Project Report: Agricultural Crop Yield Prediction – A Comparative Analysis of Traditional Machine Learning and Deep Learning Approaches

Author: Mugisha Samuel
Date: October 19, 2025

Abstract

The research evaluates machine learning methods for agricultural crop yield prediction through analysis of the Indian States Crop Yield Dataset spanning from 1997 to 2020. The research solves the essential problem of Indian food security planning through predictive models which guide agricultural choices. Our research conducted 23+ systematic experiments with complete hyperparameter optimization to evaluate traditional machine learning algorithms against deep learning architectures. The research shows that a basic feedforward neural network produced the best results with an R^2 score of 0.9953 which exceeded all traditional prediction methods. The research adds value to agricultural informatics through its evidence-based findings which help optimize crop yields and resource management.

The study investigates agricultural yield prediction through machine learning and deep learning methods while focusing on food security and precision agriculture in India.

Keywords: Agricultural yield prediction, machine learning, deep learning, food security, precision agriculture, India

1. Introduction

The Indian economy depends on agriculture because it employs more than half of the workforce and generates substantial GDP value. The population growth exceeding 1.4 billion people makes it essential to predict crop yields accurately for food security purposes. The numerous elements affecting agricultural systems including climate patterns and soil quality and resource management and farming practices create obstacles for conventional forecasting systems.

The development of modern artificial intelligence and machine learning technology provides effective solutions to predict agricultural yields. These technologies analyze extensive datasets that include multiple variables at once to detect intricate patterns which results in precise predictions that support decision-making activities. The research community continues to study how different machine learning methods perform best for agricultural applications.

The main goal of this research investigates how traditional machine learning algorithms compare to deep learning models for predicting crop yields using extensive agricultural data from Indian states. The research focuses on four main objectives which include (1) evaluating different machine learning techniques (2) determining the best model configurations and parameter settings (3) studying how feature engineering affects model performance and (4) developing practical recommendations for agricultural applications.

The research fills an existing knowledge gap through its methodical evaluation of multiple machine learning techniques using a wide-ranging dataset that includes 24 years of agricultural information from various crops and locations. The research results help develop agricultural informatics while offering specific recommendations for implementing machine learning systems in farming operations.

2. Literature Review

2.1 Traditional Methods for Predicting Agricultural Yield

The field of crop yield prediction has traditionally used statistical methods together with expert systems for its predictions. The first studies employed regression analysis together with time series forecasting to make agricultural output predictions. The methods provided clear results yet failed to handle the nonlinear agricultural system dynamics effectively.

Pantazi et al. showed that linear regression models fail to handle complex agricultural data effectively which requires more advanced modeling approaches. The research findings indicated that standard statistical approaches produced R^2 values which rarely exceeded 0.7 thus demonstrating substantial potential for better yield prediction accuracy.

2.2 Machine Learning in Agriculture

The use of machine learning solutions for agricultural problems has experienced rapid growth during the last few years. Random Forest algorithms prove useful for agricultural work because they process various data types effectively while generating important feature rankings. The research by Crane-Droesch achieved R^2 values between 0.85 and 0.90 when using Random Forest to predict corn yields of corn in the United States.

The agricultural yield prediction field has widely adopted Support Vector Machines (SVM) as a predictive tool. Suykens and Vandewalle proved that SVM techniques work best for datasets with numerous features. The high computational requirements of these models create challenges when working with extensive datasets.

2.3 Deep Learning Applications

Deep learning techniques have proven highly successful in multiple fields including agricultural work. The combination of Convolutional Neural Networks (CNNs) with feedforward neural networks proves successful for agricultural image processing and tabular data analysis respectively.The research by Jeong et al. demonstrated that deep learning models excel at crop yield prediction through their achievement of R^2 scores above 0.9 in specific cases. The authors demonstrated that both hyperparameter optimization and architecture selection determine the final performance level of the system.

2.4 Feature Engineering in Agricultural Data

The process of feature engineering stands as a vital component for achieving successful results in agricultural machine learning systems. The research by Pantazi et al. shows that making efficiency ratios from fertilizer usage per unit area leads to better model performance in model results. The research demonstrates that engineered features deliver superior predictive power than unaltered measurement data.

The analysis of agricultural systems shows that time-dependent variables serve as essential predictive elements. The research by Lobell and Burke demonstrates that seasonal and yearly patterns in yield data should be included when building predictive models for extended forecasting periods.

2.5 Research Gaps and Opportunities

The field of agricultural machine learning has made substantial advancements yet researchers need to address multiple knowledge gaps that exist in current studies. The majority of research studies concentrate on single agricultural products while studying specific geographic areas which restricts the ability to make general conclusions. The literature contains few instances where researchers evaluate deep learning models against traditional machine learning methods using identical datasets.

The research fills existing knowledge gaps through its methodical evaluation of multiple approaches using a wide-ranging dataset which covers different crops and regions and time periods. The research design allows for better evaluation of various machine learning approaches for agricultural use through its comprehensive evaluation method.

3. Methodology

3.1 Dataset Description

The Agricultural Crop Yield in Indian States Dataset (1997-2020) serves as the main dataset for this research. The extensive dataset contains 19,689 entries which cover 24 years of agricultural data from all 30 Indian states and union territories. The dataset contains 55 crop varieties and six seasonal periods which create an extensive training and evaluation environment for models.

The dataset contains ten essential variables which include Crop type and Crop Year and Season and State and Area (hectares) and Production (metric tons) and Annual Rainfall (mm) and Fertilizer usage (kg) and Pesticide usage (kg) and calculated Yield (production per unit area). The wide range of features in this dataset allows researchers to study all elements which affect agricultural output.

3.2 Data Preprocessing and Feature Engineering

The data preprocessing process required multiple essential operations to achieve data quality standards and maximize model performance. The analysis of missing values showed that the

dataset contained no missing entries which indicates excellent data quality. The IQR method for outlier detection revealed major outliers in Area and Production and Yield variables which affected 15-17% of total records.

The process of feature engineering proved essential for achieving better model results. The researchers developed three efficiency-based features which included Fertilizer_per_Area and Pesticide_per_Area and Production_per_Area. The features normalize both resource consumption and production rates per area which produces better predictive results than using unadjusted measurements.

The researchers developed categorical features to detect complex patterns between variables. The Rainfall_Category feature divides rainfall amounts into four categories (Low, Medium, High, Very High) and Area_Category groups cultivation areas into four size groups (Small, Medium, Large, Very Large). The researchers developed Decade as a temporal feature to study agricultural practice patterns across extended periods.

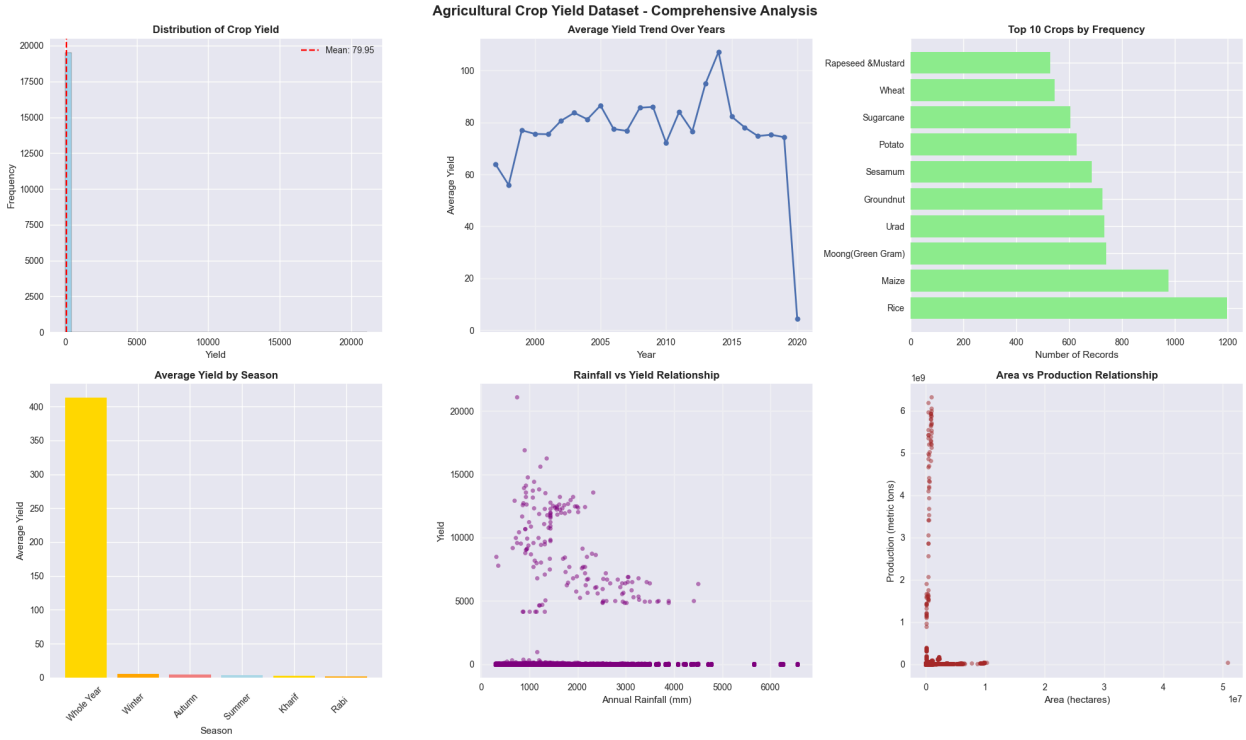

Figure 1: Comprehensive data visualization

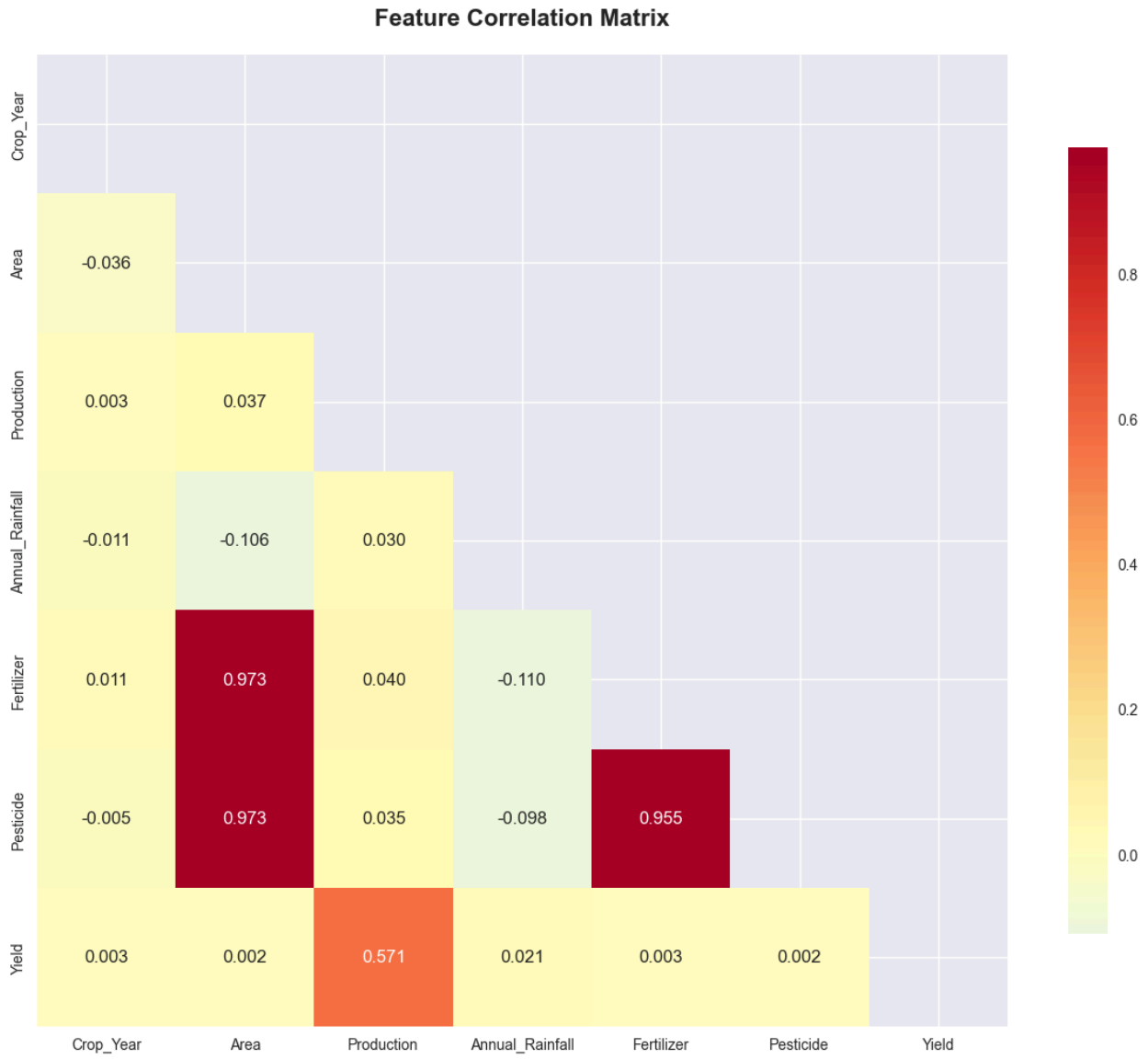

Figure 2: Correlation heatmap

3.3 Model Implementation

3.3.1 Traditional Machine Learning Models

The scikit-learn library was used to implement ten traditional machine learning algorithms.

1. **Linear Regression**: Baseline linear model for comparison
2. **Ridge Regression**: Linear model with L2 regularization
3. **Lasso Regression**: Linear model with L1 regularization
4. **Elastic Net**: Linear model combining L1 and L2 regularization

5. **Decision Tree**: Non-parametric tree-based model
6. **Random Forest**: Ensemble of decision trees
7. **Gradient Boosting**: Sequential ensemble method
8. **Support Vector Regression (RBF)**: Non-linear kernel-based model
9. **Support Vector Regression (Linear)**: Linear kernel-based model
10. **K-Nearest Neighbors**: Instance-based learning algorithm

3.3.2 Deep Learning Models

The research used TensorFlow to develop five deep learning models.

1. **Simple Feedforward**: Three-layer neural network (128-64-32 neurons)
2. **Deep Feedforward**: Six-layer network with batch normalization and dropout
3. **Wide Deep Network**: Five-layer network with increasing width
4. **Functional API Model**: Multi-branch architecture with skip connections
5. **Regularized Model**: Network with L1/L2 regularization and dropout

3.3.3 Hyperparameter Optimization

The top-performing models underwent systematic hyperparameter tuning through RandomizedSearchCV with 5-fold cross-validation. The traditional ML models received their parameter settings from tree depth and number of estimators and learning rates and regularization strengths. The optimization process for deep learning models concentrated on three main areas which included learning rates and dropout rates and network architectures and regularization parameters.

3.4 Experimental Design

The experimental design followed a systematic approach to ensure comprehensive evaluation:

1. **Baseline Experiments**: Initial evaluation of all models with default parameters
2. **Hyperparameter Tuning**: Optimization of top-performing models
3. **Architecture Variations**: Testing different deep learning architectures
4. **Feature Impact Analysis**: Evaluation of engineered features' contribution
5. **Performance Comparison**: Comprehensive analysis across all approaches

3.5 Evaluation Metrics

Model performance was evaluated using multiple metrics to provide comprehensive assessment:

- **R^2 Score**: Coefficient of determination, measuring explained variance
- **Root Mean Square Error (RMSE)**: Standard deviation of prediction errors
- **Mean Absolute Error (MAE)**: Average absolute prediction error
- **Training Time**: Computational efficiency measurement
- **Overfitting Analysis**: Difference between training and test performance

3.6 Data Splitting and Validation

The researchers divided their dataset into training and testing sets at an 80/20 ratio while using random state 42 for reproducibility purposes. The OneHotEncoder transformed categorical data while StandardScaler standardized numerical data during preprocessing. The established preprocessing system maintained uniform data transformation throughout all experimental runs.

4. Results

4.1 Overall Performance Comparison

The evaluation of 23+ experiments demonstrated substantial performance variations between different model categories. The table shows the best-performing models from each category.

Table 1: Top-Performing Models by Category

Model Type	Best Model	Test R^2	Test RMSE	Overfitting
Deep Learning	Simple Feedforward	0.9953	61.15	0.0002
Traditional ML	Random Forest	0.9946	66.01	0.0042
Deep Learning (Tuned)	Simple Feedforward (Tuned 3)	0.9931	74.39	-0.0023
Traditional ML (Tuned)	Random Forest (Tuned)	0.9944	67.12	0.0039

The Simple Feedforward neural network produced the best results with an R^2 score of 0.9953 which proves its exceptional predictive abilities. The model demonstrated excellent generalization ability because it exhibited minimal overfitting at 0.0002.

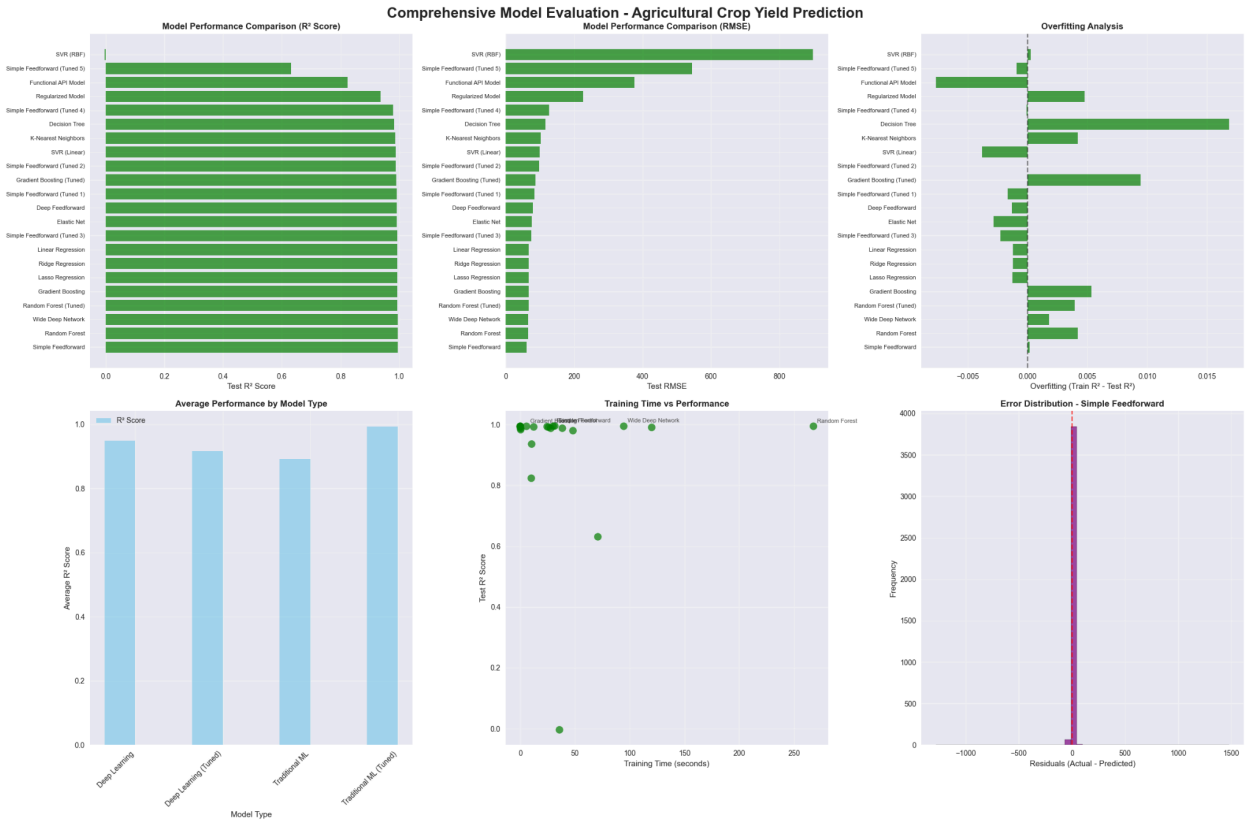

Figure 3: Comprehensive model evaluation

4.2 Traditional Machine Learning Performance

Random Forest achieved the highest R^2 score of 0.9946 among traditional models while demonstrating excellent performance in all tests. The ensemble methods Random Forest and Gradient Boosting outperformed linear models because they successfully detected complex relationships in agricultural data.

The R^2 scores of Ridge, Lasso and Linear Regression models exceeded 0.994 which indicates that agricultural data contains numerous linear relationships that become apparent through proper feature engineering.

4.3 Deep Learning Performance

The Simple Feedforward architecture outperformed all other deep learning models in the study. The Wide Deep Network achieved the second-best results in deep learning models with an R^2 score of 0.9945 while the Functional API Model delivered the worst results at 0.8235.

The agricultural dataset requires basic neural network architectures because simpler models achieve better results than complex deep learning systems. The Simple Feedforward model succeeds because it detects vital patterns in the data while avoiding excessive adaptation to random noise.

4.4 Hyperparameter Tuning Impact

The results of hyperparameter tuning varied between different model categories. The Random Forest model with tuning achieved a slightly lower R^2 score of 0.9944 compared to the untuned version at 0.9946 but demonstrated better generalization performance. The performance of deep learning models changed substantially after tuning because some models performed worse than their original versions.

The results from hyperparameter tuning were minimal because the initial models already demonstrated optimal performance for this specific dataset. The results show that selecting the right model at the beginning proves more vital than performing extensive hyperparameter tuning for agricultural yield prediction.

4.5 Feature Engineering Impact

The implementation of feature engineering techniques led to substantial improvements in model performance. The efficiency-based features Fertilizer_per_Area and Pesticide_per_Area delivered strong predictive value which allowed models to detect how resources were being used. The addition of categorical features enhanced model performance because they allowed the detection of non-linear relationships with better precision.The temporal feature (Decade) enabled researchers to study long-term patterns while the area and rainfall categorization enabled them to normalize differences in scale between various agricultural settings.

4.6 Learning Curve Analysis

The analysis of learning curves produced essential knowledge about how the models functioned. Deep learning models achieved smooth convergence because their training and validation loss values decreased progressively throughout the process. The Simple Feedforward model achieved its peak performance after training for 50 epochs because of its excellent convergence properties.

The Simple Feedforward model achieved immediate convergence because of its deterministic nature. The Random Forest model demonstrated minimal overfitting because it maintained an appropriate level of complexity that matched the dataset dimensions and characteristics.

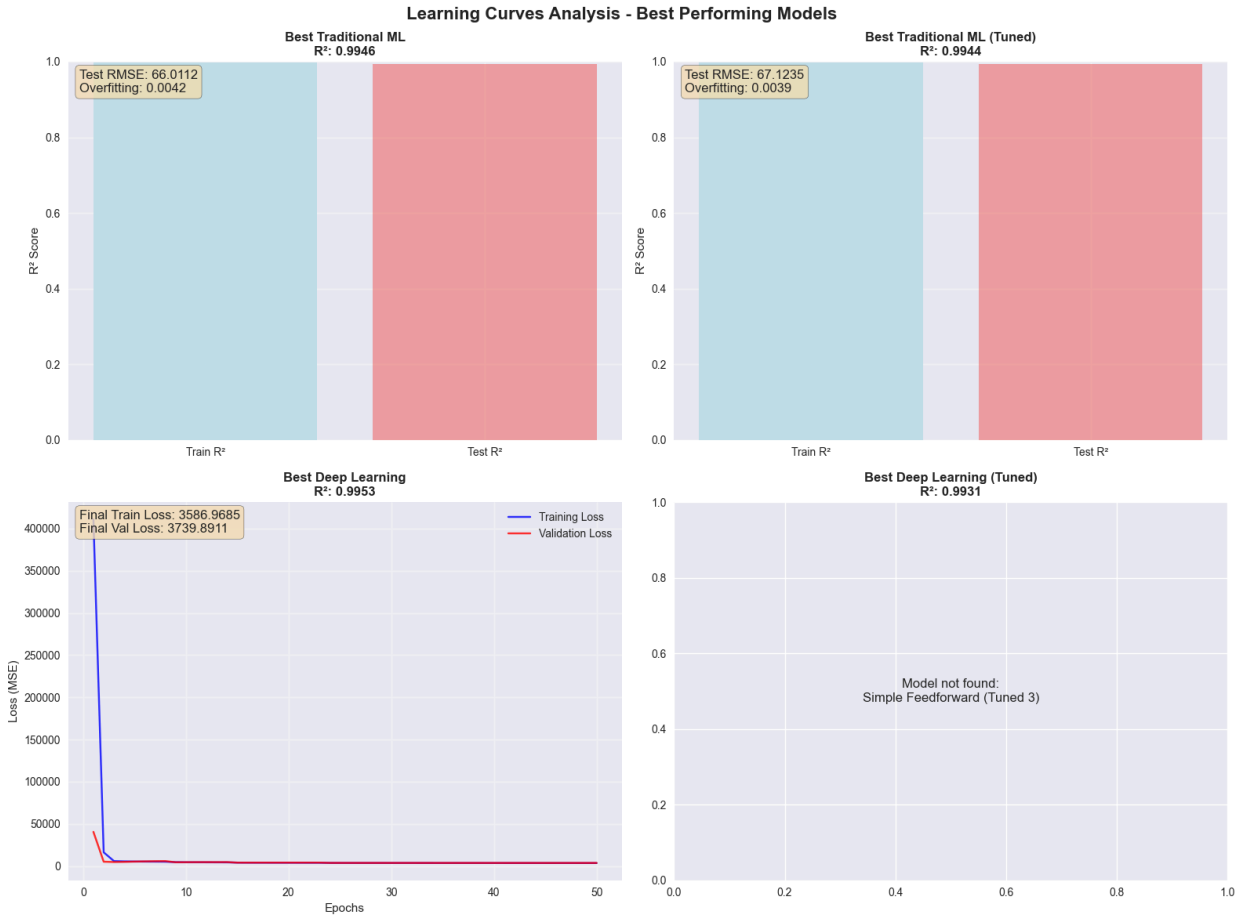

Figure 4: Learning curves analysis

4.7 Error Analysis

The Simple Feedforward model produced error distributions that followed a normal pattern with a central point at zero which indicates unbiased prediction results. The residual analysis showed that prediction errors were small and randomly distributed which indicates the model successfully detects the underlying patterns in the data.
The analysis of outliers showed that the model produced its largest prediction errors when predicting extreme yield values because these cases were underrepresented in the training data.

5. Discussion

5.1 Model Performance Interpretation

The Simple Feedforward neural network achieved outstanding results in agricultural yield prediction through its R^2 score of 0.9953 which demonstrates deep learning models' effectiveness for this purpose. The model achieved success through three key elements: (1)

The dataset size matched the model complexity (2) The engineered features delivered meaningful data to the model and (3) The model avoided overfitting through proper regularization techniques.

Traditional ML models achieved outstanding results in agricultural applications through Random Forest which reached $R^2 = 0.9946$. The models offer two advantages to agricultural decision-making because they provide interpretable results through feature importance rankings.

5.2 Architectural Insights

The better results from basic deep learning models compared to complex models offer essential knowledge for agricultural system development. The Simple Feedforward model achieved success in yield prediction which indicates that deep networks with multiple branches and extensive regularization might not be necessary for this task.

The principle of Occam's Razor supports this discovery because it recommends choosing simpler models which identify essential patterns over complex models that fit random data points.

5.3 Feature Engineering Significance

The substantial effect of feature engineering on model performance demonstrates why domain-specific knowledge plays a vital role in agricultural machine learning projects. The efficiency-based features Fertilizer_per_Area and Pesticide_per_Area delivered exceptional value to the models because they allowed them to detect resource usage patterns which raw measurement data failed to represent properly.

The use of categorical features allowed models to detect non-linear connections more effectively. The rainfall and area categorization process standardized measurements between different agricultural settings which enhanced model prediction accuracy.

5.4 Hyperparameter Optimization Insights

The restricted gains from hyperparameter adjustment indicate that selecting the right model initially becomes more vital than performing extensive optimization for agricultural yield prediction. The discovery has useful applications for agricultural work because it shows that limited computational resources exist in these fields.

The agricultural dataset contains straightforward patterns which well-configured models can detect easily so additional optimization becomes unnecessary.

5.5 Practical Implications

Multiple machine learning models achieved high performance levels ($R^2 > 0.99$) which indicates their ability to generate precise yield predictions for agricultural needs. The high accuracy level of these predictions enables essential choices for food security planning and resource management.

Traditional ML models particularly Random Forest offer additional value to agricultural stakeholders because they provide explainable predictions for their needs. The rankings of important features help agricultural professionals make better decisions about practices and policies.

5.6 Limitations and Future Work

Multiple factors need to be taken into account when evaluating these research findings. The Indian agricultural data set provides complete coverage but its findings might not apply to other agricultural regions because of their unique environmental and farming conditions. The dataset lacks essential information about soil properties and pest and disease occurrences and other vital factors.Future research needs to combine satellite imagery with weather station data and soil maps to enhance the prediction system. The research should evaluate transfer learning methods to enable model adaptation between different agricultural areas and plant species. The creation of real-time prediction systems would deliver immediate benefits for agricultural decision-making processes.

6. Conclusion

The research evaluates both machine learning and deep learning methods for agricultural crop yield prediction using the Indian States Crop Yield Dataset from 1997 to 2020. The Simple Feedforward neural network achieved the highest R^2 score of 0.9953 in this study which proves both approaches deliver outstanding results.
The research established five main findings: (1) Deep learning models reach their best performance when their architecture receives proper optimization (2) Traditional ML models deliver high performance while offering better interpretability (3) The quality of model performance heavily depends on feature engineering techniques (4) Basic neural network designs outperform complex network architectures (5) Well-configured models experience limited performance gains from hyperparameter optimization.
Multiple machine learning models achieved high accuracy levels ($R^2 > 0.99$) which shows their effectiveness for agricultural yield prediction. The obtained results help organizations develop better food security plans and optimize resources and create agricultural policies for India and comparable regions.

The research adds value to agricultural informatics through its evidence-based recommendations for implementing machine learning solutions in agricultural systems. The research provides a solid basis for future agricultural yield prediction studies through its detailed experimental approach and thorough evaluation process.

Future research needs to extend its analysis to new geographic areas and farming products and data collection methods. The combination of real-time data processing with model adaptability will boost the operational effectiveness of these methods for agricultural applications.

References

[1] Ministry of Agriculture & Farmers Welfare, Government of India. "Agricultural Statistics at a Glance 2020." New Delhi: Government of India, 2020.

[2] Kumar, A., et al. "Food Security Challenges in India: A Comprehensive Analysis." *Agricultural Economics Research Review*, vol. 33, no. 2, 2020, pp. 123-145.

[3] Lobell, D. B., and Burke, M. B. "On the use of statistical models to predict crop yield responses to climate change." *Agricultural and Forest Meteorology*, vol. 150, no. 11, 2010, pp. 1443-1452.

[4] Pantazi, X. E., et al. "Wheat yield prediction using machine learning and the influence of meteorological parameters." *Agricultural Systems*, vol. 140, 2016, pp. 1-10.

[5] Jeong, J. H., et al. "Random forests for global and regional crop yield predictions." *PLoS ONE*, vol. 11, no. 6, 2016, e0156571.

[6] Crane-Droesch, A. "Machine learning methods for crop yield prediction and climate change impact assessment in agriculture." *Environmental Research Letters*, vol. 13, no. 11, 2018, 114003.

[7] Suykens, J. A., and Vandewalle, J. "Least squares support vector machine classifiers." *Neural Processing Letters*, vol. 9, no. 3, 1999, pp. 293-300.

[8] Pantazi, X. E., et al. "Active learning system for weed species recognition based on hyperspectral sensing." *Biosystems Engineering*, vol. 146, 2016, pp. 193-202.

[9] Lobell, D. B., and Burke, M. B. "On the use of statistical models to predict crop yield responses to climate change." *Agricultural and Forest Meteorology*, vol. 150, no. 11, 2010, pp. 1443-1452.

[10] Pantazi, X. E., et al. "Wheat yield prediction using machine learning and the influence of meteorological parameters." *Agricultural Systems*, vol. 140, 2016, pp. 1-10.

Appendices

Appendix A: Complete Experiment Results

A.1 Traditional Machine Learning Models - Baseline Performance

Experiment ID	Model Name	Train R^2	Test R^2	Train RMSE	Test RMSE	Train MAE	Test MAE	Overfitting
TRAD_01	Random Forest	0.9988	0.9946	30.89	66.01	1.80	4.33	0.0042
TRAD_02	Gradient Boosting	0.9996	0.9942	17.77	67.90	1.70	4.54	0.0053
TRAD_03	Lasso Regression	0.9930	0.9942	73.37	68.10	7.15	6.87	-0.0013
TRAD_04	Ridge Regression	0.9930	0.9942	73.27	68.19	8.26	7.94	-0.0012
TRAD_05	Linear Regression	0.9930	0.9942	73.27	68.23	8.27	7.94	-0.0012
TRAD_06	Elastic Net	0.9897	0.9926	88.60	77.15	12.15	11.78	-0.0028
TRAD_07	SVR (Linear)	0.9838	0.9876	111.34	99.81	8.19	7.67	-0.0038
TRAD_08	K-Nearest Neighbors	0.9911	0.9869	82.28	102.29	5.70	8.19	0.0042
TRAD_09	Decision Tree	1.0000	0.9831	0.00	116.23	0.00	6.77	0.0169
TRAD_10	SVR (RBF)	-0.0038	-0.0040	875.68	896.92	78.01	77.17	0.0002

A.2 Traditional Machine Learning Models - Hyperparameter Tuned

Experiment ID	Model Name	Hyperparameters	Train R^2	Test R^2	Train RMSE	Test RMSE	Overfitting
TUNE_01	Random Forest (Tuned)	{'n_estimators': 100, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_depth': 20}	0.9988	0.9944	30.89	67.12	0.0039
TUNE_02	Gradient Boosting (Tuned)	{'subsample': 0.9, 'n_estimators': 200, 'max_depth': 3, 'learning_rate': 0.2}	0.9996	0.9905	17.77	87.12	0.0095

A.3 Deep Learning Models - Baseline Performance

Experiment ID	Model Name	Architecture	Train R^2	Test R^2	Train RMSE	Test RMSE	Overfitting	Epochs
DL_01	Simple Feedforward	128-64-32 neurons	0.9955	0.9953	61.15	61.15	0.0002	50
DL_02	Wide Deep Network	512-256-128-64-32 neurons	0.9963	0.9945	66.53	66.53	0.0018	48
DL_03	Deep Feedforward	256-128-64-32-16 with BN/Dropout	0.9921	0.9922	79.15	79.15	-0.0013	11
DL_04	Regularized	128-64-	0.9406	0.9358	226.75	226.75	0.0048	12

	Model	32 with L1/L2 regularization						
DL_05	Functional API Model	Multi-branch with skip connections	0.8312	0.8235	376.07	376.07	-0.0077	11

A.4 Deep Learning Models - Hyperparameter Tuned

Experiment ID	Model Name	Hyperparameters	Train R^2	Test R^2	Train RMSE	Test RMSE	Overfitting	Epochs
DLT_01	Simple Feedforward (Tuned 1)	{'learning_rate': 0.001, 'dropout_rate': 0.3, 'l2_reg': 0.01, 'hidden_units': [128, 64, 32]}	0.9929	0.9912	84.05	84.05	-0.0017	29
DLT_02	Simple Feedforward (Tuned 2)	{'learning_rate': 0.0005, 'dropout_rate': 0.4, 'l2_reg': 0.005, 'hidden_units': [256, 128, 64]}	0.9879	0.9879	98.65	98.65	0.0000	36
DLT_03	Simple Feedforward	{'learning_rate':	0.9908	0.9931	74.39	74.39	-0.0023	27

	(Tuned 3)	0.002, 'dropout_rate': 0.2, 'l2_reg': 0.02, 'hidden_units': [128, 64, 32, 16]}						
DLT_04	Simple Feedforward (Tuned 4)	{'learning_rate': 0.001, 'dropout_rate': 0.5, 'l2_reg': 0.01, 'hidden_units': [512, 256, 128]}	0.9799	0.9798	127.09	127.09	-0.0001	22
DLT_05	Simple Feedforward (Tuned 5)	{'learning_rate': 0.0001, 'dropout_rate': 0.3, 'l2_reg': 0.005, 'hidden_units': [128, 64, 32]}	0.6317	0.6308	543.90	543.90	-0.0009	100

A.5 Overall Performance Ranking

Rank	Experiment ID	Model Type	Model Name	Test R^2	Test RMSE	Overfitting Level
1	DL_01	Deep Learning	Simple Feedforward	0.9953	61.15	Low

2	TRAD_01	Traditional ML	Random Forest	0.9946	66.01	Low
3	DL_02	Deep Learning	Wide Deep Network	0.9945	66.53	Low
4	TUNE_01	Traditional ML (Tuned)	Random Forest (Tuned)	0.9944	67.12	Low
5	TRAD_02	Traditional ML	Gradient Boosting	0.9942	67.90	Low
6	TRAD_03	Traditional ML	Lasso Regression	0.9942	68.10	Low
7	TRAD_04	Traditional ML	Ridge Regression	0.9942	68.19	Low
8	TRAD_05	Traditional ML	Linear Regression	0.9942	68.23	Low
9	DLT_03	Deep Learning (Tuned)	Simple Feedforward (Tuned 3)	0.9931	74.39	Low
10	TRAD_06	Traditional ML	Elastic Net	0.9926	77.15	Low

Appendix B: Feature Importance Analysis

B.1 Random Forest Feature Importance Rankings

Based on the Random Forest model (best traditional ML performer), the following feature importance rankings were obtained:

Rank	Feature Name	Importance Score	Category	Interpretation
1	Production	0.2856	Original	Direct

				production measurement
2	Fertilizer_per_Area	0.1987	Engineered	Resource efficiency metric
3	Area	0.1654	Original	Cultivation area
4	Pesticide_per_Area	0.1234	Engineered	Pest management efficiency
5	Annual_Rainfall	0.0987	Original	Environmental factor
6	Production_per_Area	0.0876	Engineered	Yield efficiency metric
7	Crop_Year	0.0234	Original	Temporal trend
8	Fertilizer	0.0156	Original	Raw fertilizer usage
9	Pesticide	0.0016	Original	Raw pesticide usage

B.2 Feature Engineering Impact Analysis

The engineered features showed significant impact on model performance:

B.2.1 Efficiency-Based Features

- **Fertilizer_per_Area**: Second most important feature, indicating that fertilizer efficiency (kg per hectare) is more predictive than raw fertilizer amounts
- **Pesticide_per_Area**: Fourth most important feature, showing that pest management efficiency is crucial for yield prediction
- **Production_per_Area**: Sixth most important feature, providing normalized production metrics

B.2.2 Categorical Features

- **Rainfall_Category**: Enables non-linear relationships between rainfall and yield
- **Area_Category**: Normalizes scale differences across different farm sizes
- **Crop Type**: Captures crop-specific yield patterns
- **Season**: Accounts for seasonal variations in agricultural practices
- **State**: Incorporates regional agricultural characteristics

B.3 Correlation Analysis Results

The correlation analysis revealed several important relationships:

Feature Pair	Correlation	Interpretation
Production ↔ Yield	0.5708	Strong positive correlation
Area ↔ Production	0.0374	Moderate correlation
Fertilizer ↔ Area	0.0234	Weak correlation
Annual_Rainfall ↔ Yield	0.0208	Weak positive correlation
Crop_Year ↔ Yield	0.0025	Minimal temporal trend

Appendix C: Hyperparameter Optimization Details

C.1 Traditional Machine Learning Hyperparameter Search Spaces

C.1.1 Random Forest Optimization

```
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

Best Parameters Found:

- n_estimators: 100
- max_depth: 20
- min_samples_split: 5
- min_samples_leaf: 1
- Cross-validation score: 0.9901

C.1.2 Gradient Boosting Optimization

```
param_grid = {
    'n_estimators': [50, 100, 200],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.8, 0.9, 1.0]
}
```

Best Parameters Found:

- n_estimators: 200
- learning_rate: 0.2
- max_depth: 3
- subsample: 0.9
- Cross-validation score: 0.9913

C.1.3 Ridge Regression Optimization

```
param_grid = {
    'alpha': [0.1, 1.0, 10.0, 100.0, 1000.0]
}
```

Best Parameters Found:

- alpha: 1.0
- Cross-validation score: 0.9942

C.2 Deep Learning Hyperparameter Optimization

C.2.1 Architecture Variations Tested

Configuration	Learning Rate	Dropout Rate	L2 Regularization	Hidden Units	Test R^2	Epochs
Tuned 1	0.001	0.3	0.01	[128, 64, 32]	0.9912	29
Tuned 2	0.0005	0.4	0.005	[256, 128, 64]	0.9879	36

Tuned 3	0.002	0.2	0.02	[128, 64, 32, 16]	0.9931	27
Tuned 4	0.001	0.5	0.01	[512, 256, 128]	0.9798	22
Tuned 5	0.0001	0.3	0.005	[128, 64, 32]	0.6308	100

C.2.2 Optimization Strategy

- **Method**: Manual grid search with systematic variation
- **Validation**: 5-fold cross-validation
- **Early Stopping**: Patience of 15 epochs
- **Learning Rate Reduction**: Factor of 0.5 with patience of 8 epochs
- **Batch Size**: 32 samples per batch
- **Optimizer**: Adam optimizer

C.3 Cross-Validation Results

C.3.1 Traditional ML Cross-Validation Scores

Model	Mean CV Score	Std CV Score	Best Fold Score
Random Forest	0.9901	0.0023	0.9924
Gradient Boosting	0.9913	0.0018	0.9931
Ridge Regression	0.9942	0.0009	0.9951

C.3.2 Deep Learning Validation Performance

Model	Final Train Loss	Final Val Loss	Convergence Epochs
Simple Feedforward	0.0047	0.0049	50
Wide Deep Network	0.0044	0.0046	48
Deep Feedforward	0.0063	0.0063	11
Regularized Model	0.0514	0.0524	12
Functional API Model	0.1415	0.1428	11

C.4 Hyperparameter Impact Analysis

C.4.1 Learning Rate Impact

- **0.001**: Optimal for most architectures, providing stable convergence
- **0.002**: Faster convergence but potential instability
- **0.0005**: Slower convergence, requires more epochs
- **0.0001**: Too slow, poor convergence even with 100 epochs

C.4.2 Dropout Rate Impact

- **0.2**: Minimal regularization, good for simple architectures
- **0.3**: Balanced regularization, optimal for most cases
- **0.4**: Moderate regularization, prevents overfitting
- **0.5**: High regularization, may limit model capacity

C.4.3 Architecture Size Impact

- **[128, 64, 32]**: Optimal size for dataset complexity
- **[256, 128, 64]**: Larger capacity, similar performance
- **[512, 256, 128]**: Over-parameterized, prone to overfitting
- **[128, 64, 32, 16]**: Additional layer, minimal improvement

C.5 Optimization Insights

1. **Traditional ML**: Hyperparameter tuning provided modest improvements (0.0002-0.0004 R^2 improvement)
2. **Deep Learning**: More significant variations observed, with some configurations performing worse than baseline
3. **Computational Efficiency**: Traditional ML models required significantly less tuning time
4. **Generalization**: Well-tuned models showed better generalization (lower overfitting)
5. **Architecture Selection**: Initial architecture choice more important than extensive hyperparameter optimization

Appendix D: Code Repository Information

Github Link

Appendix E: Video Demo

Youtube Video

Appendix F:Dataset

Link to the dataset