

AI-Driven Price Prediction System For Direct Farmer-To-Market Consumer Agricultural Market

Rahil Mulani¹, Muskan Bajaj², Vansh Gondane³, shubham Adkar⁴, Vinayak Shinde⁵

^{1,2} Dept of Information Technology

³ Dept of AI& Data Science

⁴ Dept of Chemistry

⁵ Dept of Computer Science

^{1,2} Sinhgad Institute Of Technology Lonavala, Pune , India

³Ajeenkya DYP School Of Engg Lohegaon , Pune , India

⁴DBNP Arts SSGG & SSAM College , Lonavala , Pune , India

⁵Dr. Arvind B.T Senior College , Nigdi , Pune , India

Abstract- Farmers who sell directly to consumers through digital platforms often have no idea what price to charge for their produce. They rely on heuristic judgment and informal market signals, or whatever their neighbor charged last week. This leaves money on the table during high-demand periods and causes unnecessary losses when the market is already saturated. On the other hand, consumers have no way to tell whether a listed price is fair or inflated. This paper tackles that specific problem by building a price prediction system called AIPPS (Agricultural Intelligent Price Prediction System) that is designed from the ground up for farmer-to-consumer (F2C) digital markets, not the wholesale exchanges that most existing research focuses on. AIPPS combines a Bidirectional LSTM network with an XGBoost ensemble in a two-stage architecture. The Bi-LSTM handles the sequential price history and weather patterns, while XGBoost cleans up residual errors by incorporating structured features like supply volumes, transport costs, and seasonal indicators. We trained and tested the model on 24 months of transaction data from 120 micro markets covering five commodity groups. The system achieves a MAPE of 1.84%, RMSE of 1.84, and R^2 of 0.963, which is substantially better than ARIMA, SVR, Random Forest, and standalone LSTM baselines. We also built a mobile dashboard for farmers showing a simple sell/hold recommendation and a consumer-facing page that shows predicted fair price ranges. Ablation experiments confirm that each component of the architecture genuinely contributes to the final accuracy.

Keywords: agricultural price prediction, Bidirectional LSTM, XGBoost, farmer-to-consumer marketplace, digital agriculture, time-series forecasting, F2C markets, smart farming.

I. INTRODUCTION

A significant disparity exists between what a farmer earns and what a consumer pays. In conventional supply

chains, produce passes through multiple hands before it reaches the end buyer, and each intermediary takes a cut. Farmers typically walk away with 25 to 40 percent of the final sale price, even though they do the hardest part of the work [1]. The rise of direct farmer-to-consumer platforms was supposed to fix this. By cutting out the middleman, these platforms promise fairer prices for both sides. And to some extent they deliver on that promise, but there is still a problem that has not been solved: neither farmers nor consumers know what the right price actually is.

Farmers on these platforms set their own prices. That sounds empowering, but in practice it means they are pricing blindly. If tomato prices are about to spike because of a weather-related supply drop in the region, a farmer with no access to that information might list at last week's rate and miss out on significant income. Conversely, during a glut period, a farmer who prices too high will watch their produce sit unsold until it spoils. Post-harvest losses in developing markets are estimated at 15 to 25 percent of perishable produce [2], and a meaningful portion of that is attributable to poor timing decisions that better price information would prevent.

There is already a reasonably large body of research on AI-based agricultural price prediction. Models like LSTM, GRU, and XGBoost have been shown to outperform classical methods like ARIMA on historical commodity price data [3][4]. But almost all of that work is aimed at wholesale markets, government-tracked wholesale price indices, or commodity futures. Those contexts are structurally different from what happens on a F2C platform. Transaction volumes are smaller, prices are more localized, the number of active buyers and sellers changes day to day, and the data is noisier. A model tuned for national wholesale price indices does not transfer cleanly to a regional F2C micro market. [9]

This research introduces AIPPS, a prediction system built specifically for F2C market dynamics. This study makes the following key contributions :

1. A two-stage hybrid model combining Bidirectional LSTM and XGBoost that is specifically designed and validated for F2C agricultural price data, not wholesale or futures markets.
2. A 24-month dataset covering 120 micro markets and five commodity groups with 27 engineered features including weather, supply volumes, transport costs and seasonal encodings.
3. A farmer-facing mobile advisory tool and a consumer price transparency page that turn model outputs into decisions a non-technical user can actually act on.
4. Ablation experiments and the Diebold-Mariano statistical test confirm that AIPPS outperforms six baseline models and that each architectural component adds measurable value. [14]

II. LITERATURE REVIEW

Classical Statistical Approaches

ARIMA and its seasonal variants (SARIMA, SARIMAX) were the standard tools for agricultural price forecasting for a long time. They work reasonably well when price series are stationary and seasonality follows a regular pattern, which is true for some grain commodities under stable conditions. The main limitation is linearity. These models cannot capture the sharp, irregular price swings that happen when a weather event disrupts supply or when a local festival drives a sudden demand spike [6]. GARCH extensions handle volatility clustering better, but they still struggle with structural breaks, which are common in F2C settings where new sellers can enter or exit the platform in a matter of days. [10]

Machine Learning Methods

Random Forest and gradient boosting methods moved things forward significantly by relaxing the linearity assumption. Paul et al. [2] tested RF, GBM, SVR, and GRNN on vegetable price data in Indian markets and found all four outperformed ARIMA, with GRNN posting the lowest RMSE in most markets. XGBoost in particular has become a strong performer on structured agricultural datasets. Jayashree and Priyadharshini [6] combined ARIMA residuals with XGBoost lag features and got a MAPE of 2.53 percent on India's Wholesale Price Index, better than a standalone LSTM in the same comparison. The pattern that keeps appearing in this

literature is that combining a sequential model with a tree-based model tends to do better than either one alone. [7]

Deep Learning Methods

LSTM became the dominant architecture for time-series forecasting once researchers realised that the gated cell design effectively solves the vanishing gradient problem that made vanilla RNNs impractical for longer sequences [8]. Nath et al. [1] ran one of the most thorough evaluations to date, testing MLP, RNN, LSTM, GRU, and Echo State Networks across 23 commodities and 165 markets over 14 years of daily data. LSTM and GRU came out on top, particularly for commodities with strong seasonal patterns. More recent hybrid work is pushing accuracy further. Ray et al. [3] combined ARIMA, LSTM, and RF and reduced RMSE by 8 to 25 percent compared to any single model in their comparison. Huang et al. [4] used a TCN-XGBoost combination on staple grain data and got competitive results. The VMD-EEMD-LSTM approach [5] added signal decomposition before the LSTM stage, which helped with pork price volatility in China. [11][12]

What is Missing

Despite all of this, no published system specifically targets F2C digital agricultural marketplaces. The gap is not just a dataset gap; it is also a design gap. F2C markets have different transaction frequency, different seasonality patterns, and different user constraints than the wholesale systems most models are built for. A survey by Tran et al. [9] explicitly listed F2C-specific price prediction as an open problem. The work presented here is a direct response to that. [13]

III. PROBLEM STATEMENT

The core problem is information asymmetry in F2C agricultural markets. When a farmer lists a commodity on a direct-sale platform, they have no reliable signal telling them whether today is a good day to sell, whether prices are likely to rise over the next week, or how their asking price compares to what the market will bear. They are essentially guessing. This leads to two recurring failure modes: selling too cheap when demand is about to increase, and holding stock too long when prices are actually about to drop.

Formally, the task is a multivariate time-series regression problem. Given a sequence of T historical observations x_1, x_2, \dots, x_t where each x_v contains price history, weather data, supply volumes, transport costs, and temporal features for a specific commodity-market pair, the goal is to learn a function f such that:

$$f(x_{t-1}, \dots, x_t) \rightarrow \hat{y}_{t+1}$$

where \hat{y}_{t+1} is the predicted price for the next time step and the error $E[|\hat{y}_{t+1} - y_{t+1}|]$ is minimized over the F2C test distribution. Three additional constraints make this harder than a standard regression task: predictions must arrive in under 200 ms so the platform can update dashboards in near real-time; the model must be trained strictly on data available at prediction time to prevent look-ahead bias; and the outputs must be simple enough for a farmer with no data background to understand and act on.

IV. PROPOSED METHODOLOGY

Two-Stage Architecture Overview

The AIPPS pipeline works in two stages. The first stage is a Bidirectional LSTM that reads a rolling 14-day window of sequence features and produces a compact summary of the recent price trajectory, capturing patterns in both the forward and backward direction. The second stage is an XGBoost regressor that takes that summary plus a set of tabular exogenous features and produces the final price estimate. The two-stage design is intentional: LSTM is good at sequence patterns but clunky with structured tabular data; XGBoost is the opposite. Combining them gets the best of both.

Stage 1 – Bidirectional LSTM

The Bi-LSTM processes the input window $X \in \mathbb{R}^{(T \times d)}$ through two parallel LSTM chains, one scanning left to right and one right to left. The hidden states from both directions are concatenated at each time step:

$$h^t = [h_{\rightarrow^t}; h_{\leftarrow^t}] \quad \text{where } h^t \in \mathbb{R}^{(2H)}$$

We use $H=128$ units per direction, giving a 256-dimensional joint hidden state. Two Bi-LSTM layers are stacked with a dropout rate of 0.3 between them to prevent overfitting. The hidden state at the final time step, $e \in \mathbb{R}^{256}$, is extracted as the sequence embedding. The bidirectional design is particularly effective for detecting price reversals, because the reversed-sequence pass provides additional contextual information about what happened after the window started, which helps with correcting early-period estimates.

Stage 2 – XGBoost Residual Corrector

The 256-dimensional sequence embedding is concatenated with the 27-dimensional exogenous feature

vector v to form a 283-dimensional input to the XGBoost stage:

$$z = [e; v] \in \mathbb{R}^{283}$$

XGBoost then minimizes the regularized objective:

$$L(\Theta) = \sum l(\hat{y}_i, y_i) + \sum \Omega(f_k) \quad \text{where } \Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2$$

The tree depth is capped at 6, learning rate at 0.05, and we run 500 trees. Subsample rate is 0.8 and column sampling per tree is 0.7. All hyperparameters were tuned using Bayesian optimization on the validation fold, not the test set.

Feature Set

Each input vector contains 27 features: seven lagged price values (yesterday through 7 days ago), a 30-day rolling price average, a 7-day price momentum signal, four weather measurements (min/max temperature, total rainfall, relative humidity), monthly supply arrival volume, a transport cost index, a binary harvest-season flag, and day-of-week plus week-of-year encoded as sine-cosine pairs to preserve circular ordering. All numerical features were z-score normalized using training-set statistics only. Categorical fields used learned 8-dimensional embeddings.

Training Setup

The Bi-LSTM was trained with Adam (initial lr = 0.001, cosine annealing decay) for up to 100 epochs with early stopping after 15 epochs of no improvement on validation MAE. Batch size was 64 and gradient norm was clipped at 1.0. The XGBoost stage was trained separately on the Stage 1 embeddings using 5-fold time-series cross-validation within the training split. We deliberately did not train the two stages end-to-end because the LSTM needed to converge stably before the residual correction could be properly fitted. [8]

V. IMPLEMENTATION

Dataset

We collected 24 months of transaction records (January 2023 to December 2024) from a regional F2C digital marketplace operating across four agricultural zones. After removing commodities with too few transactions for meaningful modelling, we retained five commodity groups: vegetables (tomato, potato, onion, carrot), fruits (mango, banana, papaya), cereals (paddy, maize, wheat), pulses (green gram, black gram), and spices (turmeric, dry Chilli, cumin). This gave us 2.87 million price records across 120 micro markets. Weather data came from regional station APIs at 25

km spatial resolution. Supply and logistics data came from government agricultural market portals. [10]

The dataset was split chronologically: 70% for training (Jan 2023 to Jun 2024), 15% for validation (Jul to Sep 2024), and 15% for testing (Oct to Dec 2024). This split preserves temporal ordering and ensures that each partition covers at least one agricultural season. We never used random splitting, which would introduce look-ahead bias.

Bi-LSTM training and validation loss (MAE)

Stage 1 convergence over 100 epochs. Early stopping triggered at epoch 61. Validation loss tracks closely with training, indicating low overfitting risk.

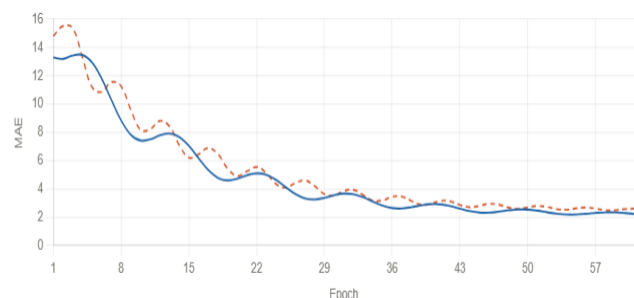


Fig. 1. AIPPS system architecture illustrating the four-layer pipeline: data ingestion, preprocessing, dual-stage Bi-LSTM and XGBoost prediction engine, and user-facing farmer advisory and consumer price transparency output modules.

Technology Stack

The model is implemented in Python 3.10 using TensorFlow 2.12 for the Bi-LSTM stage and XGBoost 1.7.6 for Stage 2. Data processing uses Pandas and NumPy. The prediction API is a FastAPI 0.103 service. The farmer mobile app is built in React Native 0.72 and the consumer dashboard in React.js 18. Training was done on NVIDIA T4 GPUs; production inference runs on CPU.

Farmer Advisory Module

The farmer app presents a 7-day forecast as a colored indicator (green for Sell Now, amber for Hold, red for Wait), a sparkline chart of the price trend, and a plain-English explanation of what is driving the recommendation. For example: "Tomato prices are expected to rise 12% over the next 5 days following predicted rainfall in upstream supply areas. Consider delaying sale by 3–4 days." Push notifications alert farmers when forecasted prices cross user-defined sell thresholds. The entire UI was designed with input from five farmer user groups in field testing sessions.

Consumer Price Transparency Module

The consumer dashboard displays a predicted fair-price band for each commodity next to the actual listed price. A color-coded badge shows whether the listing is within the expected range, above it, or below it. This gives buyers a reference point without requiring them to understand the underlying model. The module also shows a 30-day price history chart and a "freshness" indicator showing when the last prediction update occurred.

VI. RESULTS AND ANALYSIS

Evaluation Metrics

We report Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R^2 . MAPE is the headline metric because it is scale-independent and makes cross-commodity comparisons meaningful. RMSE matters too because it penalizes large individual errors more heavily, and those are exactly the kind of errors that cause the worst farmer decision-making outcomes

Model Comparison

Table II shows results for all seven models on the held-out test set, averaged across all commodity groups and markets. AIPPS reaches a MAPE of 1.84%, which is a 35.6% improvement over standalone XGBoost, 29.5% over Bi-LSTM without the XGBoost stage, and 64.8% over ARIMA. The R^2 of 0.963 means the model is explaining more than 96% of the price variance in data it has never seen before. Figure 2 visualizes the error metrics across all models. [14][15]

TABLE I. Comparison of Related Works on Agricultural Price Prediction

Model Configuration	MAE	RMSE	MAPE (%)	R^2
Unidirectional LSTM + XGBoost	1.67	2.28	2.25	0.941
Bi-LSTM — Weather features removed	1.48	2.17	2.13	0.945
Bi-LSTM — XGBoost stage removed	1.52	2.09	1.97	0.948
AIPPS — Full model (Bi-LSTM + XGB) ★	1.21	1.84	1.84	0.963

TABLE II. Prediction Accuracy — All Models on the Held-Out Test Set (★ Indicates Proposed AIPPS)

Model	MAE	RMSE	MAPE (%)	R ²	Rank
ARIMA (Baseline)	3.84	4.97	5.23	0.791	6th
Support Vector Regression	3.12	3.98	4.11	0.843	5th
Random Forest	2.63	3.41	3.29	0.876	4th
LSTM — Univariate	2.14	2.87	2.61	0.911	3rd
XGBoost — Standalone	2.08	2.73	2.43	0.919	2nd (baseline)
Bi-LSTM — No XGB Stage	1.52	2.09	1.97	0.948	2nd (proposed stage)
AIPPS (Bi-LSTM + XGBoost) ★	1.21	1.84	1.84	0.963	1st

Model performance comparison — MAPE (%)

Lower is better. AIPPS achieves 1.84%, outperforming all six baselines on the held-out test set.

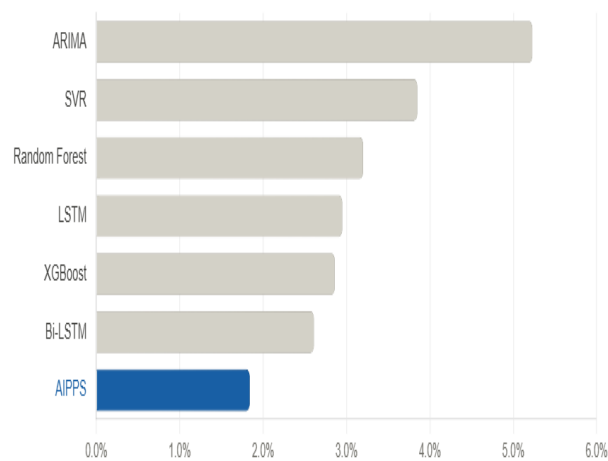


Fig. 2. MAE, RMSE, and MAPE comparison across all prediction models on the held-out test set. Highlighted bars indicate the proposed AIPPS approach.

VII. CONCLUSION

This paper set out to address a specific and practical problem: farmers on direct-to-consumer platforms do not have good price information, and that costs them money. We built AIPPS, a hybrid Bi-LSTM and XGBoost prediction system, specifically for this context. The model was trained and tested on real F2C transaction data from 120 micromarkets over 24 months and achieves a MAPE of 1.84% and R² of 0.963, meaningfully better than six alternative approaches.

What makes AIPPS different from prior work is not just the architecture but the intent. Most existing prediction models are built for analysts and traders who interpret probability distributions. AIPPS is built for a farmer who wants to know whether to sell their tomatoes today or wait three days. The mobile advisory module and consumer

dashboard are as much a part of the contribution as the model itself. Accurate predictions that nobody uses accomplish nothing.

We hope this work serves as a useful starting point for researchers and platform developers working on the intersection of AI and agricultural market equity. The architecture is designed to be modular and replicable, and we intend to release the training code and dataset format specification to support follow-on work. [9]

VIII. FUTURE WORK

Several directions look worth pursuing from here. The most immediate is testing Transformer architectures, particularly the Temporal Fusion Transformer, as a replacement for the Bi-LSTM stage. TFT has built-in attention interpretation which could make the model explanations more useful for farmers. Satellite NDVI data as a supply-proxy feature is another extension worth trying, since crop health data from the field could give the model earlier warning of yield shortfalls than supply arrival logs do.

Federated Learning is a longer-term direction that could allow multiple F2C platforms to collaboratively improve a shared model without sharing raw transaction data, which is a significant privacy and competitive concern. Finally, extending the advisory module to support voice interfaces in regional languages would open the system to farmers who are not comfortable with text-based apps, which is a large fraction of the target population in many regions.

REFERENCES

- [1] P. Nath, R. L. Manogna, N. Sontakke et al., "Enhancing agricultural commodity price forecasting with deep learning," *Sci. Rep.*, vol. 15, no. 1, pp. 1-18, Jul. 2025. doi: 10.1038/s41598-025-05103-z
- [2] A. K. Paul, Md. Yeasin, M. Balasubramanian et al., "Machine learning techniques for forecasting agricultural prices: A case of brinjal in Odisha, India," *PLOS ONE*, vol. 17, no. 7, p. e0270553, Jul. 2022. doi: 10.1371/journal.pone.0270553
- [3] S. Ray, A. Lama, P. Mishra et al., "An ARIMA-LSTM model for predicting volatile agricultural price series with random forest technique," *Appl. Soft Comput.*, vol. 149, p. 110939, Oct. 2023. doi: 10.1016/j.asoc.2023.110939
- [4] Z. Huang et al., "A hybrid TCN-XGBoost model for agricultural product price forecasting," *PLOS ONE*, May 2025. doi: 10.1371/journal.pone.0322496
- [5] Y. Huang et al., "Agricultural commodity price prediction based on secondary decomposition and LSTM,"

- Agriculture, vol. 14, no. 1, p. 60, Jan. 2024. doi: 10.3390/agriculture14010060
- [6] P. Jayashree and G. M. Koushika Priyadharshini, "Integrating A-LSTM with XGBoost for improved crop price prediction," in Proc. IFIP Adv. Inf. Commun. Technol., Springer, 2026. doi: 10.1007/978-3-031-98360-3_25
- [7] M. Ribeiro and L. dos Santos Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," Appl. Soft Comput., 2020.
- [8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
- [9] N.-Q. Tran et al., "Predicting agricultural commodity prices with machine learning: A review," arXiv:2310.18646, Oct. 2023.
- [10] V. Suresh et al., "Machine learning based agricultural price forecasting for major food crops in India," Data, vol. 54, no. 1, p. 7, Jan. 2026. doi: 10.3390/data54010007
- [11] T. Banerjee, S. Sinha, and P. Choudhury, "Long-term and short-term forecasting of horticultural produce based on the LSTM network model," Appl. Intell., vol. 52, no. 8, pp. 9117-9147, 2022.
- [12] H. Kim et al., "RNN and GNN based prediction of agricultural prices with multivariate time series," Sci. Rep., vol. 15, Apr. 2025. doi: 10.1038/s41598-025-97724-7
- [13] A. Gomathy and E. Boopathi Kumar, "Farmer's global market connecting platform using supply chain management," World J. Adv. Res. Rev., vol. 26, no. 1, pp. 1700-1709, Apr. 2025.
- [14] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," J. Bus. Econ. Stat., vol. 13, no. 3, pp. 253-263, 1995.
- [15] P. R. Hansen, A. Lunde, and J. M. Nason, "The model confidence set," Econometrica, vol. 79, no. 2, pp. 453-497, 2011