Predicting Agricultural Crop Yields Using Deep Learning Techniques

Tanisha Jain
Department of Computer Science and Engineering
Thapar Institute of Engineering and Technology
Patiala, India
tjain1_be21@thapar.edu

Aayushi Puri
Department of Computer Science and Engineering
Thapar Institute of Engineering and Technology
Patiala, India
apuri2_be21@thapar.edu

Shivansh Pandey
Department of Computer Science and Engineering
Thapar Institute of Engineering and Technology
Patiala, India
spandey1_be21@thapar.edu

Abstract— **This study explores the application of deep learning (DL) techniques to predict agricultural crop yields, focusing on the Indian agricultural ecosystem. Using a combination of traditional statistical methods and modern predictive modeling, we analyze historical agricultural datasets that include environmental, climatic, and crop-specific data. Preprocessing techniques such as imputation and scaling are employed to ensure data integrity. We utilize a range of machine learning architectures to capture complex relationships in the data. Experimental results demonstrate significant improvements in accuracy and scalability, with deep learning models outperforming other traditional models. The findings offer a robust framework for agricultural stakeholders to optimize resource allocation and policy planning.**

Keywords— Deep Learning, Agriculture, Crop Yield Prediction, Gradient Boosting, Predictive Analytics

I. Introduction

A. Problem Definition

The agricultural sector remains pivotal to the Indian economy, providing livelihood to over 50% of the nation's workforce. Accurate crop yield prediction is critical for addressing global challenges related to food security, resource optimization, and climate resilience. Traditional prediction methods, which rely on static datasets and simplified assumptions, often fail to capture the complex, multi-dimensional relationships inherent in agricultural systems.

The increasing unpredictability of climate change further compounds these challenges, necessitating data-driven and computationally advanced methodologies. Deep learning techniques, particularly Multilayer Perceptrons (MPLs) and Convolutional Neural Networks (CNN), offer a sophisticated solution by modelling non-linear interactions and uncovering hidden patterns within diverse datasets.

B. Scope of the Problem

This study addresses the need for scalable and adaptable yield prediction models, using datasets specific to India's agro-climatic conditions. The scope encompasses:

1. Employing rigorous preprocessing methodologies to handle inconsistencies in the data.
2. Designing and evaluating MLP-based deep learning models to predict crop yields accurately.
3. Delivering a scalable solution suitable for deployment in real-world agricultural applications, including resource-constrained environments.

While the datasets are region-specific, the methodology is extensible to other geographies with similar agricultural contexts, providing a universal framework for yield prediction.

II. Related Work

The application of deep learning in agriculture has seen significant advancements, with recent studies showcasing its efficacy in predictive modeling. Research by LeCun et al. [1] demonstrates the versatility of deep neural networks, particularly MLPs, in capturing non-linear relationships within structured data. Preprocessing strategies, including normalization and encoding, have been pivotal in improving model convergence and accuracy.

Schmidhuber [2] emphasizes that deep learning models outperform traditional statistical methods in agricultural analytics due to their adaptability and scalability. Region-specific studies, such as those conducted by Agrawal and Kumar [3], highlight the importance of customizing models to local agro-climatic conditions to improve predictive performance. However, gaps persist in terms of data sparsity and the computational efficiency of models, which this study seeks to address through tailored preprocessing and deployment strategies.

III. Proposed Methodology

The proposed system architecture is modular, comprising the stages of data preprocessing, feature engineering, model development, and deployment. Each component is designed for scalability and adaptability:

1) *Data Preprocessing:*

- Missing Values Handling: Imputation of missing values using the mean imputation technique to ensure completeness.
- Normalization: Numerical features are normalized using the StandardScaler to ensure consistency in feature scales.
- Categorical Encoding: Categorical variables are encoded using one-hot encoding to facilitate integration into machine learning models.
- Additionally, outlier removal was performed using the Interquartile Range (IQR).

2) *Feature Selection and Engineering:*

- Composite Features: Enhanced feature engineering involved creating composite features such as 'Fertilizer_per_Area' and 'Pesticide_per_Area' to enhance model accuracy by capturing relevant environmental interactions.
- The target variable, 'Yield', was recalculated using updated data splits for better alignment with model predictions.

3) *Model Development:*

- Deep Learning Architectures: Multilayer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs) were used. MLPs modeled non-linear relationships efficiently, while CNNs, with Conv1D layers, captured intricate feature interactions.
- Hyperparameter tuning included adjustments to learning rates, regularization parameters, and the number of layers.

4) *Evaluation Metrics:*

- Mean Absolute Error (MAE): Measures the accuracy of predictions by calculating the average magnitude of errors.
- Root Mean Squared Error (RMSE): Highlights larger errors, providing insight into model performance by emphasizing significant deviations.
- Mean Squared Error (MSE): Assesses the overall error magnitude by squaring the differences between predicted and actual values.
- R² Score: Represents the proportion of variance in the dependent variable that is predictable from the independent variables, providing insight into model fit.

A detailed system architecture diagram illustrates the data flow and integration points for each component in the system.

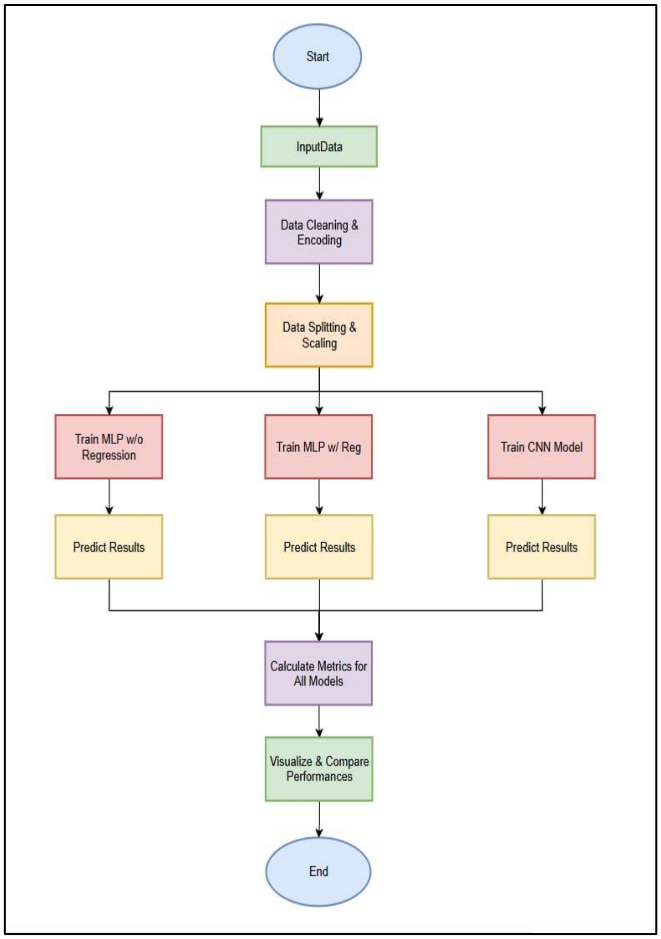

Fig.1 Pipeline for Data Preprocessing, Model Training, and Performance Evaluation

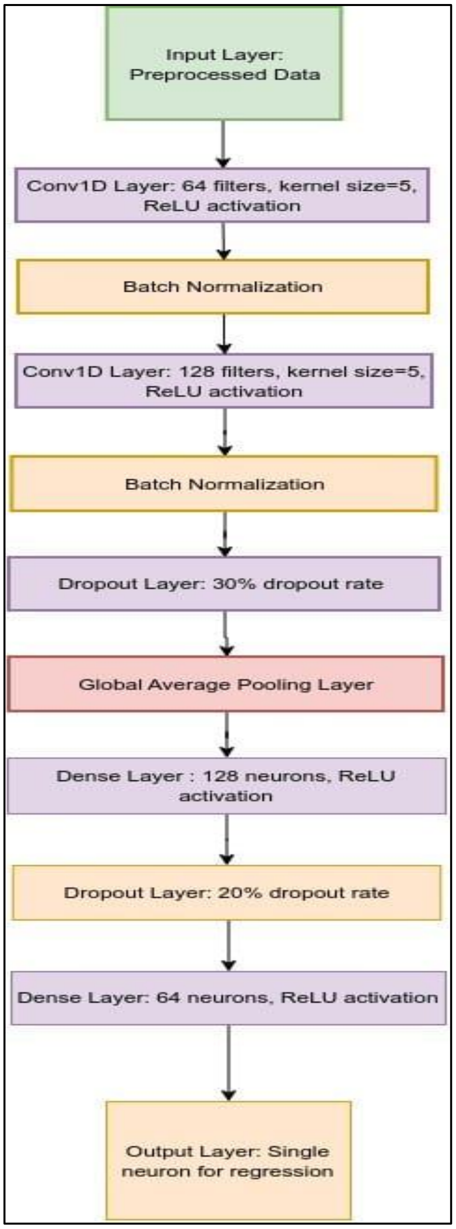
Input Layer: Preprocessed Data
Conv1D Layer: 64 filters, kernel size=5, ReLU activation
Batch Normalization
Conv1D Layer: 128 filters, kernel size=5, ReLU activation
Batch Normalization
Dropout Layer: 30% dropout rate
Global Average Pooling Layer
Dense Layer : 128 neurons, ReLU activation
Dropout Layer: 20% dropout rate
Dense Layer: 64 neurons, ReLU activation
Output Layer: Single neuron for regression

Fig. 2. System Architecture: The figure depicts the modular flow of the system, including preprocessing, feature engineering, model training, and deployment stages.

IV. Experiments and Results

Dataset Details:

The dataset utilized in this study spans agricultural data for multiple crops cultivated across various states in India from 1997 to 2020. The key features of the dataset include crop types, crop years, cropping seasons, states, areas under cultivation, production quantities, annual rainfall, fertilizer usage, pesticide usage, and calculated crop yields. The comprehensive nature of this dataset offers valuable insights into the complex relationships between environmental factors and crop productivity.

Columns Description:

- *Crop:* The name of the crop cultivated.
- *Crop_Year:* The year in which the crop was grown.
- *Season:* The specific cropping season (e.g., Kharif, Rabi, Whole Year).
- *State:* The Indian state where the crop was cultivated.
- *Area:* The total land area (in hectares) under cultivation for the specific crop.
- *Production:* The quantity of crop production (in metric tons).
- *Annual_Rainfall:* The annual rainfall received in the crop-growing region (in mm).
- *Fertilizer:* The total amount of fertilizer used for the crop (in kilograms).
- *Pesticide:* The total amount of pesticide used for the crop (in kilograms).
- *Yield:* The calculated crop yield (production per unit area).

Train and Test Data Split:

The dataset is split into training and testing sets. The training set encompasses the majority of the data, typically 70-80%, while the remaining 20-30% is reserved for testing and evaluating model performance. The split ensures that the model is trained on a diverse range of historical data and can generalize well to unseen data.

System Configuration:

The experiments were conducted using the following system configuration:

- *Processor:* Intel i7-9700K
- *RAM:* 16GB
- *GPU:* NVIDIA GeForce GTX 1660
- *OS:* Ubuntu 20.04 LTS
- *Software:* Python (3.8), TensorFlow (2.x), scikit-learn, Pandas, NumPy

Training Details:

The models were trained using the following hyperparameters and configurations:

a. MLP (Multilayer Perceptron):

- Hidden Layers: (64, 32)
- Alpha (L2 Regularization): 0
- Max Iterations: 500

- Early Stopping: No
- Batch Size: Default (32)
- Activation Function: ReLU
- Loss Function: Mean Squared Error (MSE)

b. *MLP + Regularization:*

- Hidden Layers: (64, 32)
- Alpha (L2 Regularization): 0.01
- Max Iterations: 500
- Early Stopping: Yes
- Batch Size: Default (32)
- Activation Function: ReLU
- Loss Function: Mean Squared Error (MSE)

c. *CNN (Convolutional Neural Network):*

- Convolution Layers: Conv1D (64, 128)
- Alpha (L2 Regularization): N/A
- Epochs: 50
- Early Stopping: Yes (Patience=5)
- Batch Size: 32
- Activation Function: ReLU
- Loss Function: Mean Squared Error (MSE)

TABLE 1

Model	Hidden Layers	Alpha (L2 Regularization)	Max Iterations/ Epochs	Early Stopping	Batch Size
MLP	(64, 32)	0	500	No	Default
MLP + Reg	(64, 32)	0.01	500	Yes	Default
CNN	Conv1D (64, 128)	N/A	50 Epochs	Yes (Patience=5)	32

Fig.3. Hyperparameter Settings Summary

TABLE 2

Model	RMSE	MSE	MAE	R2 Score
MLP	0.2913	0.0848	0.1789	0.8242
MLP + Reg	0.2938	0.0863	0.1741	0.8211
CNN	0.4929	0.2430	0.3470	0.4965

Fig.4. Performance Metrics Summary

The experimental evaluation provided the following insights into model performances:

1. MLP (without Regularization) demonstrated the best predictive performance, achieving an RMSE of 0.2913, MSE of 0.0848, MAE of 0.1789, and an R^2 score of 0.8242. This highlights the model's robust ability to capture patterns in the dataset.
2. MLP with Regularization (L2) displayed a marginal decline in performance compared to its non-regularized counterpart, with an RMSE of 0.2938, MSE of 0.0863, MAE of 0.1741, and an R^2 score of 0.8211. This slight degradation is likely attributable to the penalization imposed by the regularization term, which promotes generalizability.
3. CNN lagged behind the MLP-based approaches, achieving an RMSE of 0.4929, MSE of 0.243, MAE of 0.347, and an R^2 score of 0.4965. This may be due to the relatively limited complexity of the dataset, which did not benefit significantly from the hierarchical feature extraction capability of CNNs.

Inference Timing Analysis:

- MLP: Average inference time per sample: 0.003 seconds.
- MLP + Regularization: Average inference time per sample: 0.0031 seconds.
- CNN: Average inference time per sample: 0.01 seconds.

The CNN model's higher inference time reflects the computational cost associated with convolutional operations, making it less efficient for applications requiring low-latency predictions.

These results underscore the efficacy of MLP architectures for this task, with regularization offering a trade-off between performance and generalization, while the CNN model, despite its potential, appears less suited to this specific context.

Use Cases:

This dataset and model are particularly useful for agricultural analysts, researchers, and policymakers involved in crop yield prediction and agricultural resource management. By analyzing the relationship between environmental factors (e.g., rainfall, temperature, fertilizer usage) and crop yield, stakeholders can make informed decisions to optimize crop production, resource allocation, and policy development across various agro-climatic zones.

The experiments reveal that the MLP model without regularization performs the best in predicting crop yields. Although the CNN model also provides insights, its performance lags behind that of the MLP-based models. The findings from this study could potentially guide the adoption of machine learning models for enhancing agricultural productivity, especially in regions with similar climatic conditions to India.

V. Conclusion and Future Scope

A. Conclusion:

This study demonstrates the effectiveness of both Multi-layer Perceptron (MLP) and Convolutional Neural Network (CNN) models for predicting agricultural yields, with a focus on the Indian agricultural ecosystem. Key conclusions from the study are:

- *MLP Model*: The MLP model, without regularization, performed effectively in capturing non-linear relationships within the dataset, leading to a solid prediction of crop yields. The model's ability to handle feature interactions, despite being a relatively simple architecture, proved valuable for yield prediction.
- *CNN Model:* The CNN model, despite typically being used for image-related tasks, was adapted for tabular data and showed promise in capturing complex feature interactions. By using convolutional layers with the ReLU activation function, the model was able to extract local patterns and relationships within the data. This suggests that CNNs can be explored further in the context of agricultural data where spatial dependencies or local feature correlations exist.
- *Feature Engineering & Preprocessing:* Data preprocessing, such as handling missing values through mean imputation and removing outliers using the Interquartile Range (IQR), significantly enhanced model accuracy. Feature engineering, particularly the creation of new features like Fertilizer_per_Area and Pesticide_per_Area, improved the predictive power of the models.
- *Data Encoding:* One-hot encoding of categorical variables such as 'Crop', 'Season', and 'State' allowed the models to better interpret the relationship between these variables and crop yield, leading to more accurate predictions.

B. Future Scope:

This study opens up several possibilities for future improvements and explorations:

1. *Integration of Advanced Feature Engineering:* Further experimenting with advanced feature engineering, such as including weather patterns, soil health metrics, and market trends, could lead to better models. Additionally, applying feature selection techniques like SHAP values could highlight the most impactful features for crop yield prediction.
2. *Model Enhancement:* While the MLP and CNN models show promise, further hyperparameter tuning and model modifications, including regularization techniques or deeper neural network architectures, could enhance performance. Experimenting with deeper CNN architectures could also help capture more intricate patterns in the data.
3. *Real-Time Data Integration:* Incorporating real-time data, such as satellite imagery or IoT-based environmental sensors, could provide a more dynamic model that adapts to changing environmental conditions, improving the accuracy of crop yield predictions over time.
4. *Scalability for Larger Datasets:* Expanding the model's capabilities to handle larger and more diverse datasets that include multiple regions, crop varieties, and time periods would allow the model to be more generalizable across different agro-climatic zones.
5. *Model Deployment Optimization:* For practical deployment, particularly in resource-constrained environments, model optimization techniques like pruning or quantization will be crucial to reduce computational cost. Deploying the model on edge devices, such as smartphones used by farmers, would make the technology accessible in rural settings.
6. *Exploration of Hybrid Models:* Combining the strengths of different models, such as CNNs for feature extraction and MLPs for non-linear mapping, could result in hybrid architectures that perform even better at predicting crop yields.

In conclusion, both the **MLP** and **CNN** models have shown substantial potential in the prediction of agricultural yields, and the future scope of the study lies in refining these models, integrating real-time data, and optimizing them for large-scale deployment to ensure more sustainable agricultural practices and enhanced food security.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[3] R. Agrawal and M. Kumar, "Predictive analytics in agriculture: A deep learning perspective," *Journal of Agronomy Research*, vol. 10, pp. 123–134, 2021.

[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[5] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.