ISSN: **3139-0862**

AgriTrade Analytics

An Evidence-Based Crop Bidding Marketplace with Real Mandi Price Decision Support for Farmers

Gayathri N[#1], Nayana Rao S[#2], Shreyas P N[*3], Soujanya U[*4], Sindhu M N[*5], Saniya Banu P Kanakwad[*6]

[#]*Assistant Professor, Department of Computer Science and Engineering, City Engineering College, Bengaluru, Karnataka, India*

[*]*Undergraduate Scholar, Department of Computer Science and Engineering, City Engineering College, Bengaluru, Karnataka, India*

[1]gayathrin@cityengineeringcollege.ac.in, [2]nayana_rao@cityengineeringcollege.ac.in, [3]narasimhashreyas2580@gmail.com, [4]soujanyau85@gmail.com, [5]sindhumn09@gmail.com, [6]saniyabanupkanakwad@gmail.com

Abstract

Farmers often decide when and where to sell on the strength of rumor, and most price forecasting tools never say how far a predicted price can be trusted. AgriTrade Analytics was built to tackle both problems. It is a working marketplace in which farmers list crops, buyers place bids, and an accepted bid moves on to payment, shipment tracking and delivery confirmation. A separate data science service supports every one of these stages. It forecasts prices from marketplace records and from official mandi prices published by the Government of India, compares nine forecasting methods using rolling origin validation, and names a winner only when that method beats the naive last price forecast after correction for multiple comparisons. Each forecast carries a conformal prediction interval whose real coverage is measured. A decision card then turns the results into plain advice on price, market, profit and timing, together with a confidence level and a list of limitations. The platform also groups buyers, flags unusual bids, reports how much of its data is synthetic, and answers questions through an assistant that names the source of every answer. On simulated data with known answers, false winners fell from four in forty to none. On imported mandi history for onion, potato and wheat, no model was significantly better than the naive forecast, so the system works as decision support and not as a price predictor. The assistant routed 69 percent of held out questions correctly.

Keywords—Agricultural marketplace; price forecasting; mandi prices; rolling origin validation; conformal prediction; decision support; shipment traceability; buyer segmentation

I. INTRODUCTION

Right after a harvest, the farmer is usually the person in the supply chain who knows least about what the crop is worth. Which mandi, the regulated wholesale market, will pay more this week? Will prices recover once the flood of new produce has passed? Is it wiser to sell today or to hold on? Traders and middlemen tend to hold these answers and have little reason to share them, so the people who carry the risk of production end up with the thinnest information.

The tools available today meet only part of that need. Public price portals report what markets paid yesterday, which is useful but looks backward. Research prototypes do forecast prices, yet most of them end with a table of accuracy figures. They seldom tell a farmer how far a number can be relied upon, or whether simply repeating the last observed price would have worked just as well [1]. Neither kind of tool is present at the moment a sale is actually made.

AgriTrade Analytics places decision support at that moment. Farmers list their crops, buyers compete for them, and an analytics service runs beside the marketplace to help both sides choose. One rule guided the whole project: nothing is presented as a finding unless a test supports it. A model counts as better only if it beats the naive last price forecast under a corrected significance test, every forecast comes with an interval whose true coverage we measure, and when a price series is too erratic to predict, the system says so. The main contributions are listed below.

- A complete marketplace with atomic bid acceptance, payment and shipment tracking, including a quick response (QR) code, a status lifecycle, alerts and delivery verification by a one-time code. The Node.js and Python services share a single login.
- A forecast laboratory that compares nine methods by rolling origin validation, tests each against the naive baseline with the Diebold-Mariano test and Holm correction, builds conformal intervals with measured coverage, and checks whether a series can be predicted at all.
- A price series free of look-ahead leakage, together with a provenance meter that reports the synthetic share of the data and lowers confidence when that share is high.
- A decision card that shows conservative, expected and optimistic profit with its limitations printed beside it, plus buyer segmentation, bid anomaly detection, an assistant that labels its sources and has a held-out evaluation, and a companion planting guide checked against published evidence.

ISSN: **3139-0862**

- An evaluation on real imported mandi history, supported by an automated test suite and a continuous integration pipeline.

II. Related Work

Most earlier work on agricultural prices asks a single question: which model predicts best? Paul et al. compared the autoregressive integrated moving average (ARIMA) method with neural, kernel and tree-based learners on brinjal prices from seventeen markets in Odisha, and backed their ranking with significance tests [2]. Kasa et al. embedded a long short-term memory (LSTM) network in a web application [3]. Sindhur et al. combined a random forest for crop suitability, an LSTM for prices and a Kannada voice interface, and left cost and profit modeling for later [4]. In all three, the work stops at the forecast and never reaches an actual sale. Customer behavior, on the other hand, has mostly been studied outside agriculture. Wong et al. found that recency, frequency and monetary value (RFM) combine well with clustering for e-commerce customers [5], and we carry the same idea over to bidders. Table I summarizes these studies.

TABLE I. Summary of Related Work

Ref.	Contribution	Left open
[1]	Statistical versus machine learning forecasts; the missing simple benchmark	Evaluation study; no deployed system
[2]	ARIMA against four machine learning models on brinjal prices, with significance tests	No profit figure
[3]	LSTM price prediction inside a web application	No market comparison, no trading
[4]	Crop suitability plus price forecasting with a voice interface	Cost and profit not built
[5]	Recency, frequency and monetary value with clustering	Retail only; not tried in bidding

A second body of work deals with judging forecasts fairly. Tashman argued for rolling origin evaluation in place of a single train and test split [6]. Hyndman and Koehler proposed the mean absolute scaled error (MASE), which measures a method directly against the naive forecast [7]. To check whether one forecast is truly better than another, the Diebold-Mariano test is widely used [8], together with the small-sample correction of Harvey et al. [9], while the Holm procedure keeps false findings in check when many models are compared [10]. Conformal prediction produces intervals without assuming any error distribution [11], although its guarantee is weaker for time series [12]. These tools normally appear separately in the literature. Here we join them in one application that farmers can actually use.

III. System Design

A. Architecture

The platform has three parts that can be deployed independently: a React web client, a Node.js marketplace application programming interface (API), and a Python analytics service built with FastAPI. All three read the same MongoDB database (Fig. 1). Node.js suits the marketplace chores of uploads, payments and shipments, while Python is the natural home for forecasting, clustering and explanation because of libraries such as pandas, statsmodels and scikit-learn [13]. Keeping the two apart has a practical benefit as well: a slow forecast can never hold up a farmer who is accepting a bid. The analytics service only reads the marketplace collections. The one thing it writes is the imported mandi prices.

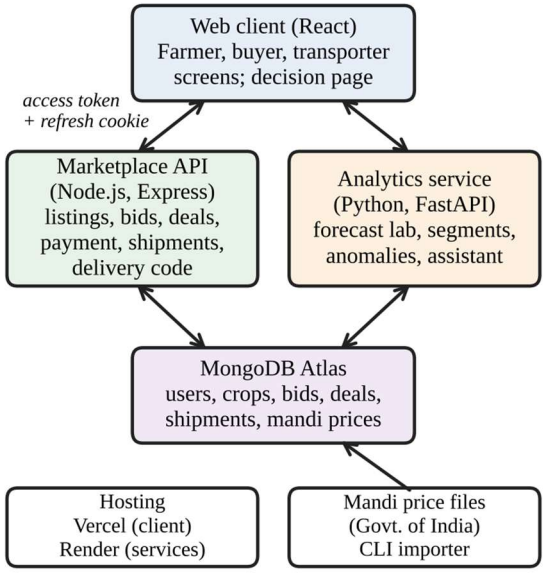

Fig. 1. System architecture of AgriTrade Analytics

B. Sessions and Shared Identity

We did not want people to log in twice. At sign-in, the Node server issues a JSON Web Token (JWT) signed with HS256. The token lasts 30 minutes and the client keeps it in memory only. A rotating httpOnly refresh cookie keeps the person signed in. If an old cookie is ever replayed, the whole login chain is revoked, and the authentication routes also demand a custom request header as a defense against cross-site request forgery (CSRF). On every route, the analytics service verifies the same token with the same secret and fixes the permitted algorithm, so one account works across both services. If the secret is missing in production, the analytics service returns a 503 response instead of letting requests through; in other words, it fails closed.

C. Deal Lifecycle and Shipment Safety

A listing begins with the farmer's asking price. Buyers place bids and the farmer accepts one. Acceptance is a single conditional database update that succeeds only while the listing is still open, so two clicks, or two requests arriving together, cannot create two deals. A concurrency test checks this. Payment goes through Razorpay when it is configured, and in that case the manual Unified Payments Interface (UPI) route labeled “I have paid” is switched off, so it cannot be used to skip real payment.

Each accepted deal receives a shipment with an identifier of the form SHP-YYYYMMDD-XXXXXXXX, whose random characters avoid look-alike symbols, and a QR code that holds nothing but a link placed behind the login. Anyone who is not the farmer, the buyer or the assigned transporter gets the same “not found” reply as for an identifier that does not exist, so identifiers cannot be guessed. A Transporter role records pickup and transit checkpoints in an append-only event log, and every entry stores the time, role and name. The lifecycle has ten statuses: created, ready for pickup, picked up, in transit,

ISSN: **3139-0862**

delayed, out for delivery, delivered, delivery failed, damage reported and under investigation. A transition table defines who may make each move. Skipping a step or acting in the wrong role is refused, and “delivered” can be reached only through the buyer's delivery code.

On receipt, the buyer enters a one-time code and reports the quantity received and its condition. Photos are checked for content and size. The code is stored only as a hash-based message authentication code (HMAC), expires after 60 minutes, works once, and locks after five wrong attempts. It is sent by e-mail through Brevo or Resend when either is configured, and is shown directly only in a clearly labeled development mode. While a dispute is open, the payout is held back. Rule-based alerts cover delay, quantity mismatch, damage and failed verification. Each alert states its reason and the numbers behind it, only one alert stays open for a given cause, repeated problems are escalated instead of re-sent, and an alert closes by itself once its cause clears. All of this makes a delivery traceable and backed by evidence, but it does not make it safe. The code proves control of the buyer's inbox rather than physical receipt of the goods, and the server cannot verify positions reported from a carrier's phone.

D. Security and Implementation

An early version was reviewed against common web security checklists, and the findings were fixed. Analytics routes return accurate status codes and reveal no internal details. Cross-origin access is limited to an allow-list, and analytics requests are rate-limited for each user. Uploads must be PNG, JPEG or WebP images of at most 400 kilobytes, and the server enforces this. Passwords are hashed with bcrypt. A sign-up code allows five wrong guesses and is then destroyed. Analytics shown across the whole marketplace replace other users' identities with pseudonyms. Frequently used results are cached, and the cache is cleared whenever listings, bids or deals change.

The client uses React with Recharts. The marketplace API is written with Express and Mongoose, and the analytics service with FastAPI, pandas, statsmodels, scikit-learn and Prophet. On every push, GitHub Actions runs four jobs: the backend, the analytics service, the client with its production build, and a Playwright browser suite. The client is hosted on Vercel and both services on Render. Secrets live in environment variables and are never stored in the repository.

IV. METHODOLOGY

A. Data and Preparation

The analytics service draws on two sources. The first is the marketplace itself, which we filled with generated records during development; the application reports how large the synthetic share is. The second is the Agricultural Marketing Information Network (AGMARKNET) of the Government of India [14]. A command-line importer loads it in three steps: it scans the files, performs a dry run that applies every cleaning rule, and only then writes to the database under a unique key, so repeating an import creates no duplicates. A row that fails a check is dropped, counted and reported. Dates must parse and fall within a sensible range, and duplicates for the same commodity, market and date are removed. Units are checked as well, because a price per quintal and a price per kilogram differ by a factor of one hundred, and prices that make no sense for their unit are removed. Farmers' direct-sale markets, which charge close to retail prices, are flagged and kept out of the wholesale figures.

Bidding sets a trap for anyone who forecasts. The highest bid keeps rising until the farmer accepts, so using it as the historical price lets later information leak into earlier rows. We train on the asking price instead, since it is fixed when the listing is created, and a test confirms that the series does not change when bids change. Part of the marketplace data is generated, so a forecast built mostly on it would describe the generator and not a market. A provenance meter therefore calculates the synthetic share from the data, displays it, and caps the confidence of the decision card when the share is high. A forecast learned from history older than 45 days is withheld, and today's live median price is shown in its place.

B. Forecast Laboratory

For each commodity the laboratory trains nine candidates: naive, seasonal naive (weekly and yearly), drift, moving average, mean reversion, ridge regression, gradient boosting and ARIMA. All of them are scored with rolling origin validation [6]. At every origin a model is fitted on earlier observations only, predicts the next period, and has its error recorded before the origin moves forward. Every model sees the same origins, which keeps the comparison fair. Shuffled splits were avoided because they would let a model see the future (Fig. 2). Each model is summarized by four measures: mean absolute error (MAE) and root mean square error (RMSE) in rupees, symmetric mean absolute percentage error (sMAPE), and MASE [7]. MASE divides the error of a model by the error of the naive forecast, as in (1), so a value below one means the model beats naive.

$$MASE = MAE \,/\, MAE_{\text{naive}} \tag{1}$$

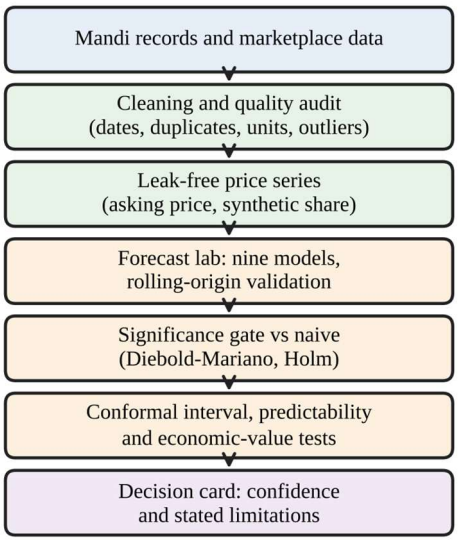

Fig. 2. Analytics pipeline from raw records to the decision card

The nine candidates span a range of ideas. Naive repeats the last price, and the two seasonal versions repeat the value from a week or a year earlier. Drift extends the trend of the last 60 days, the moving average smooths the last seven days, and mean reversion assumes that prices drift back towards their 90-day average. Ridge regression and gradient boosting learn from lagged price changes, and ARIMA(1,1,1) serves as the

ISSN: **3139-0862**

classical time series baseline. Setting simple rules next to learned models shows whether the added complexity buys anything.

The test windows are built so that no model can see its own future. A test with a spy model confirms that each model receives only the values recorded before the origin. Seasonality is computed only to describe the series and is never fed to a model as an input.

C. Significance Gate

A lower error can simply be luck. For every model we take the loss difference against naive at each test point and apply the Diebold-Mariano test [8] with the Harvey correction [9]. In (2), D is the mean loss difference and $SE(D)$ is its standard error, estimated so that it allows for correlation between neighboring errors.

$$DM = D / SE(D) \tag{2}$$

All nine models are compared with the same baseline, so a few will look significant by chance alone. For that reason the p-values are adjusted with the Holm procedure [10]. Let $p_{(i)}$ be the i-th smallest of m p-values and α the chosen level. Hypothesis i is rejected only when (3) holds and every smaller p-value was also rejected. A model is recommended only if it passes this gate; otherwise the naive forecast is shown, with an explanation.

$$p_{(i)} \le \alpha / (m - i + 1) \tag{3}$$

D. Intervals, Predictability and Economic Value

For the champion model, the absolute errors saved during validation give a conformal interval [11]. If q is the quantile of those errors at level $\lceil (n + 1)(1 - \alpha) \rceil / n$, the interval around a forecast $\hat{y}$ is (4). With $\alpha = 0.10$ this is a 90 percent interval.

$$I = \hat{y} \pm q \tag{4}$$

A time series does not satisfy exchangeability, so the guarantee holds only approximately [12]. The system therefore counts how many held-out points actually fall inside the interval and shows that figure next to the nominal level. When the conformal step cannot run, an ordinary least squares interval is used rather than a band of zero width. Before trusting any model, we also ask whether the series contains a pattern at all, using the Ljung-Box test and the variance-ratio test. If a series behaves like a random walk, failing to beat naive is the expected outcome, and the system reports it that way. Finally, an economic-value test compares a farmer who waits whenever the model predicts a rise with one who always sells immediately, and gives a block bootstrap interval for the difference. Storage, spoilage and interest are not deducted, and the result says so.

E. Decision Card

The decision card turns these results into advice. It shows the predicted price and demand, compares markets using the imported mandi records under stated cost assumptions, and suggests whether to wait or sell. Beside it sit a confidence level with its reasons, such as a short history, a high synthetic share or a weak result against naive, and a list of limitations. Net profit appears as a conservative, expected and optimistic band, along with a check on whether the best market still wins under pessimistic assumptions. The headline figure is the expected case, and thin margins are pointed out. The card is built to be safe by default: when the evidence is weak, it says so.

F. Segmentation, Anomalies and the Assistant

Each buyer is described by recency, frequency and monetary value. These values are log-transformed so that one very large buyer cannot dominate, and the buyers are then grouped with K-Means [13]. The number of groups is chosen by silhouette score [13] and checked for stability by bootstrap resampling. To find unusual bids we use the ratio of a bid to the asking price of its listing, because the same rupee amount can be normal for one crop and extreme for another. Three detectors examine this ratio: a Z-score rule, an interquartile range rule and an Isolation Forest [15]. A bid enters the high-confidence tier only when all three agree.

The assistant sends each question through a routing function that picks a route by intent. Questions about prices, demand and where to sell are answered directly by the analytics models, without any language model. General farming questions go to an optional language model. Every answer names its source, and any price figure that appears in a language model answer is withheld. The idea is close to retrieval-augmented generation, although retrieval is not used in the strict sense. Shapley additive explanations (SHAP) values explain the random forest price model.

G. Evidence-Checked Companion Planting

The guide works from a catalogue of 116 crops but shows only the 23 pairings that have published support: 10 supported, 11 potentially suitable and 2 unsuitable. The status of a pairing is not chosen by hand. Written rules derive it from the pairing's registered sources; for example, “supported” needs at least two primary studies that report a consistent benefit, and a test fails if a stored status differs from the derived one. These rules downgraded two pairings. Every other pair is reported as having insufficient evidence, because intercropping results depend on species, density, soil, climate and management.

V. Results and Discussion

A. Testing

Testing was carried out at four levels. Unit tests cover validators, metrics and model code. Integration tests cross the boundary between the two services. Whole-journey tests follow a deal from listing to confirmed delivery, including a shipment simulation of 39 tests that needs no database, and Playwright tests exercise the browser. The analytics suite defines 497 test functions in 47 modules, and the backend has 24 test modules. In an independent audit run, 480 analytics tests passed and none failed. The suites that need MongoDB, a FastAPI test client or a browser could not run in that environment, and they run in continuous integration instead.

B. Validation on Simulated Data

Real price histories do not come with a known right answer, so each method was first tested on simulated data, where the truth is known (Table II).

ISSN: **3139-0862**

TABLE II. METHOD VALIDATION ON SIMULATED DATA

Experiment	Before the fix	After the fix
False winners among 40 random walks	4	0
Fallback interval coverage, 1,500 runs, nominal 90%	Zero-width band	86 to 94%
Segment sizes with one very large buyer	1, 20, 40	20, 20, 21
Bids 20% above asking, strictest anomaly tier	Not applicable	76% caught, 79% precision

The first row matters most. A random walk contains nothing a model can exploit, so any winner reported on it is a false discovery. Without correction, four of the forty series produced one; with Holm correction, none did. The fallback interval is not perfect, which is why the interface shows its measured coverage and does not promise 90 percent. Requiring three detectors to agree makes the anomaly tier cautious: it misses some unusual bids, but about four in five of its flags are correct. We designed these simulations to be hard on the methods. They show that the methods behave correctly under pressure, but they do not show how accurate forecasts are on real markets.

C. *Real Mandi History*

We ran the importer on AGMARKNET records from 2023-06-06 to 2025-06-11, about 728 thousand rows covering four crops. The daily value is the median across markets. The full laboratory was then run on onion, potato and wheat (Table III). For onion, gaps longer than three days break the history, so the longest unbroken stretch is analyzed and nothing is filled across the gaps.

TABLE III. REAL-DATA BACKTEST: BEST CHALLENGER AGAINST NAIVE

Crop	Days (markets)	Origins	Best challenger, MASE gain	Holm p
Onion	367 (1,043)	16	Mean reversion, 5.2%	0.69
Potato	737 (970)	34	Ridge, 7.2%	0.54
Wheat	246 (737)	10	Mean reversion, 9.5%	1.00

For every crop the champion remained the naive forecast. The best challenger was five to ten percent better on MASE, but never significantly better after correction. The predictability tests did find structure that a model could in principle use: mean reversion for potato and wheat (five-day variance ratios of 0.41 and 0.34) and momentum for onion (a ten-day ratio of 2.04). The models did not turn it into a reliable gain. Following the forecast did not pay either. The models never predicted a rise big enough to justify waiting, so their policy was the same as selling at once. Always waiting would have changed the return per decision by −2.9 percent for potato, +1.6 percent for onion and −0.2 percent for wheat, before storage costs. The 90 percent intervals covered 100, 97 and 97 percent of the newer test points, but those points number only 17, 8 and 5, so the figures carry little weight. We regard this as an honest and useful result. For these crops the system offers advice, with market comparisons, ranges and stated limits, and does not pretend to predict. The data end in June 2025. A scheduled daily collector, switched off until it is configured, is meant to extend them.

D. *Assistant Routing*

We tested the assistant on 190 labeled questions in three sets. The development set of 87 questions scored 100 percent. The second set of 48 questions, whose failures we used to improve the router, scored 96 percent. Both figures are optimistic. The held-out set of 55 questions was written after tuning finished and was never used to change the router. It scored 69 percent (95 percent confidence interval 56 to 80), or 75 percent (62 to 84) if a clarifying question counts as acceptable. Of the 29 held-out data questions, 11 reached the language model instead of the analytics code, and one reached the wrong analytics function. The price guard limits the damage from the first case. Paraphrases such as “quote for this morning” defeated the keyword rules, so the held-out figure is the one to quote, and a trained intent classifier is the obvious next step. People have not yet rated the quality of the language model's own general answers.

E. *Comparison with Earlier Work*

Table IV compares the system with the typical studies in Table I. The comparison is qualitative, because those studies do not report measurements for the same task.

TABLE IV. COMPARISON WITH TYPICAL EARLIER WORK

Feature	Typical earlier work	AgriTrade Analytics
Models compared	One, such as ARIMA or LSTM	Nine, including naive
Significance	Often not tested	Diebold-Mariano, Holm
Uncertainty	Point forecast	Conformal, coverage measured
Validation	Random split common	Rolling origin
Trading	None	Bidding, payment, tracked delivery
Data honesty	Rarely stated	Synthetic share shown

F. *Limitations*

About 99 percent of the marketplace records are synthetic, so the marketplace-level forecasts, segments and anomaly flags demonstrate methods and not a real market. The real price data cover four crops and end in June 2025, and no model beat naive on them. Mandi prices are wholesale prices and not what a farmer receives at the farm gate, and weather, policy and export restrictions are not modeled. Net profit depends on flat transport and storage assumptions that are disclosed but not measured, and its upper end is an optimistic bound. The Razorpay route has not been tried with live payments, no money is held in escrow, and the delivery code proves control of an inbox and not receipt of goods. Alert thresholds are uncalibrated defaults. The language model has been tested only with mocked responses, the companion guide covers only 23 pairings, and we have not carried out a usability study with farmers. A protocol and scoring tools exist, but no responses have been collected.

VI. CONCLUSION AND FUTURE WORK

AgriTrade Analytics pairs a working bidding marketplace, complete with traceable shipments and verified delivery, with an analytics service that is willing to say “I do not know.” In our view a forecast earns trust in three ways: by passing a fair test against a trivial baseline, by carrying an interval whose coverage is measured, and by appearing beside its own limitations. The simulated studies show that these safeguards remove false winners and unstable segments. The real mandi evaluation then put them to use, and it found that no model reliably beat the naive forecast. The system reports exactly that.

ISSN: **3139-0862**

Several steps come next. We plan to extend the real price history with the daily collector and repeat the evaluation, to replace the keyword router with a trained classifier, and to ask people to rate the general answers. On the marketplace side we intend to move to gateway-confirmed payments and to a delivery flow in which the party handing over the goods enters the code. On the analytics side we want to add weather and policy inputs, approximate farm-gate prices, bring storage and spoilage costs into the economic test, and support Kannada and other local languages. Finally, a usability study with farmers is planned.

ACKNOWLEDGMENT

The authors thank Prof. Gayathri N, Assistant Professor, Department of Computer Science and Engineering, City Engineering College, Bengaluru, for her guidance and supervision throughout this work.

REFERENCES

[1] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," PLoS ONE, vol. 13, no. 3, p. e0194889, 2018.

[2] R. K. Paul et al., "Machine learning techniques for forecasting agricultural prices: A case of brinjal in Odisha, India," PLOS ONE, vol. 17, no. 7, p. e0270553, 2022.

[3] V. Kasa, K. Mahajan, S. Dhautre, and S. Kapare, "An LSTM based crop price prediction system," Int. J. Creative Research Thoughts, vol. 9, no. 6, pp. f690–f697, 2021.

[4] N. M. Sindhur, C. Pavithra, and N. Muchikel, "A hybrid machine learning framework for optimizing crop selection via agronomic and economic forecasting," arXiv:2507.08832, 2025.

[5] C.-G. Wong, G.-K. Tong, and S.-C. Haw, "Exploring customer segmentation in e-commerce using RFM analysis with clustering techniques," J. Telecommunications and the Digital Economy, vol. 12, no. 3, pp. 97–125, 2024.

[6] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: An analysis and review," Int. J. Forecasting, vol. 16, no. 4, pp. 437–450, 2000.

[7] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," Int. J. Forecasting, vol. 22, no. 4, pp. 679–688, 2006.

[8] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," J. Business & Economic Statistics, vol. 13, no. 3, pp. 253–263, 1995.

[9] D. Harvey, S. Leybourne, and P. Newbold, "Testing the equality of prediction mean squared errors," Int. J. Forecasting, vol. 13, no. 2, pp. 281–291, 1997.

[10] S. Holm, "A simple sequentially rejective multiple test procedure," Scandinavian J. Statistics, vol. 6, no. 2, pp. 65–70, 1979.

[11] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," arXiv:2107.07511, 2021.

[12] R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani, "Conformal prediction beyond exchangeability," Annals of Statistics, vol. 51, no. 2, pp. 816–845, 2023.

[13] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[14] Directorate of Marketing and Inspection, Government of India, "AGMARKNET: Agricultural Marketing Information Network." [Online]. Available: https://agmarknet.gov.in

[15] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), Pisa, Italy, 2008, pp. 413–422.