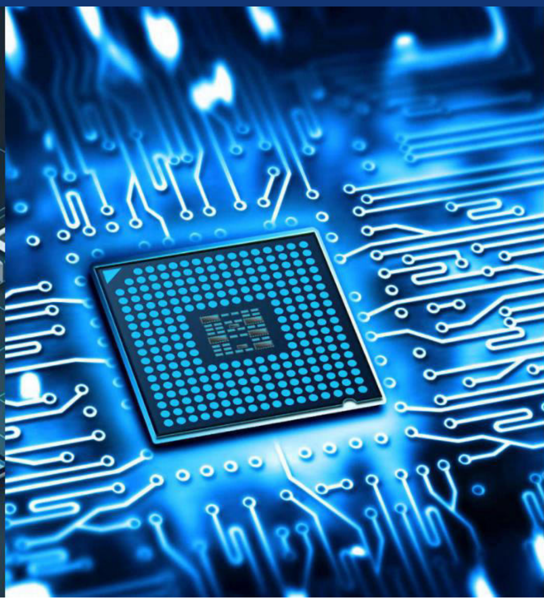
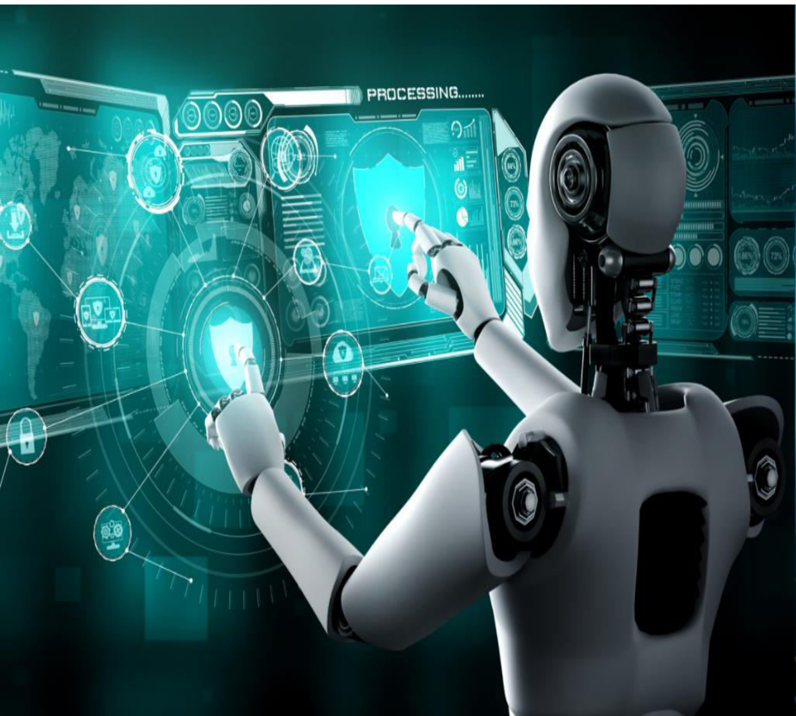
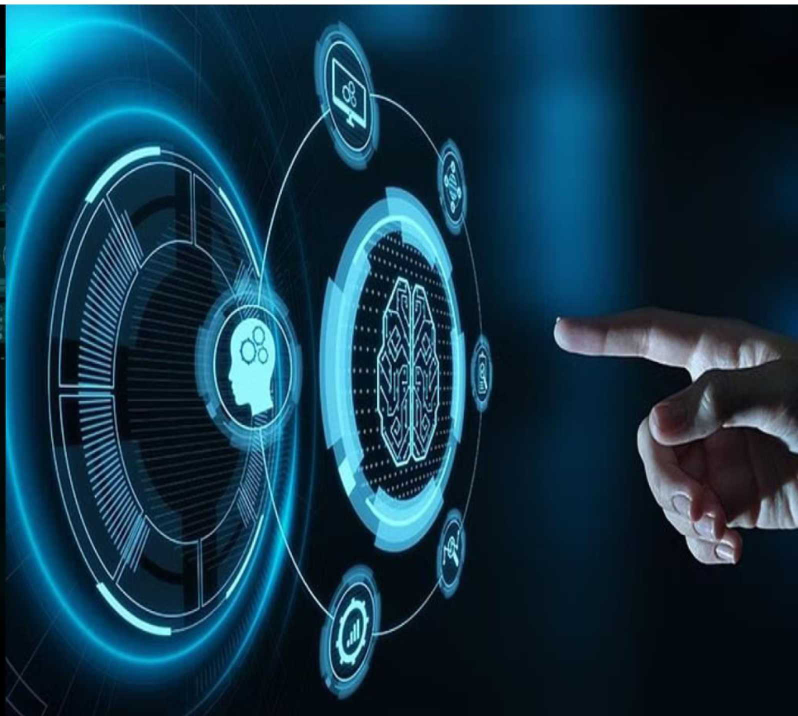
ISSN(O): 2320-9801
ISSN(P): 2320-9798


International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771 **Volume 13, Issue 5, May 2025**

www.ijircce.com ijircce@gmail.com +91-9940572462 +91 63819 07438

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305074

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A Hybrid XGBoost and Per-Commodity LSTM Ensemble Model for Agri-Horticultural Price Forecasting with Recursive Future Prediction

Ritu Bhakal[1], Akankssha[2], Sai Praneeth L[3], Dr. Leelambika KV[4]

Department of Computer Science, Presidency University, Bengaluru, India[1-3]

Assistant Professor, Department of Computer Science, Presidency University, Bengaluru, India[4]

ABSTRACT: Accurate forecasting of agricultural commodity prices is crucial for market stability, food security, and the economic well-being of farmers. This paper proposes a hybrid ensemble learning framework designed to predict daily agricultural commodity prices by integrating the strengths of Extreme Gradient Boosting (XGBoost) and Long Short-Term Memory (LSTM) networks. The methodology leverages a dataset of historical daily market prices and associated meteorological data. XGBoost is utilized to model complex non-linear relationships across a comprehensive feature set, including lagged variables and temporal indicators. Concurrently, distinct LSTM models are trained for each commodity to capture specific sequential patterns and temporal dependencies within their respective price histories. Predictions from these models are ensembled using a simple averaging technique. For forecasting prices on future dates beyond the historical data horizon, a recursive prediction strategy is employed, where predicted prices and estimated environmental features iteratively feed back into the models. The ensemble model was rigorously validated on a held-out test set using an instance-wise approach that leverages actual historical sequences for LSTM inputs. The proposed system achieved a Mean Absolute Error (MAE) of 2.9682, Root Mean Squared Error (RMSE) of 5.2986, an R-squared (R^2) value of 0.9865, and a Mean Absolute Percentage Error (MAPE) of 6.02%, demonstrating high predictive accuracy and its potential as a valuable tool for agricultural market stakeholders.

KEYWORDS: Agricultural Price Prediction, Ensemble Learning, XGBoost, LSTM, Time Series Forecasting, Recursive Prediction, Feature Engineering, Hybrid Models.

I. INTRODUCTION

The agricultural sector plays a pivotal role in global economies, directly impacting food security and the livelihoods of millions. However, agricultural commodity markets are inherently volatile, influenced by a confluence of factors such as climatic conditions, seasonality, geopolitical events, and supply-demand dynamics [1]. This price instability presents significant challenges for farmers in planning and obtaining fair returns, for traders in managing inventory and risk, and for policymakers in ensuring market stability and consumer affordability.

Traditional econometric models, such as ARIMA, have long been employed for time series forecasting but often struggle to capture the complex non-linearities and the impact of diverse exogenous variables prevalent in agricultural markets. The advent of machine learning (ML) and deep learning (DL) has offered more powerful tools. ML algorithms like Support Vector Machines and tree-based ensembles can model non-linear relationships effectively. DL models, particularly Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM), have shown exceptional capability in learning long-range dependencies from sequential data, making them well-suited for price time series.

This research presents a hybrid ensemble approach that synergizes Extreme Gradient Boosting (XGBoost), a powerful gradient boosting algorithm, with per-commodity LSTM networks. XGBoost is employed to capture complex interactions among a wide array of features, including lagged prices, weather data, and temporal indicators, providing a global perspective. Simultaneously, individual LSTM models are trained for each specific commodity, allowing them to specialize in the unique temporal patterns inherent in that commodity's price history. The predictions from these complementary models are then ensembled. A key aspect of this work is the implementation of a recursive forecasting mechanism that enables multi-step ahead predictions for future dates where actual data is unavailable. The model's performance is validated using an instance-wise approach on a test set, which provides a realistic assessment of its predictive capabilities on unseen data by ensuring LSTM inputs are based on actual prior historical sequences.

The primary contributions of this study are:

• The development and evaluation of a hybrid XGBoost and per-commodity LSTM ensemble model for daily agricultural price prediction.

• The implementation and assessment of a recursive forecasting strategy for multi-step ahead predictions.

• A robust validation methodology demonstrating the ensemble's high accuracy on a held-out test set.

II. RELATED WORK

The domain of agricultural price forecasting has witnessed a significant evolution, moving from traditional statistical methods to more sophisticated machine learning and deep learning techniques. Understanding this progression is crucial for contextualizing the contributions of the present study.

2.1 Statistical and Econometric Models

Early research in time series forecasting, including agricultural prices, predominantly relied on statistical models. The Autoregressive Integrated Moving Average (ARIMA) model and its seasonal counterpart, SARIMA, as formalized by Box et al. [1], have been widely adopted. These models excel at capturing linear dependencies and seasonality within price series. For instance, Weng et al. [2] compared ARIMA with neural network approaches for horticultural product prices, noting ARIMA's better performance for monthly data but struggles with daily fluctuations. Similarly, Wang [3] utilized ARIMA for soybean futures, and Sneha and Bhavana [4] applied it for sugarcane price forecasting in India. However, a primary limitation of these models is their inherent assumption of linearity, which often proves insufficient for capturing the complex, non-linear dynamics frequently observed in volatile agricultural markets; Kim [5], for instance, notes such limitations before exploring SVMs.

2.2 Conventional Machine Learning Approaches

With advancements in computational power, machine learning (ML) algorithms offered powerful alternatives capable of handling non-linearities. Support Vector Machines (SVM), particularly Support Vector Regression (SVR), have been applied in financial time series forecasting, as demonstrated by Kim [5]. Tree-based ensemble methods also gained prominence. Random Forests (RF) [6] and optimized gradient boosting implementations like XGBoost, detailed by Chen & Guestrin [7], have become popular choices. XGBoost, in particular, has shown significant utility in predicting daily vegetable prices by incorporating meteorological factors, as highlighted by Li et al. [8], underscoring its suitability for tasks involving diverse exogenous variables. Cifci [9] further showcased the broad applicability of various ML models, including RF, SVM, and boosting variants, in complex system predictions.

2.3 Deep Learning Models for Time Series

Deep learning, especially Recurrent Neural Networks (RNNs), marked a significant advancement for modeling sequential data like price time series. Long Short-Term Memory (LSTM) networks, proposed by Hochreiter & Schmidhuber [10], effectively addressed the vanishing gradient problem inherent in simple RNNs, enabling them to learn long-range dependencies. This capability has led to widespread applications. Fischer & Krauss [11] demonstrated LSTM's effectiveness in financial market predictions. In the agricultural domain, Meena & Chaitra [12] applied LSTM, Gated Recurrent Units (GRU), and 1D-Convolutional Neural Networks (CNN) for Ragi price prediction. Halim et al. [13] focused on multivariate LSTM for predicting multiple commodity prices simultaneously, emphasizing the importance of handling inter-commodity relationships. Wang & Gao [14] also employed LSTM for soybean futures prediction. More recently, Transformer models, such as those described by Vaswani et al. [15] and further developed for time series by works like Zhou et al. [16] (Informer), have shown promise in capturing long-range dependencies. Applications include the EEMD-NAGU model by Li et al. [17] and multimodal transformers for stock prediction by Lee et al. [18].

2.4 Hybrid and Ensemble Strategies

Recognizing that different models possess unique strengths, researchers have increasingly explored hybrid and ensemble strategies to achieve superior performance. Hybrid models often combine statistical approaches with ML/DL techniques; for instance, Babu & Reddy [19] proposed a hybrid ARIMA-ANN model. Li et al. [17] combined Ensemble Empirical Mode Decomposition (EEMD) with an attention-based GRU variant (NAGU). Yu et al. [20] similarly used EMD as a preprocessing step for an ensemble neural network approach for crude oil price forecasting. These strategies aim to either decompose complex time series or leverage the complementary modeling strengths of different algorithms. The approach

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

in our paper aligns with this trend by creating a hybrid ensemble of XGBoost and per-commodity LSTMs, seeking to harness XGBoost's strength in handling structured, tabular data with diverse features and LSTM's proficiency in modeling temporal sequences specific to individual commodities. The per-commodity specialization of LSTMs is a key aspect, aiming to tailor temporal modeling more precisely than a single, global LSTM might achieve.

III. METHODOLOGY

The proposed methodology for agricultural commodity price prediction involves several stages:

3.1. Data Acquisition and Preprocessing

The dataset utilized in this study comprises historical daily data for agricultural and horticultural commodities like tomato, potato, onion,rice,wheat,etc, including minimum, maximum, and modal prices (originally in Rs./Quintal), along with corresponding daily meteorological data (e.g., average temperatures, rainfall, humidity, wind speed).
The data is collected from government websites, namely agmarknet.gov.in and India Meteorological Department.
The preprocessing pipeline includes:
Data Loading and Initial Cleaning: The raw dataset (e.g., finaldataset.csv) is loaded.
Date Handling: The 'Date' column is parsed into datetime objects. Any records with invalid dates are dropped. The data is then sorted chronologically by 'Commodity' and 'Date', which is essential for time series integrity and lagged feature creation.
Price Unit Conversion: All price columns are converted from Rs./Quintal to Rs./Kg by dividing by 100. The modal price in Rs./Kg (termed PRICE_COLUMN_KG or 'Price') is designated as the target variable for prediction.
Categorical Feature Encoding: The 'Commodity' names are converted into numerical representations using LabelEncoder (e.g., Commodity_enc), enabling ML models to process this information.

3.2. Feature Engineering

Effective feature engineering is crucial for model performance. The following features are generated:
Temporal Features: From the 'Date', DayOfYear, Month, and Year are extracted. These help the model capture seasonality, cyclical patterns, and long-term trends.
Lagged Price Features: To capture the strong autocorrelation often present in price series, lagged values of the target 'Price' (and potentially other price-related metrics like original quintal price for consistency if used for lagging before conversion) are created. Specifically, prices from 1, 2, and 3 days prior (e.g., Price_lag_1, Price_lag_2, Price_lag_3) are generated. This is performed on a per-commodity basis using groupby('Commodity').shift(lag) to ensure lags are commodity-specific and do not cross-contaminate series.
Lagged Weather Features: Given that weather can have a delayed impact on agricultural markets, lagged values of key weather variables (e.g., avg_max_temp_lag_1, avg_max_temp_lag_2, avg_max_temp_lag_3) are also created, again on a per-commodity basis.
Final Feature Set and Cleaning: The comprehensive list of features (all_input_features) includes original prices (Min, Max in Rs./Kg), encoded commodity, current and lagged weather variables, temporal features, and lagged price features. After generating all features, rows containing any NaN values (which primarily arise from the lag creation process at the beginning of each commodity's time series) are removed to ensure complete data for model training.

3.3. Model Architectures

Two distinct modeling paradigms are employed: a global XGBoost model and specialized per-commodity LSTM networks.

3.3.1. XGBoost Model Component

Extreme Gradient Boosting (XGBoost) is a highly efficient and effective tree-based ensemble learning algorithm [Reference Chen & Guestrin]. It builds decision trees sequentially, with each new tree aiming to correct the errors of the previous ones. XGBoost is known for its performance, regularization capabilities (L1 and L2) to prevent overfitting, and its ability to handle sparse data.
In this framework, a single XGBRegressor model is trained using the all_input_features from all commodities. The input features for XGBoost (X_train_full, X_test_full) are scaled using a global MinMaxScaler (fit only on the training data) prior to training. Typical hyperparameters configured include n_estimators, learning_rate, max_depth, subsample, and colsample_bytree, with the objective set to reg:squarederror.

3.3.2. Per-Commodity LSTM Model Component

Long Short-Term Memory (LSTM) networks are a type of RNN adept at learning long-term dependencies in sequential data, making them suitable for time series forecasting [Reference Hochreiter & Schmidhuber]. LSTMs use internal gating mechanisms (forget, input, and output gates) to regulate information flow, mitigating the vanishing/exploding gradient problems common in simpler RNNs.

Recognizing that different agricultural commodities can exhibit unique price dynamics, seasonality, and responses to market factors, separate LSTM models are trained for each unique commodity present in the dataset. This allows each LSTM to specialize its learning. For each commodity:

Data Subsetting: The historical data is filtered for the specific commodity.

Scaling: The input features for that commodity are scaled using a commodity-specific MinMaxScaler (fit only on that commodity's training data).

Sequence Generation: The scaled features and target prices are transformed into input sequences of a fixed length (N_TIMESTEPS_LSTM = 7 in the code). Each input sample (X) for the LSTM consists of feature data from the past N_TIMESTEPS_LSTM days, and the corresponding target (y) is the price on the day immediately following the sequence.

LSTM Architecture: A Sequential Keras model is constructed, typically comprising:

- An LSTM layer with a specified number of units, return_sequences= True (if followed by another LSTM layer), and an input shape of (N_TIMESTEPS_LSTM, number_of_features_per_timestep).
- Dropout layers (e.g., rate 0.2) for regularization to prevent overfitting.
- A subsequent LSTM layer (e.g., 32 units), with return_sequences=False.
- A Dense hidden layer (e.g., 16 units with 'relu' activation).
- A final Dense output layer with a single unit for predicting the price.

Compilation and Training: Each LSTM model is compiled using the 'adam' optimizer and 'mean_squared_error' as the loss function. EarlyStopping (monitoring validation loss, e.g., val_loss) is employed during training with a defined patience to prevent overfitting and restore the best model weights. Commodities with insufficient data to form meaningful sequences or train/test splits are skipped.

3.4. Ensemble Strategy

To leverage the diverse strengths of both XGBoost (handling broad feature interactions) and the specialized LSTMs (capturing commodity-specific temporal patterns), their predictions are combined. In this implementation, a simple averaging ensemble is used:

Ensemble Prediction = (XGBoost Prediction + LSTM Prediction) / 2

This straightforward approach provides a balanced forecast. For instance-wise validation on the test set, if an LSTM prediction is unavailable for a specific commodity or instance (e.g., insufficient historical data for sequence formation for that test point), the LSTM prediction component might effectively default to the XGBoost prediction, or a mechanism to handle missing LSTM predictions would be in place to ensure the ensemble calculation can proceed.

3.5. Recursive Forecasting for Future Dates

A critical requirement for a practical price forecasting system is the ability to predict prices for future dates where actual market and weather data are not yet available. The predict_price_for_future_date function implements a recursive multi-step forecasting strategy:

1. Initialization:

- The process starts from the day immediately following the last known historical date for the selected commodity. The most recent actual features, including lagged prices and weather, serve as the initial state for generating future lags.
- An LSTM input deque (lstm_sequence_input_deque) is populated with the scaled feature vectors from the last N_TIMESTEPS_LSTM actual historical days for that commodity (if an LSTM model exists for it).

2. Iterative Prediction Loop: The loop progresses one day at a time (current_pred_date) from the day after the last known data up to the user-specified target_date. For each current_pred_date:

- **Estimate Exogenous Features:** Non-price, non-lagged features (primarily weather, but also date-derived features like DayOfYear, Month, Year) for current_pred_date are estimated. The get_estimated_weather_and_other_features function in the code uses historical averages for the commodity around the same day of the year, with fallbacks to broader averages if specific data is sparse. Min/Max prices are also naively estimated (e.g., as a percentage of the previous day's predicted modal price).

• **Construct Feature Vector:** A full feature vector for current_pred_date is assembled. This includes the estimated exogenous features and lagged features. Lagged prices (e.g., Price_lag_1) are derived from the previous day's (which would be a predicted price if beyond the first recursive step) state. Similarly for lagged weather features.
• **XGBoost Prediction:** The constructed feature vector is scaled (using the global XGBoost scaler) and fed to the trained XGBoost model to obtain predicted_price_xgb_for_step
• **LSTM Prediction:** If an LSTM model exists for the commodity and the lstm_sequence_input_deque contains a full sequence of N_TIMESTEPS_LSTM feature vectors, this sequence is fed to the LSTM model to obtain predicted_price_lstm_for_step.
• **Ensemble for Step:** The XGBoost and LSTM predictions are ensembled (averaged) to get ensembled_price_for_step. A non-negativity constraint is applied (price >= 0).
• **Update State for Next Step:** The dictionary holding the current day's features (current_day_features_dict) is updated. The ensembled_price_for_step becomes the 'Price' for current_pred_date. Lagged price and weather features are shifted (e.g., previous Price_lag_1 becomes new Price_lag_2), and current weather features are updated with the estimated values.
• **Update LSTM Deque:** The feature vector representing current_pred_date (which was used to make the predictions) is scaled using the commodity-specific LSTM scaler and appended to the lstm_sequence_input_deque.

3.Termination: The loop continues until the target_date is reached. The ensembled prediction for this target date is then returned. This recursive process allows for projections, though accuracy typically diminishes with an increasing forecast horizon due to error accumulation.

IV. EXPERIMENTAL SETUP

4.1. Dataset and Splitting

The study utilize a consolidated dataset comprising daily historical commodity prices and corresponding meteorological data. For model training and evaluation, the entire dataset is split chronologically into a training set (80% of the data) and a test set (the remaining 20%). The shuffle=False parameter is strictly enforced during this split to ensure that the test set represents data points that are chronologically later than the training set, simulating a real-world forecasting scenario and preventing data leakage from the "future" into the "past." This global split is used for the XGBoost model and the final ensemble validation. For per-commodity LSTM models, their respective data subsets are also split chronologically for training and internal validation.

4.2. Evaluation Metrics

The performance of the proposed ensemble model is assessed using standard regression evaluation metrics:
• **Mean Absolute Error (MAE):** Measures the average absolute difference between actual and predicted values.
$MAE = (1/n) * \Sigma|actual_i - predicted_i|$
• **Root Mean Squared Error (RMSE):** Represents the square root of the average of squared differences, penalizing larger errors more heavily. It is in the same units as the target variable.
$RMSE = sqrt((1/n) * \Sigma(actual_i - predicted_i)^2)$
• **R-squared (R^2):** Indicates the proportion of the variance in the dependent variable (price) that is predictable from the independent variables (features). Values closer to 1 indicate a better fit.
$R^2 = 1 - (SS\_res / SS\_tot)$, where SS_res = sum of squared residuals, SS_tot = total sum of squares.
• **Mean Absolute Percentage Error (MAPE):** Expresses the average absolute error as a percentage of the actual values. It provides a relative measure of error.
$MAPE = (1/n) * \Sigma(|(actual_i - predicted_i) / actual_i|) * 100\%$
(Note: MAPE can be sensitive if actual values are zero or very close to zero.)

4.3. Validation of Ensemble on Test Data (Instance-wise)

A key aspect of this study's evaluation is the rigorous validation of the ensemble strategy on the global test set (X_test_full, y_test_full). This is performed instance-wise for each data point in the test set:
1. For a given test instance i, the XGBoost model generates a prediction (xgb_pred) using its scaled features.
2. To obtain the LSTM prediction for the same instance i:
• The specific commodity and date of instance i are identified from the test set metadata.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

• The N_TIMESTEPS_LSTM sequence of actual historical feature vectors immediately preceding the date of instance i for that specific commodity is retrieved from the complete historical dataset (df). This ensures the LSTM input is based on true past data, not recursively generated data, for this validation step.
• This historical sequence is scaled using the appropriate commodity-specific LSTM scaler.
• The scaled sequence is fed into the trained LSTM model for that commodity to get its prediction (lstm_pred).
• If an LSTM model is unavailable for the commodity or a full sequence cannot be formed, a fallback is used (e.g., lstm_pred defaults to xgb_pred to allow ensemble calculation).
3. The xgb_pred and lstm_pred are then combined using the simple averaging ensemble strategy.
This instance-wise validation provides a realistic assessment of how the ensemble would perform when LSTM has access to genuine preceding history for each point it predicts in the test phase. The performance metrics reported in Section 5 are derived from this meticulous validation process.

V. RESULTS AND DISCUSSION

The hybrid XGBoost and per-commodity LSTM ensemble model was evaluated on the held-out test set according to the instance-wise validation protocol described in Section 4.3. The aggregated performance metrics are presented in Table 1.
Table 1: Ensemble Model Performance on Test Set

Metric	**Value**
MAE	2.9682
RMSE	5.2986
R^2	0.9865
MAPE (%)	6.02

5.1. Quantitative Performance Analysis

The results shown in Table 1 indicate a strong predictive capability of the proposed ensemble model. An R-squared (R^2) value of 0.9865 is particularly noteworthy. This suggests that approximately 98.65% of the variance in the daily modal prices (in Rs./Kg) within the test set can be explained by the model, signifying an excellent fit to the unseen data and high explanatory power.
The Mean Absolute Error (MAE) of 2.9682 indicates that, on average, the model's price predictions deviate from the actual prices by approximately ₹2.97 per Kg. The Root Mean Squared Error (RMSE) of 5.2986, being higher than the MAE as expected due to its penalization of larger errors, still represents a relatively low error margin in the context of potentially volatile agricultural prices. The magnitude of these errors is generally acceptable for many practical decision-making scenarios in agricultural trading and planning.
A Mean Absolute PercentageError (MAPE) of 6.02% further supports the model's accuracy, implying that the average prediction error is approximately 6% relative to the actual price. This level of percentage error is often considered good in price forecasting applications.

5.2. Qualitative Visual Analysis

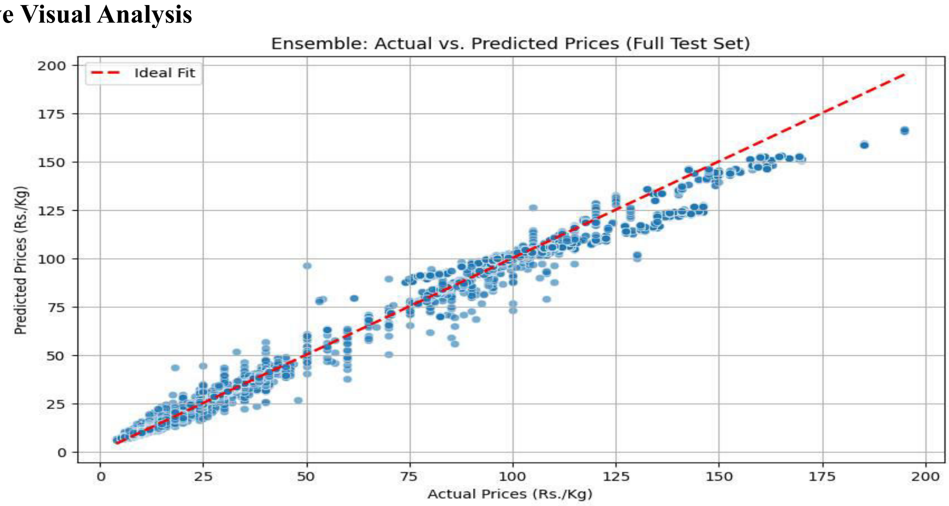


Figure 1: Ensemble: Actual vs. Predicted Prices (Full Test Set)

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The scatter plot presented in Figure 1 visually compares the actual prices against the ensemble model's predicted prices for the entire test set. The points generally cluster tightly around the "Ideal Fit" line (y=x), which is dashed in red. This visual evidence corroborates the high R^2 value, indicating a strong positive correlation and good agreement between the predicted and actual values across a wide range of prices. While most points are close to the ideal line, some scatter, particularly at higher price points, suggests areas where the model might face greater challenges, possibly due to heightened volatility or less frequent extreme price events in the training data. Overall, the plot demonstrates the model's robust ability to track actual price movements.

5.3. Discussion of Findings

The strong performance metrics and visual agreement demonstrate the efficacy of the hybrid ensemble approach. The combination of XGBoost's ability to model complex interactions across a diverse feature set and the per-commodity LSTMs' specialization in capturing unique temporal dependencies appears to be synergistic. XGBoost likely excels at incorporating the immediate impact of lagged prices, weather, and broader market conditions, while LSTMs refine predictions by learning subtle, commodity-specific sequential nuances over time.

The recursive forecasting mechanism, while essential for practical future prediction, inherently faces challenges. The reliance on estimated weather features (based on historical averages) and naively estimated Min/Max prices for future steps introduces potential inaccuracies. The further into the future the prediction extends, the more these estimations and the model's own prediction errors from previous steps can compound, typically leading to a degradation in forecast accuracy over longer horizons. This is a common limitation in multi-step time series forecasting.

VI. CONCLUSION AND FUTURE WORK

This research successfully developed and validated a hybrid ensemble model combining XGBoost and per-commodity LSTM networks for daily agricultural price prediction. The model demonstrated high predictive accuracy on a held-out test set, achieving an R^2 of 0.9865, MAE of 2.9682, RMSE of 5.2986, and MAPE of 6.02%. The inclusion of comprehensive feature engineering, tailored LSTM models for individual commodities, and a practical recursive forecasting function highlights its potential utility for various stakeholders in the agricultural sector, aiding in informed decision-making regarding trading, storage, and policy.

Future research could explore several avenues to enhance the current framework:

1. Hyperparameter Optimization: Rigorous hyperparameter optimization for both XGBoost and the LSTM architectures (e.g., using Bayesian optimization or genetic algorithms) could further fine-tune model performance.

2.Attention Mechanisms and Transformers: Incorporating attention mechanisms within LSTMs or exploring Transformer-based models for the time series component might allow the model to better capture salient long-range dependencies and complex temporal patterns.

3.Causal Inference: Moving beyond correlation to explore causal inference techniques could provide deeper insights into the actual drivers of price movements.

By addressing these areas, the robustness and utility of agricultural price forecasting systems can be continually improved, contributing to greater efficiency and stability in the agricultural domain.

REFERENCES

[1] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control. John Wiley & Sons.

[2] Weng, Y., Wang, X., Hua, J., Wang, H., Kang, M., & Wang, F.-Y. (2019). Forecasting Horticultural Products Price Using ARIMA Model and Neural Network Based on a Large-Scale Data Set Collected by Web Crawler. IEEE Transactions on Computational Social Systems, 6(3), 547–553.

[3] Wang, C. Y. (2016). Forecast on price of agricultural futures in China based on ARIMA model. Asian Journal of Agricultural Research, 11(9), 12-16.

[4] Sneha, V., & Bhavana, V. (2023). Sugarcane yield and price prediction using forecasting models. In 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF) (pp. 1-6). IEEE.

[5] Kim, K. J. (2003). Financial time series forecasting using support vector machines. Neurocomputing, 55(1–2), 307-319.

[6] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
[7] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).
[8] Li, K., Shen, N., Kang, Y., Chen, H., Wang, Y., & He, S. (2021). Livestock product price forecasting method based on heterogeneous GRU neural network and energy decomposition. IEEE Access, 9, 158322-158330.
[9] Cifci, A. (2025). Interpretable Prediction of a Decentralized Smart Grid Based on Machine Learning and Explainable Artificial Intelligence. IEEE Access, 13, 36285-36305.
[10] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.
[11] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.
[12] Meena, K., & Chaitra, B. (2024). A Novel Framework Using Deep Learning Techniques for Ragi Price Prediction in Karnataka. IEEE Access, 12, 136103-136119.
[13] Halim, M. P., Yudistira, N., & Dewi, C. (2024). Multicommodity Prices Prediction Using Multivariate Data-Driven Modeling: Indonesia Case. IEEE Transactions on Computational Social Systems, 11(6), 8106-8117.
[14] Wang, C., & Gao, Q. (2018). High and low prices prediction of soybean futures with LSTM neural network. In 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS) (pp. 140-143). IEEE.
[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
[16] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 12, pp. 11106-11115).
[17] Li, X., Li, L., Zhang, K., Chen, X., Feng, T., Zhao, Y., & Yin, S. (2025). Multivariate Correlation Self-Distillation Transformer for Time Series Forecasting With Incomplete Data. IEEE Transactions on Industrial Informatics, 1-11.
[18] Lee, T.-W., Teisseyre, P., & Lee, J. (2023). Effective Exploitation of Macroeconomic Indicators for Stock Direction Classification Using the Multimodal Fusion Transformer. IEEE Access, 11, 10275-10287.
[19] Babu, C. N., & Reddy, B. E. (2014). A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data. Applied Soft Computing, 23, 27-38.
[20] Yu, L., Wang, S., & Lai, K. K. (2008). Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. Energy Economics, 30(5), 2623-2635.
[21] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems 30 (pp. 4765-4774).