A GNN-RNN Approach for Harnessing Geospatial and Temporal Information: Application to Crop Yield Prediction

Joshua Fan[*1], **Junwen Bai**[*1], **Zhiyun Li**[*2], **Ariel Ortiz-Bobea**[2], **Carla P. Gomes**[1]

[1] Department of Computer Science, Cornell University, USA
[2] Department of Applied Economics & Management, Cornell University, USA
{jyf6, jb2467, zl547, ao332}@cornell.edu, gomes@cs.cornell.edu

Abstract

Climate change is posing new challenges to crop-related concerns, including food insecurity, supply stability, and economic planning. Accurately predicting crop yields is crucial for addressing these challenges. However, this prediction task is exceptionally complicated since crop yields depend on numerous factors such as weather, land surface, and soil quality, as well as their interactions. In recent years, machine learning models have been successfully applied in this domain. However, these models either restrict their tasks to a relatively small region, or only study over a single or few years, which makes them hard to generalize spatially and temporally. In this paper, we introduce a novel graph-based recurrent neural network for crop yield prediction, to incorporate both geographical and temporal knowledge in the model, and further boost predictive power. Our method is trained, validated, and tested on over 2000 counties from 41 states in the US mainland, covering years from 1981 to 2019. As far as we know, this is the first machine learning method that embeds geographical knowledge in crop yield prediction and predicts crop yields at the county level nationwide. We also laid a solid foundation by comparing our model on a nationwide scale with other well-known baseline methods, including linear models, tree-based models, and deep learning methods. Experiments show that our proposed method consistently outperforms the existing state-of-the-art methods on various metrics, validating the effectiveness of geospatial and temporal information.

Introduction

Climate change (**?**) has become a real and pressing challenge that poses many threats to our everyday life. Besides the evident extreme events (**?**), climatic variations also gradually impact the yields of major crops (**?**). Crop production is vulnerable and sensitive to fluctuations in climatic factors such as temperature, precipitation, soil, moisture and many other factors (**?**). As the planet gets warmer, all of these swiftly-changing factors could perturb annual crop yields. Many recent works urge rethinking crop production practices under climate change (**??**), which motivates the crop yield prediction problem (**?**). Crop yield prediction can help with food security (**?**), supply stability (**?**), seed breeding (**?**), and economic planning (**?**).

However, crop yield depends on numerous complex factors including weather, land, water, etc. While there exist specialized process-based models to simulate crop growth (**?**), they often produce highly-biased predictions, require strong assumptions about management practices, and are computationally expensive (**?**). Therefore, in recent years, powerful yet inexpensive machine learning methods have been widely adopted in crop yield prediction and demonstrated impressive results (**??????**). Machine learning models (especially deep learning models) benefit from the large capacity, sophisticated non-linearity and mature techniques inherited from other application domains.

Despite the enormous amount of machine learning papers for crop yield prediction, many of them share similar methods. Among around 70 papers we surveyed, 48 used neural networks, 10 used tree-based methods (e.g. decision tree, random forest), and 10 used linear regressions (e.g. lasso). These methods often only differ in location (US, Brazil, India), study granularity (province, county, site/farm), crop types (soybean, corn), and time range (weeks to years). Similar findings are also reported in (**?**). For many relatively small self-collected datasets, simpler models are preferred. But these models may not perform well on a large and diverse region like the entire US.

In this paper, we compare various machine learning techniques on a nationwide scale, and propose a novel graph-based framework, GNN-RNN, which integrates both geospatial and temporal knowledge into inference. We train and compare these methods on over 2,000 counties from 41 states in the US mainland, with data covering years 1981 to 2019. There are up to 49 climatic and soil factors for each county, including precipitation, temperature, wind, soil moisture, soil quality, etc. Most of these factors vary across time within the year, or vary across different soil layers. All features are publicly available from sources such as PRISM, NLDAS, and gSSURGO. Furthermore, although not every county has crops planted, USDA provides corn yield labels from 41 states, and soybean yields from 31 states.

Recent works using machine learning demonstrate promising results for crop yield prediction (**?**). Nevertheless, these methods treat each county as an i.i.d. sample in their models, which is plausible in small regions but may

[*]Equal contribution.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

not fully utilize the spatial structure of a larger region. For instance, if one county has a splendid harvest, the neighborhood counties are very likely to have high yields as well, which violates the independence assumption. It is also problematic to treat counties in the northern US and southern US as i.i.d. samples, as the distribution of their climatic and soil conditions is very different. We hence introduce **graph neural networks** (GNN) (**?**) to take into account the geographical relationships among counties. When the model makes a prediction for a county, it can combine the features from neighboring counties with its own features to boost the predictive power. GNN models have been successful in many tasks such as election prediction (**?**) and COVID forecasting (**?**). Additionally, we show that GNNs can work synergistically with RNNs to combine both geospatial and temporal information for prediction. We will show the novel **GNN-RNN** model can achieve superior performance in experiments. As far as we know, our work is the first to incorporate geographical knowledge into crop yield prediction. To further lay a solid foundation in this task, we compare various widely adopted machine learning methods with our method, including lasso (**?**), gradient boosting tree (**?**), CNN, RNN, CNN-RNN (**?**). These are also predominant methods among the papers we surveyed. The experimental results show that GNN-based methods consistently outperform these existing models on the nationwide benchmark. On both RMSE and R^2, our GNN-RNN outperforms the state-of-the-art CNN-RNN model by 10%.

Methods

Problem Formulation

In crop yield prediction, we denote each county's climatic features by $\mathbf{x}_{c,t}$ and ground-truth crop yield (for a particular crop) by $y_{c,t} \in \mathbb{R}$, where c, t represent county and year respectively. Each $\mathbf{x}_{c,t}$ contains four types of features (detailed descriptions of these features can be found in the Experiments section): weather features $\mathbf{x}^w_{c,t} \in \mathbb{R}^{n_w \times 52}$, land surface features $\mathbf{x}^l_{c,t} \in \mathbb{R}^{n_l \times 52}$, soil quality features $\mathbf{x}^s_c \in \mathbb{R}^{n_s \times 6}$, and some extra features (e.g. crop production index) $\mathbf{x}^e_c \in \mathbb{R}^{n_e}$. Namely, $\mathbf{x}_{c,t} = (\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}, \mathbf{x}^s_c, \mathbf{x}^e_c)$. We denote the number of weather, land surface, soil quality, and extra variables as n_w, n_l, n_s, n_e respectively. Among these features, $\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}$ change both spatially and temporally, while $\mathbf{x}^s_c, \mathbf{x}^e_c$ are county-specific and remain stable over time. The goal is to predict $y_{c,t}$ given $\mathbf{x}_{c,t}$. Recent work (**?**) also showed features from past years can help with the prediction, so we reformulate our task as predicting $y_{c,t}$ with $\{\mathbf{x}_{c,t}, \mathbf{x}_{c,t-1}, ..., \mathbf{x}_{c,t-\Delta t}\}$. Δt is the length of year dependency. If $\Delta t = 0$, the model will not consider features from prior years.

Per-Year Embedding Extraction

Regardless of whether the models use historical features or not, the first step is always to extract an embedding for each year from $\mathbf{x}_{c,t}$. Then a prediction can be made based on the embedding from the current year or the embeddings from the last few years.

The four types of features $\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}, \mathbf{x}^s_c, \mathbf{x}^e_c$ have different structures. Using a uniform neural network to extract the embedding may not effectively exploit the structure in the raw data. For example, weekly features $\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}$ naturally incorporate a temporal order, but county-specific soil features $\mathbf{x}^s_c$ do not change temporally and are measured at different depths underground. Therefore, we use separate neural networks to process the differently structured-parts from $\mathbf{x}_{c,t}$:

$$\begin{aligned} \mathbf{h}^{wl}_{c,t} &= f_{wl}(\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}) \\ \mathbf{h}^s_c &= f_s(\mathbf{x}^s_c) \\ \mathbf{h}_{c,t} &= (\mathbf{h}^{wl}_{c,t}, \mathbf{h}^s_c, \mathbf{x}^e_c) \end{aligned} \tag{1}$$

$f_{wl}(\cdot)$ handles the features that vary over time. Since land surface features like soil moisture from $\mathbf{x}^l_{c,t}$ are weekly data closely related to weather, we concatenate $\mathbf{x}^l_{c,t}$ and $\mathbf{x}^w_{c,t}$ before further passing to f_{wl}. Given the temporal order, an RNN or a CNN can be used for f_{wl} to facilitate information aggregation along the time axis. On the other hand, $f_s(\cdot)$ aggregates information along soil depths. We use CNN as the architecture for f_s. $\mathbf{x}^e_c$ only contains six scalar values, so we directly pass it to the output embedding. The final embedding $\mathbf{h}_{c,t}$ is the concatenation of $\mathbf{h}^{wl}_{c,t}, \mathbf{h}^s_c, \mathbf{x}^e_c$.

Temporal Dependency

Though new crops are planted every year and yields primarily depend on climatic factors within one year, it has been observed that the trend and variations captured by recent history can be very informative for prediction (**?**). For example, crop yields have tended to increase over the past few decades due to improvements in technology and genetics (**?**). While data on the underlying technological improvement is unavailable (**?**), we can observe recent trends in crop yield. Our per-year embedding extraction makes it easy to incorporate historical knowledge. All we need is an RNN that reads the per-year embeddings from the current year and several prior years. The output from the last time step would be our prediction for the crop yield of the current year:

$$\widehat{y}_{c,t} = r(\mathbf{h}_{c,t-\Delta t}, ..., \mathbf{h}_{c,t-1}, \mathbf{h}_{c,t}) \tag{2}$$

where $r(\cdot)$ is an RNN, and $\mathbf{h}_{c,t'}$ is the embedding from year t' for county c. The model described so far follows the CNN-RNN framework, which has previously been shown to outperform single-year NN models (**?**).

Incorporating Geographical Knowledge

Eq. 2 shows how one can extend the use of embeddings from Eq. 1 temporally. Then a natural question is, Can we take advantage of the embeddings geospatially as well? Intuitively, if a county has good yields, nearby counties tend to have good yields as well. The weather and soil conditions should also transition smoothly across the continent. The additional features from neighboring counties could boost the prediction if used properly. A recent success in COVID-19 forecasting (**?**) with similar insights could further support incorporating geographical knowledge, where the graph-based representation learning greatly improves case prediction.

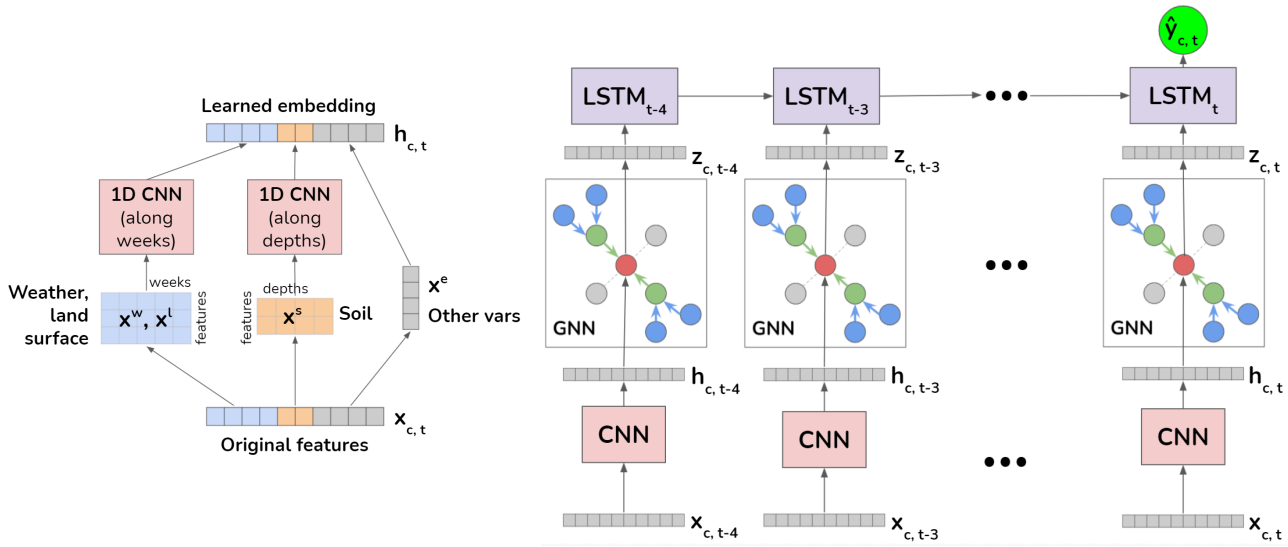

Figure 1: **Left**: The CNN model used for per-year embedding extraction. **Right**: Our overall GNN-RNN framework. For each county c and year t', the CNN extracts an embedding $\mathbf{h}_{c,t'}$. Then we apply a GNN to refine each year's embedding by aggregating information from neighboring counties, producing a new embedding $\mathbf{z}_{c,t'}$. Finally, an LSTM processes the embeddings from each year and outputs the yield prediction $\widehat{y}_{c,t}$.

Graph Neural Network Graph Neural Network (GNN) (**?**) is a novel type of neural network proposed to unravel the complicated dependencies inherent in graph-structured data sources. Given its strong power in representation learning, GNN has demonstrated prominent applications in chemistry (**?**), traffic (**?**), biology (**?**), and computer vision (**?**) with sophisticated model architectures (**???**). Formally, a graph is denoted by $G = (V, E)$ where V is the set of nodes and E is the set of edges between nodes. In our crop yield prediction task, each node is a county. E is represented as a symmetric adjacency matrix $A \in \{0, 1\}^{N \times N}$ where $A_{i,j} = 1$ if two counties $v_i, v_j \in V$ border and $A_{i,j} = 0$ otherwise. N is the total number of counties. Each node is associated with $\mathbf{x}_{c,t}$ for every year.

GraphSAGE A popular GNN model, GraphSAGE, (**?**) is a general framework that leverages node feature information and learns node embeddings through aggregation from a node's local neighborhood. Unlike many other methods based on matrix factorization and normalization (**?**), GraphSAGE simply aggregates the features from a local neighborhood, and is thus less computationally expensive. The features can be aggregated from a different number of hops or search depth. Therefore the model often generalizes better. GraphSAGE is suitable for crop yield prediction because most counties only border a few others and the adjacency matrix is sparse. It also provides flexible aggregation methods.

Formally, for the l-th layer of GraphSAGE,

$$\begin{aligned} \mathbf{a}_{c,t}^{(l)} &= g_l(\{\mathbf{z}_{c',t}^{(l-1)}, \forall c' \in \mathcal{N}(c)\}) \\ \mathbf{z}_{c,t}^{(l)} &= \sigma(\mathbf{W}^{(l)} \cdot (\mathbf{z}_{c,t}^{(l-1)}, \mathbf{a}_{c,t}^{(l)})) \end{aligned} \tag{3}$$

where $\mathbf{z}_{c,t}^{(0)} = h_{c,t}$ from Eq. 1, and $l \in \{0, 1, ..., L\}$. $\mathcal{N}(c) = \{c', \forall A_{c,c'} = 1\}$ is the set of neighboring counties for c. The aggregation function for the l-th layer is denoted $g_l(\cdot)$, which could be mean, pooling, or graph convolution (GCN) function. In practice, we found mean or pooling are effective and computationally efficient. $\mathbf{a}_{c,t}^{(l)}$ is the aggregated embedding from the bordering counties. We concatenate $\mathbf{a}_{c,t}^{(l)}$ with the last layer's embedding $\mathbf{z}_{c,t}^{(l-1)}$ before the transformation using $\mathbf{W}^{(l)}$. $\sigma(\cdot)$ is a non-linear function.

GNN-RNN The output embedding from GNN's last layer $\mathbf{z}_{c,t}^{(L)}$ thus extracts the information (e.g., weather, soil) from the whole local neighborhood for year t. To integrate the historical knowledge, we can do the same as in Eq. 2, by taking the GNN output embeddings from prior years:

$$\widehat{y}_{c,t} = r(\mathbf{z}_{c,t-\Delta t}^{(L)}, ..., \mathbf{z}_{c,t-1}^{(L)}, \mathbf{z}_{c,t}^{(L)}) \tag{4}$$

where $\mathbf{z}_{c,t'}^{(L)}$ is the GNN embedding from year t'.

Loss Function We use log-cosh function as our objective:

$$L(\widehat{y}_{c,t}, y_{c,t}) = \log(\cosh(\widehat{y}_{c,t} - y_{c,t})) \tag{5}$$

Log-cosh works similarly to mean square error, but is not as strongly affected by the occasional wildly incorrect prediction. It is also twice differentiable everywhere. Mini-batch training is adopted during optimization. Batch loss is the average log-cosh loss of all samples in a batch.

Related Work

As one of the early works, (**?**) attempted a shallow neural network on corn yield prediction. It was shown that neural networks could beat conventional regression algorithms (**?**). In recent years, owing to the development of deep learning,

neural network-based models have become more prevalent in the crop yield prediction field (**??**). As we mentioned in the introduction, 48 out of 70 recent works we surveyed employed neural nets. More than half of the NN-based works adopted CNN and one fourth of them employed RNN.

There are two groups of research directions according to different input sources. The first group of methods read remote sensing data such as satellite images or normalized difference vegetation index (NDVI), and used that to estimate the yields. (**?**) applied a deep Gaussian process to predict crop yields from a series of the multi-spectral satellite images. (**?**) employed deep CNNs to reduce the prediction uncertainty on RGB images. (**?**) did a case study in the Midwest and compared various AI models on satellite product datasets. 20 out of all the 70 papers primarily or only dealt with remote sensing data. These methods illuminate the use of the widely available remote sensing data, but it is hard to directly model the relations between crop yields and environmental factors that actually affect the yields. Therefore, another line of research aims at collecting environmental factors and directly training models with these factors as inputs. (**?**) used temperature, rainfall and other meterological parameters to predict wheat production. (**?**) collected crop genotypes and environments to predict the performance of corn hybrids. More recently, CNN-RNN (**?**) incorporated historical environment knowledge and proved benefits. Our GNN-RNN takes one more step by further adding neighborhood information.

Dataset-wise, most papers have their own small-scale datasets, which vary largely in different dimensions. Scale-wise, (**?**) only studied a single zone in Mexico, while (**?**) studied the whole country of Argentina. Time-wise, (**?**) spanned over 5 years, while (**?**) investigated 13 years. It is thus hard to fairly compare models or validate the performance. There have been several efforts to evaluate models on a large and consistent dataset. The closest one to ours is the one from the CNN-RNN paper (**?**). However, they only used 13 states from the Corn Belt and used fewer features than us (for example, they did not use land surface data such as soil moisture). By contrast, we evaluate our models at a nationwide scale, using data from 41 states and 39 years. This forces our models to generalize to a diverse range of locations that have very different climatic and geographic conditions, instead of overfitting to a single region.

Experiments

We compare 11 representative machine learning models, including GNN and GNN-RNN, on US county-level crop yields for corn and soybean. We evaluate performance on three metrics: RMSE, R^2, and correlation coefficient. Given a test year t, we use year $t-1$ for validation and all the prior years for training. For example, if the test year is 2019, we train on data from years 1981-2017 (inclusive), validate on 2018 crop yields, and test on 2019 crop yields.

Dataset Details

Crop yield labels for corn and soybean are available from the USDA Crop Production Reports (**?**) for numerous counties in the US. Not all counties report data in every year, but the coverage is still quite comprehensive. For example, for corn, all years between 1981 and 2003 have over 2,000 counties across 41 states reporting data. We train and evaluate our model on all counties where yield data is available. (When computing the loss, we ignore counties that do not have yield labels for that year.)

We use a variety of climate, land surface, and soil quality variables as input features; these features are available for almost all counties in the contiguous 48 US states (3,107 counties in total[1]). We draw 7 weather features from the PRISM climate mapping system (**?**): precipitation, min/mean/max temperature, min/max vapor pressure deficit, and mean dewpoint temperature. These features are available at a 4×4 km grid for each day.

We acquire 16 land surface features from the North American Land Data Assimilation System (NLDAS) (**?**), which is a large-scale land surface model that closely simulates land surface parameters. These features include soil moisture content, moisture availability, and soil temperature (all at various soil depths), as well as observed weather variables such as wind speed and humidity. These variables are available at a 0.125×0.125 degree (~ 14 km) spatial resolution, every hour.

Soil quality features were acquired from the Gridded Soil Survey Geographic Database (gSSURGO) (**?**), at a 30×30 meter resolution. These features include available water capacity, bulk density, and electrical conductivity, pH, and organic matter. Unlike the weather and land surface features, the gSSURGO soil quality features are fixed and *do not change over time*. In addition to the raw features, we use the raw sand, silt, and clay percentages to compute the "soil texture type" of each pixel based on the Natural Resources Conservation Service Soil Survey's classification scheme (**?**), and then compute the fraction of each county occupied by each soil texture type. In total, we have a total of 20 gSSURGO variables that are depth-dependent (so there are values for 6 different soil depth levels), and 6 "extra" variables which are not depth-dependent (such as crop productivity indices). Finally, as in (**?**), we use the average crop yield (over all counties) of the previous year as an additional input feature, to capture the increasing trend in crop yield over time. A full list of the features can be found in the Appendix.

All of these datasets were originally available as gridded raster data at a variety of spatial resolutions. We aggregated each feature to the county level by computing the weighted average of the variable over all grid cells that overlap with the county. Each grid cell is weighted by the percentage of the cell that lies inside the county, multiplied by the percentage of that grid cell which is cropland, pasture, or grassland; the land cover percentages are computed using the National Land Cover Database (**?**). In addition, the time-dependent

[1]The only exception is Nantucket County, Massachusetts, where land surface model data is missing, since it is an offshore island. Also note that some counties have feature data but not label (yield) data. Only the GNN and GNN-RNN models can make use of these unlabeled county features.

variables (weather and land surface) were aggregated from daily to weekly frequency to make the prediction task more tractable.

Compared Methods

We consider two types of methods: **(a) single-year methods** that only use features from year t to predict yield for the same year t, and **(b) 5-year methods** that use features from a 5-year series (years $\{t-4, t-3, \ldots, t\}$) to predict yield for year t.

Single-year methods. We first consider methods that only use a single year of data to make predictions, to provide a fair comparison to the single-year GNN. For non-deep baseline methods, we select lasso, ridge regressor and gradient boosting regressor. For these methods, we flatten all the features from the entire year into a single feature vector, ignoring the temporal and soil-depth structure in the data. Next, we tried three baseline deep learning architectures for $f_{wl}(\cdot)$: LSTM (**?**), GRU (**?**), and 1-D CNN (**?**). All of these methods process the weekly time-series of weather and land surface data within the year. We compare these methods with our single-year GNN model (Eq. 3), which incorporates geospatial context in making predictions.

5-year methods. For history-dependent models, we follow (**?**) by considering a 5-year dependency for a consistent and fair comparison. Two baseline models using LSTM and GRU respectively handle the raw inputs $\{\mathbf{x}_{c,t-\Delta t}, ..., \mathbf{x}_{c,t}\}$ directly with $r(\cdot)$. Specifically, they flatten the features *for each year* into a single vector (disregarding the weekly structure of the weather data or the depth structure of the soil data), and then feed the 5 year-vectors into the LSTM or GRU. The most recent CNN-RNN model (**?**) pre-processes the raw features with a CNN (choosing CNN for $f_{wl}(\cdot)$) and then uses a LSTM to model the sequence embeddings as described in Eq. 2. Finally, the GNN-RNN model proposed in this paper (Eq. 4) still uses a CNN for $f_{wl}(\cdot)$ to encode the raw features into an embedding for each year, then uses the GNN to refine the embeddings using information from the county's spatial context, and then passes those embeddings into an LSTM.

Evaluation Metrics

We evaluate all methods on three popular metrics for regression: root mean square error (RMSE), the coefficient of determination (R^2), Pearson correlation coefficient (Corr). RMSE and R^2 tell us how well a regression model can predict the value of the response variable in absolute terms and percentage terms respectively. Note that our RMSE figures are expressed in units of the standard deviation of that crop's yield (across all years). Corr is essentially a normalized measurement of the covariance between two sets of data, and captures the strength of the linear correlation between true and predicted values. See Appendix for formal definitions.

Model Details

For the shallow models (ridge regression, lasso, and gradient boosting regressor), we used scikit-learn's implementations.

For the baseline single-year models, we evaluated using LSTM, GRU, and CNN as $f_{wl}(\cdot)$ to process the weekly weather and land surface data. For CNN, we used a 1-D CNN similar to the one in (**?**), but we process all weather and land surface parameters together. The CNN contains series of 1D convolutions, ReLUs, and average pooling layers; this sequence is repeated four times. For all methods that use LSTM or GRU, we used PyTorch's implementation with 64 hidden states.

The same CNN is used as the encoder for the weekly weather and land surface data in the CNN-RNN, GNN, and GNN-RNN models. (We also tried using an LSTM as the encoder for the weekly data for these models, but this did not improve results.) For all methods except for the 5-year LSTM/GRU, we processed the soil data using another small 1-D CNN (with three convolutional layers, and without average pooling), where the convolutions operate across 6 different soil depths.

For the simple 5-year baseline models (LSTM and GRU), we fed the flattened feature vectors for each year through an LSTM or GRU, followed by a 2-layer fully connected network.

For the GNN and GNN-RNN models, we used the implementation of GraphSAGE from the dgl library; we used a 2-layer GNN, with edge dropout of 0.1. The adjacency graph of US counties is provided by the US Census Bureau. We used stochastic mini-batch training to train the model, where each layer samples 10 neighbors to receive messages from. We tried different aggregation functions and found that the "pooling" approach generally performed best.

For all methods, we use the Adam optimizer (**?**), sometimes with a mild cosine or step decay. We tried learning rates between 1e-5 and 1e-3, used a weight decay of 1e-5 or 1e-4, and a batch size of 32, 64, or 128. We trained the model for 100 to 200 epochs (until the validation loss clearly stopped improving). We chose the epoch and hyperparameter setting that produced the lowest RMSE on the validation year (the year before the test year). We ran the GNN and GNN-RNN models 3 times with different random seeds to evaluate the variance in the results. The Appendix contains more details about hyperparameters.

Crop Yield Prediction Results

We evaluate the model on four test datasets: 2018 corn, 2018 soybean, 2019 corn, and 2019 soybean. These datasets span a wide geographic area, as well as differing growing conditions (2019 was a bad year due to the wet spring in the Midwest, which caused planting to be delayed). The results on these datasets are shown in Table 1. For the methods that only use 1 year when making predictions, our GNN model clearly outperforms comparable baselines across all datasets and metrics (except for 2018 soybean Corr, where it is slightly worse than GRU). For the methods that use a history of 5 years, our GNN-RNN outperforms competing baselines in almost all cases (except for 2018 soybean Corr, where it is slightly worse than CNN-RNN). For example, in 2019, our corn yield prediction on R^2 score is 16% better than the prediction of a state-of-the-art work (**?**). On average, we achieve a relative R^2 improvement of 10.44% over the recent CNN-RNN model, 16.16% over the 5-year LSTM, and a relative RMSE improvement of 9.6% over the CNN-

Method	RMSE	R^2	Corr
lasso 1y	0.7846	0.3839	0.7778
ridge 1y	0.9255	0.1428	0.7626
gradient-boosting 1y	0.7402	0.4516	0.7794
gru 1y	0.5938	0.6472	0.8158
lstm 1y	0.6146	0.6220	0.8303
cnn 1y	0.5824	0.6606	0.8235
gnn 1y **(ours)**	**0.4846**	**0.7517**	**0.8759**
(std)	*(0.0097)*	*(0.0100)*	*(0.0019)*
gru 5y	0.6765	0.5419	0.8194
lstm 5y	0.6542	0.5716	0.8060
cnn-rnn 5y	0.5511	0.6936	0.8425
gnn-rnn 5y **(ours)**	**0.4900**	**0.7595**	**0.8731**
(std)	*(0.0191)*	*(0.0186)*	*(0.0092)*

(a) 2018 corn results

Method	RMSE	R^2	Corr
lasso 1y	0.6838	0.3122	0.6715
ridge 1y	0.7081	0.2623	0.6723
gradient-boosting 1y	0.7345	0.2064	0.6857
gru 1y	0.5890	0.4897	0.7381
lstm 1y	0.6245	0.4262	0.7096
cnn 1y	0.5572	0.5432	0.7384
gnn 1y **(ours)**	**0.4930**	**0.6286**	**0.8011**
(std)	*(0.0068)*	*(0.0102)*	*(0.0037)*
gru 5y	0.5279	0.5900	0.7785
lstm 5y	0.5311	0.5849	0.7821
cnn-rnn 5y	0.5212	0.5842	0.7868
gnn-rnn 5y **(ours)**	**0.4677**	**0.6782**	**0.8272**
(std)	*(0.0035)*	*(0.0049)*	*(0.0038)*

(b) 2019 corn results

Method	RMSE	R^2	Corr
lasso 1y	0.6226	0.6090	0.7912
ridge 1y	0.7633	0.4125	0.7550
gradient-boosting 1y	0.6686	0.5492	0.7986
gru 1y	0.6376	0.5932	**0.8356**
lstm 1y	0.6459	0.5825	0.8129
cnn 1y	0.6584	0.5661	0.7988
gnn 1y **(ours)**	**0.5637**	**0.6794**	0.8273
(std)	*(0.0144)*	*(0.0163)*	*(0.0095)*
gru 5y	0.6094	0.6254	0.8218
lstm 5y	0.5430	0.7026	0.8459
cnn-rnn 5y	0.5647	0.6784	**0.8650**
gnn-rnn 5y **(ours)**	**0.5333**	**0.7129**	0.8591
(std)	*(0.0194)*	*(0.0206)*	*(0.0049)*

(c) 2018 soybean results

Method	RMSE	R^2	Corr
lasso 1y	0.5731	0.6137	0.8089
ridge 1y	0.6069	0.5668	0.7944
gradient-boosting 1y	0.6802	0.4558	0.7899
gru 1y	0.5742	0.5150	0.7569
lstm 1y	0.5907	0.4867	0.7195
cnn 1y	0.5699	0.5222	0.7385
gnn 1y **(ours)**	**0.4916**	**0.7148**	**0.8505**
(std)	*(0.0335)*	*(0.0395)*	*(0.0165)*
gru 5y	0.5751	0.6109	0.8158
lstm 5y	0.5512	0.6427	0.8156
cnn-rnn 5y	0.5365	0.6615	0.8423
gnn-rnn 5y **(ours)**	**0.4745**	**0.7349**	**0.8602**
(std)	*(0.0160)*	*(0.0179)*	*(0.0076)*

(d) 2019 soybean results

Table 1: Evaluation results. For RMSE, lower is better; for R^2 and Corr, higher is better. We grouped the methods based on whether they use 1 year of data (1y) or 5 years of data (5y) to make predictions.

Method	RMSE	R^2	Corr
lstm 1y	0.6347	0.5968	**0.8148**
cnn 1y	0.7253	0.4736	0.7004
gnn 1y **(ours)**	**0.5877**	**0.6543**	0.8124
lstm 5y	0.7004	0.5091	0.7708
cnn-rnn 5y	0.6532	0.5730	0.7732
gnn-rnn 5y **(ours)**	**0.5836**	**0.6591**	**0.8259**

Table 2: Early prediction results (2018 corn, after June 1).

RNN model, 13.18% over the 5-year LSTM. These indicate the importance of exploiting geospatial context in making these predictions.

Figure 2 shows an example scatterplot of true vs. predicted corn yields for the test year 2018. The GNN-RNN model is able to capture differences in yield between counties quite well. One minor issue is that the model is not able to capture the counties with very high yields very well; the model almost never predicts a yield higher than 220, but there are actually several counties with a true yield higher than this. This may stem from the fact that such high yields were almost never seen before in the training years.

We can also see these trends in the map (Figure 3). While the GNN-RNN model captures large-scale trends in crop yield very well, it sometimes outputs overly smooth predictions within a region, and under-predicts the area of high true yield in the Midwest. Improving the GNN's ability to detect fine-scale variations without smoothing them out is an important area for future work.

We can see that crop yield prediction on a large scale is rather challenging, due to the complexity of the prediction task and the data involved. Each data point (one county/one year) has over 6,000 features, and standard models can easily overfit to noise in the data and fail to generalize. In order for the prediction task to be tractable, a model needs to take advantage of the various forms of structure in the data; temporal structure within a year (to capture weather patterns in different times of the year), temporal structure across years (to capture long-term trends such as technological improvements), and geospatial structure (to capture correlations between nearby county yields). Our GNN-RNN model is the first model to take all of these aspects into account when making crop yield predictions, and achieves superior perfor-

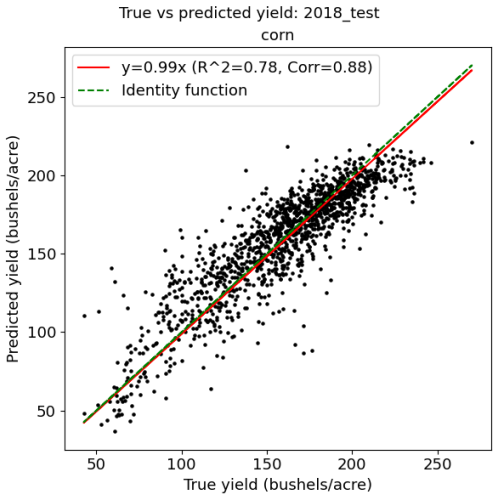

Figure 2: Predicted vs. ground truth corn yields in 2018

mance compared with the existing state-of-the-art.

Early Prediction

In practice, crop yield predictions are most useful if they can be made well before harvest, as this gives time for markets to adapt, and humanitarian aid to be organized in cases of famine (**?**). To simulate this, at test time only, for each county we mask out all weather and land surface features from after June 1 (week 22) of the test year, and replace them with the average values for that county during the training years. Then we pass the masked features through a pre-trained model to obtain predictions. The results for several methods for 2018 corn are presented in Table 2. The graph-based models (GNN and GNN-RNN) clearly outperform competing baselines in this scenario, again illustrating the importance of utilizing geospatial context.

Conclusion

In this paper, we propose a novel GNN-RNN framework to innovatively incorporate both geospatial and temporal knowledge into crop yield prediction, through graph-based deep learning methods. To our knowledge, our paper is the first to take advantage of the spatial structure in the data when making crop yield predictions, as opposed to previous approaches which assume that neighboring counties are independent samples. We conduct extensive experiments on large-scale datasets covering 41 US states and 39 years, and show that our approach substantially outperforms many existing state-of-the-art machine learning methods across multiple datasets. Thus, we demonstrate that incorporating knowledge about a county's geospatial neighborhood and recent history can significantly enhance the prediction accuracy of deep learning methods for crop yield prediction.

Acknowledgements

This research was supported by USDA Cooperative Agreement 58-6000-9-0041 and USDA NIFA Hatch Project 1017421. We would like to thank Rich Bernstein for constructive suggestions and Samuel Porter for help in processing the gSSURGO dataset.

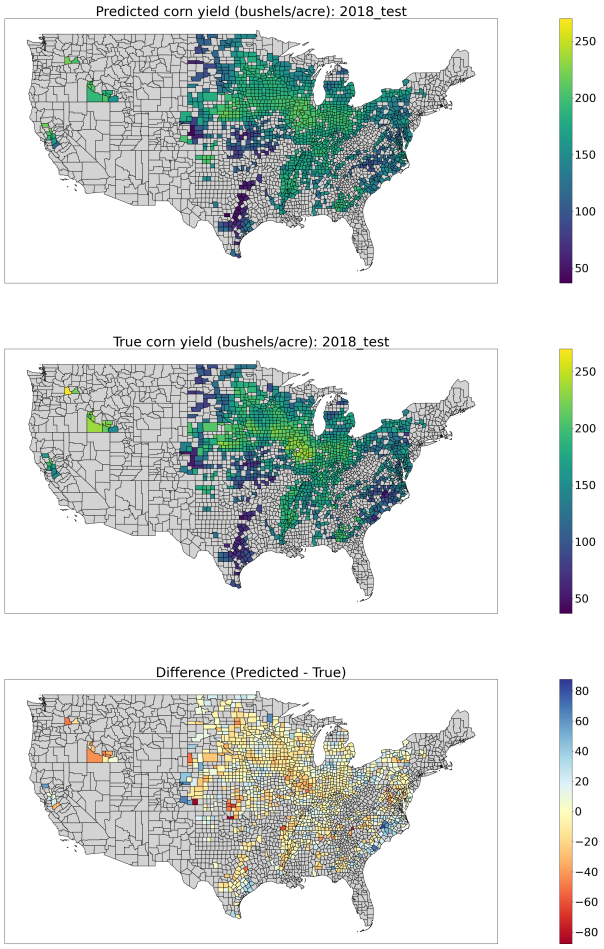

Figure 3: Maps of predicted (top) and true (middle) corn yields in 2018, along with the difference (bottom). For the Difference plot, yellow means an accurate prediction, blue means the model predicted too high, and red means the model predicted too low. Gray means no data.

Esse maiores eos, placeat porro fuga beatae id neque odit minus nulla a, dolores fugit esse temporibus nemo omnis iste autem incidunt deleniti veritatis consequatur.Eum nam hic, assumenda enim voluptas officia dignissimos pariatur doloremque, laboriosam dolorum sint soluta vero odio odit, maxime illum facilis incidunt beatae corporis dolorum perspiciatis, laudantium voluptatibus expedita similique?Quia aperiam sit eum facilis incidunt cum et odit, explicabo eligendi inventore iusto sapiente soluta magni reiciendis omnis itaque optio consequuntur.Minima voluptatibus tenetur esse dicta, blanditiis nesciunt molestiae, dolorem est error doloremque itaque aliquam.Ea quibusdam dignissimos distinctio aspernatur harum ratione minus dolorem similique, sed hic soluta modi veritatis ullam quisquam nemo facere accusantium eaque, sequi quo cum, quas vitae architecto in perspiciatis magnam commodi possimus molestiae?Nam as-

pernatur vitae sapiente culpa modi ut velit assumenda odit alias unde, laudantium possimus ipsa suscipit et dolore dolorem quae dignissimos pariatur officia, a