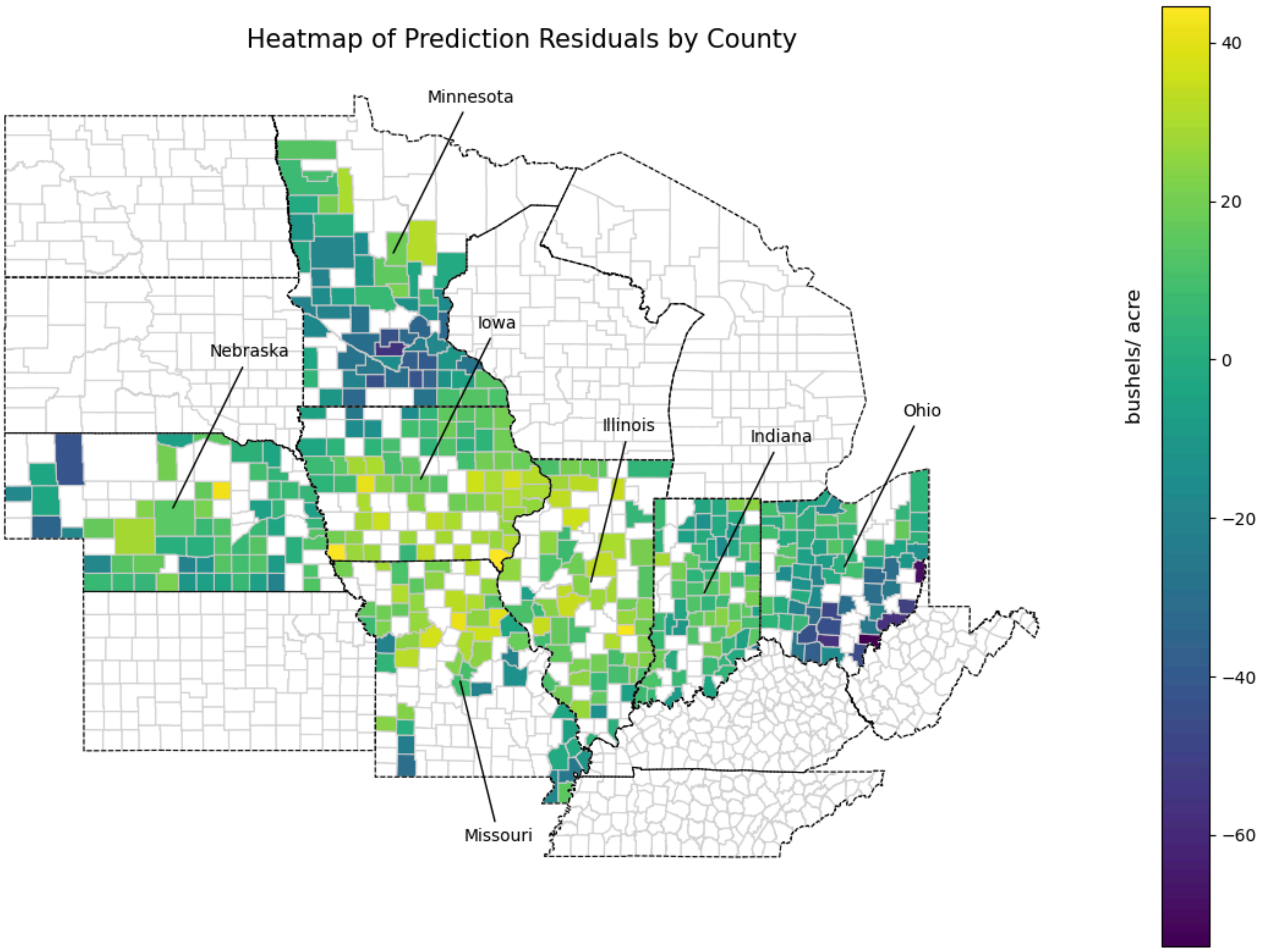


Figure 5. Residuals between transformer model predictions and actual corn yield by county for the test dataset