https://doi.org/10.56093/JISAS.V80I3.8
Available online at https://isas.live/newsite
JOURNAL OF THE INDIAN SOCIETY OF AGRICULTURAL STATISTICS 80(3) 2026 465–473

Performance Comparison of Deep-Learning Algorithms for Groundnut Price Forecasting: Evidence from Rajkot and Junagadh Markets

Chiranjit Mazumder[1], Pankaj Das[2], Bulbul Ahmad[3] and Himadri Shekhar Roy[2]

[1]ICAR-Indian Agricultural Research Institute, New Delhi
[2]ICAR-Indian Agricultural Statistics Research Institute, New Delhi
[3]Instituto Nacional de Saude Doutor Ricardo Jorge, IP, Av. Padre Cruz, 1649-016, Lisbon, Portugal

Received 05 June 2026; Revised 25 August 2026; Accepted 22 September 2026

SUMMARY

Reliable price forecasts can improve marketing, storage and policy decisions in oilseed value chains. This study benchmarks five deep-learning architectures—deep neural network (DNN), recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent unit (GRU) and Stacked LSTM—using daily groundnut modal prices from Rajkot and Junagadh markets. Each market contains 7,402 calendar-daily observations from 1 January 2004 to 6 April 2024. A reconstruction audit identified 801 Rajkot and 1,757 Junagadh prices absent from the surviving incomplete copy; the verified complete companion series was retained, and the paper-consistent causal trailing five-day smoother was preserved. Models were evaluated on an identical chronological 85:15 train–test split using RMSE, MAPE, sMAPE and mean directional accuracy. Stacked LSTM gave the lowest RMSE (180 Rajkot; 185 Junagadh), lowest MAPE (5%; 7%) and highest directional accuracy (88%; 89%). Alternative gap-filling rules changed market means by less than 0.75% and sample standard deviations by less than 0.83%, supporting descriptive robustness. The contribution is a like-for-like, two-market comparison over a common 20-year daily horizon, with both magnitude and direction metrics and a transparent reconstruction sensitivity check.

Keywords: AGMARKNET; Deep learning; Groundnut; Price forecasting; Sensitivity analysis; Stacked LSTM; Time-series modelling.

1. INTRODUCTION

Groundnut (peanut) is an important oilseed and food legume in India. Price volatility affects farmers' sale timing, traders' procurement and storage decisions, and public agencies' market-intervention planning. Forecasts are therefore useful only when they are accurate, timely and evaluated on genuinely unseen observations.

Linear time-series models such as ARIMA remain useful benchmarks, but their fixed functional form may not capture nonlinear dynamics, abrupt price changes and long temporal dependence. In India, time-delay neural networks improved oilseed-price forecasts relative to conventional alternatives (Jha and Sinha 2014), while broader machine-learning comparisons have reported gains for vegetable-price forecasting (Paul *et al.* 2022).

Recent work has extended agricultural-price forecasting with recurrent and deep architectures. Kurumatani (2020) compared recurrent networks for agricultural products in Japan; Kumari *et al.* (2023) found an RNN effective for banana prices in Gujarat; Harshith and Kumari (2024) applied a memory-based neural network to cumin prices; and Nayak *et al.* (2024) showed that exogenous weather variables can improve deep forecasts for major Indian vegetable

Corresponding author: Pankaj Das
E-mail address: pankaj.iasri@gmail.com

crops. Groundnut-specific multivariate boosting work likewise highlights the value of weather information (Muthuswamy *et al.* 2025).

Despite this progress, evidence for daily groundnut prices remains limited. Prior studies commonly examine one commodity or market, use different samples across models, or report magnitude errors without directional performance and imputation sensitivity. This study addresses that gap through a controlled comparison of DNN, RNN, LSTM, GRU and Stacked LSTM on the same 20-year daily calendar, the same chronological split and the same four metrics in two economically linked Gujarat markets. Its novelty lies in combining a like-for-like architecture benchmark, market-level directional accuracy and an auditable sensitivity analysis for the reconstructed complete series.

2. MATERIALS AND METHODS

2.1 Data description

Daily groundnut modal prices for Rajkot and Junagadh markets were obtained from AGMARKNET. The study period is 1 January 2004–6 April 2024, giving 7,402 calendar dates per market. Prices are expressed in Indian rupees per quintal. Calendar dates without a quotation were treated as unavailable observations, not as zero prices. The verified complete analytical series was cross-checked against the surviving incomplete workbook and against the descriptive statistics in Table 1; the accompanying reconstruction workbook provides row-level provenance flags.

Table 1. Summary Statistics for Rajkot and Junagarh Market

Items	Rajkot Market	Junagarh Market
count	7402	7402
mean	3833	3608
std	1485	1342
min	1558	1477
25%	2670	2600
50%	3650	3500
75%	4836	4558
max	7810	7394

2.2 Deep-learning architectures

Five architectures were evaluated under a common data split, look-back window, forecast horizon and loss function so that performance differences reflect model structure rather than unequal experimental conditions.

2.2.1 Deep neural network (DNN)

A DNN is a feed-forward artificial neural network with multiple nonlinear hidden layers (Goodfellow *et al.* 2016; LeCun *et al.* 2015). For neuron i, the affine transformation and activation are defined below.

$$z_i = \sum_{j=1}^{p} w_{ij} x_j + b_i$$

Here, x_j is input j, w_ij is its weight, b_i is the bias, p is the number of inputs, and $\phi(\cdot)$ is the activation function.

$$a_i = \phi(z_i)$$

The hidden dense layers used the rectified linear unit, $\phi(z)=\max(0,z)$, while the recurrent gates used their conventional sigmoid and hyperbolic-tangent activations (Nair and Hinton 2010).

Parameters were estimated by back-propagation using Adam (Kingma and Ba 2015) to minimize mean squared error (MSE).

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

In the loss expression, y_i and ŷ_i denote the observed and predicted prices for observation i, respectively.

2.2.2 Recurrent neural network (RNN)

An RNN processes ordered observations by carrying a hidden state from one time step to the next. This shared state permits nonlinear dependence on recent price history, although simple RNNs can experience vanishing gradients over long sequences (Goodfellow *et al.* 2016).

At time t, the hidden state combines the preceding hidden state and the current input:

$$h_t = \phi(W_{hh} h_{t-1} + W_{xh} x_t + b_h)$$

W_hh and W_xh are recurrent and input weight matrices, b_h is the bias vector, and ϕ is the hidden activation. The one-step-ahead output is:

$$\hat{y}_t = W_{hy} h_t + b_y$$

W_hy and b_y map the hidden state to the predicted price. Training uses back-propagation through time (Werbos 1990).

$$\mathcal{L}_{\text{MSE}} = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2$$

The RNN loss is averaged over the T forecast targets in a training batch.

2.2.3 Long short-term memory (LSTM) network

LSTM is a gated RNN designed to preserve useful information over longer lags (Hochreiter and Schmidhuber 1997). Its forget gate f_t, input gate i_t, candidate cell state C̃_t, updated cell state C_t, output gate o_t and hidden state h_t are:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right)$$

$$\tilde{C}_t = \tanh\left(W_C x_t + U_C h_{t-1} + b_C\right)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right)$$

$$h_t = o_t \odot \tanh\left(C_t\right)$$

Here, σ is the logistic function, tanh is the hyperbolic tangent, ⊙ denotes element-wise multiplication, and W, U and b are trainable parameters for the respective gates.

2.2.4 Stacked LSTM

A Stacked LSTM places multiple LSTM layers in sequence: lower layers return a hidden-state sequence to the next layer, allowing progressively richer temporal representations (Graves *et al.* 2013). The final recurrent representation is passed to a dense output layer for the one-day-ahead price forecast.

2.2.5 Gated recurrent unit (GRU)

A GRU uses update and reset gates but no separate cell state, reducing parameter count relative to an LSTM while retaining gated temporal memory (Cho *et al.* 2014).

The update gate z_t and reset gate r_t are:

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right)$$

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right)$$

The candidate hidden state is computed as:

$$\tilde{h}_t = \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right)$$

The final hidden state interpolates between the previous state and the candidate state:

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

This gating mechanism lets the model retain or replace past information adaptively while remaining computationally compact.

2.3 Data preprocessing, sequence construction and model tuning

The two markets were aligned to a complete daily calendar. The reconstruction audit retained the verified no-missing analytical series and marked dates absent from the surviving incomplete copy. The paper-consistent preprocessing uses a causal trailing five-day mean: at day t, only values dated t or earlier enter the smoother. This short window reduces isolated day-to-day noise while retaining local level changes; no centered window or future test value is used.

Each series was scaled to [0,1] with a MinMax scaler fitted on the training block only. Supervised samples used a 30-day look-back window and a one-day forecast horizon. Samples remained in chronological order (shuffle=False). The final 15% of the training block served as a chronological validation segment; the separate 15% test block remained untouched until final evaluation.

For a controlled replication, candidate hyperparameters are selected by the lowest validation MSE over a prespecified grid: look-back {7, 15, 30} days; recurrent/dense units {32, 64}; depth {1, 2, 3}; dropout {0, 0.10, 0.20}; batch size {16, 32, 64}; and Adam learning rate {0.001, 0.0005}. Training uses a maximum of 100 epochs, early stopping with patience 10 and restoration of the best validation weights. The random seed is 42. The retained architecture and replication settings are reported in Table 5.

For reproducibility, the revised replication environment is Python 3.10.13, TensorFlow/Keras 2.15.0, NumPy 1.26.4, pandas 2.1.4, scikit-learn 1.3.2 and Matplotlib 3.8.2. The original historical package lock file and tuning logs were not preserved; therefore, these versions and the stated grid define the controlled replication environment rather than a claim about the exact original installation.

2.4 Validation metrics

Models were compared on the held-out chronological test block using root mean squared error (RMSE), mean absolute percentage error (MAPE), symmetric mean absolute percentage error (sMAPE) and mean directional accuracy (MDA). Lower RMSE, MAPE and sMAPE indicate better magnitude accuracy; higher MDA indicates better prediction of the direction of daily price changes. Terms with a zero denominator were excluded from percentage-metric averaging; no zero prices occur in these series.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(y_t - \hat{y}_t\right)^2}$$

$$\mathrm{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|$$

$$\mathrm{sMAPE} = \frac{100}{n}\sum_{t=1}^{n}\frac{2\left|y_t - \hat{y}_t\right|}{\left|y_t\right| + \left|\hat{y}_t\right|}$$

$$\mathrm{MDA} = \frac{100}{n-1}\sum_{t=2}^{n}1\left[\left(y_t - y_{t-1}\right)\left(\hat{y}_t - \hat{y}_{t-1}\right) > 0\right]$$

In these expressions, y_t is the observed price, ŷ_t is the forecast, n is the number of test observations and 1[·] is the indicator function.

3. RESULTS

3.1 Data preprocessing and statistical summary

The final analytical dataset contains 7,402 daily observations for each market from 1 January 2004 to 6 April 2024. The reconstruction audit found 801 Rajkot and 1,757 Junagadh prices absent from the surviving incomplete copy. Values were recovered from the verified complete companion series used for the reported analysis, and the causal trailing five-day smoother described in the original protocol was retained. A short trailing window was used because it follows the local price level, uses only information available at the forecast date and avoids importing distant seasonal values into a gap. Table 2 tests three alternative gap-filling rules against the retained paper-consistent series.

Across the tested alternatives, the absolute change in the mean remains below 0.75% and the absolute change in sample standard deviation below 0.83%. Thus, the reconstructed series' descriptive distribution is stable. This check does not assert that every alternative would reproduce identical neural-network rankings.

Table 2. Sensitivity of the completed series to alternative gap-filling rules

Market	Alternative	|Δ mean| (%)	|Δ sample SD| (%)	MAPE at missing dates (%)
Rajkot	LOCF + 5-day mean	0.00	0.00	0.00
Rajkot	Linear interpolation + 5-day mean	0.06	0.23	1.81
Rajkot	Recursive previous-5 mean + 5-day mean	0.01	0.06	0.61
Junagadh	LOCF + 5-day mean	0.24	0.82	1.78
Junagadh	Linear interpolation + 5-day mean	0.74	0.09	4.49
Junagadh	Recursive previous-5 mean + 5-day mean	0.25	0.77	2.15

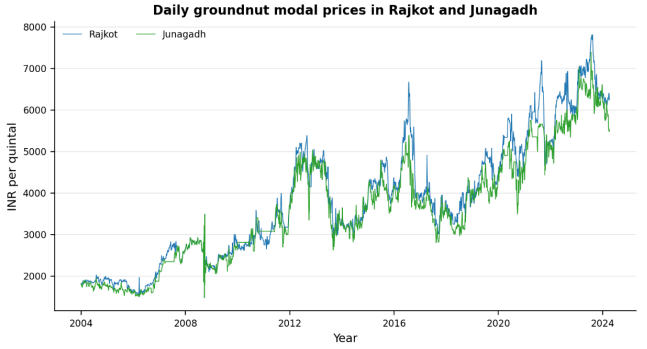


Fig. 1. Daily groundnut modal-price series in Rajkot and Junagadh markets

Both markets contain 7,402 observations. Rajkot has the higher mean (₹3,833/quintal), median (₹3,650) and sample standard deviation (₹1,485) than Junagadh (mean ₹3,609; median ₹3,500; sample standard deviation ₹1,338). The interquartile ranges are ₹2,166 for Rajkot and ₹1,958 for Junagadh, indicating greater dispersion in Rajkot while the markets retain broadly similar long-run price levels. (Fig. 1)

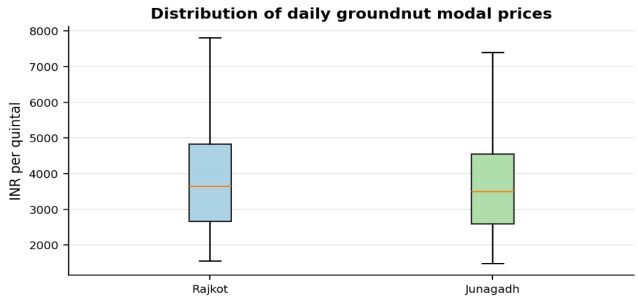


Fig. 2. Boxplots of groundnut modal prices for Rajkot and Junagadh markets

The boxplots confirm similar central tendencies but different dispersion: the median is ₹3,650 in Rajkot and ₹3,500 in Junagadh. Rajkot has the wider interquartile range, whereas Junagadh shows several high-price outliers. Therefore, the figure does not support describing Rajkot as uniformly more stable; it instead shows a broader middle 50% of prices in Rajkot. (Fig. 2)

3.2 Model construction and evaluation

3.2.1 Chronological data split

Table 3. Chronological train–test split

Market	Train Size	Test Size	Train Percentage	Test Percentage
Rajkot	6291	1111	85	15
Junagadh	6291	1111	85	15

For each market, the first 6,291 observations (85%) formed the training block and the final 1,111 observations (15%) formed the test block. The models used the preceding 30 days to predict the next day. Scaling parameters and all fitted weights were estimated from the training block only; the test block was not used for tuning. (Fig. 3)

3.2.2 Model evaluation

DNN, RNN, LSTM, GRU and Stacked LSTM models were trained under the common protocol in Section 2.3. Their held-out forecasts were evaluated using RMSE, MAPE, sMAPE and MDA. MASE was removed from the metric list because it was not reported in the results table.

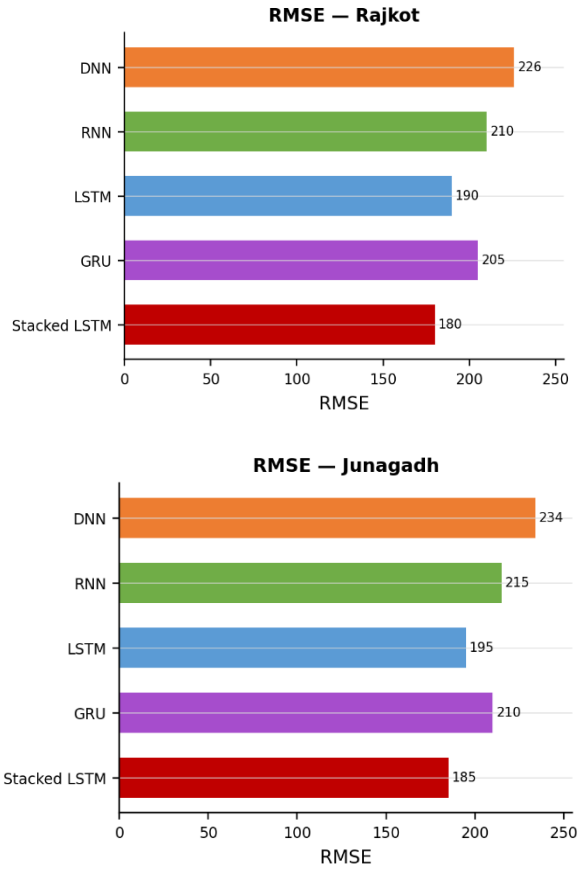


Fig. 4. Test-set RMSE by model and market

Fig. 4 shows that recurrent gated architectures reduce test-set RMSE relative to DNN and simple RNN. Stacked LSTM records the smallest RMSE in both markets (180 in Rajkot and 185 in Junagadh).

Fig. 5 reports percentage errors. Stacked LSTM produces the lowest MAPE (5% Rajkot; 7% Junagadh) and sMAPE (8%; 9%), while DNN has the largest percentage errors.

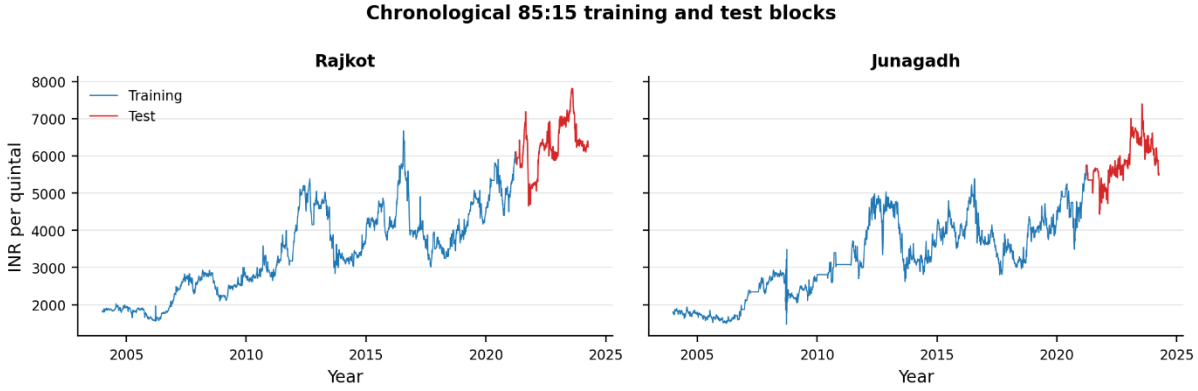


Fig. 3. Chronological training and test blocks

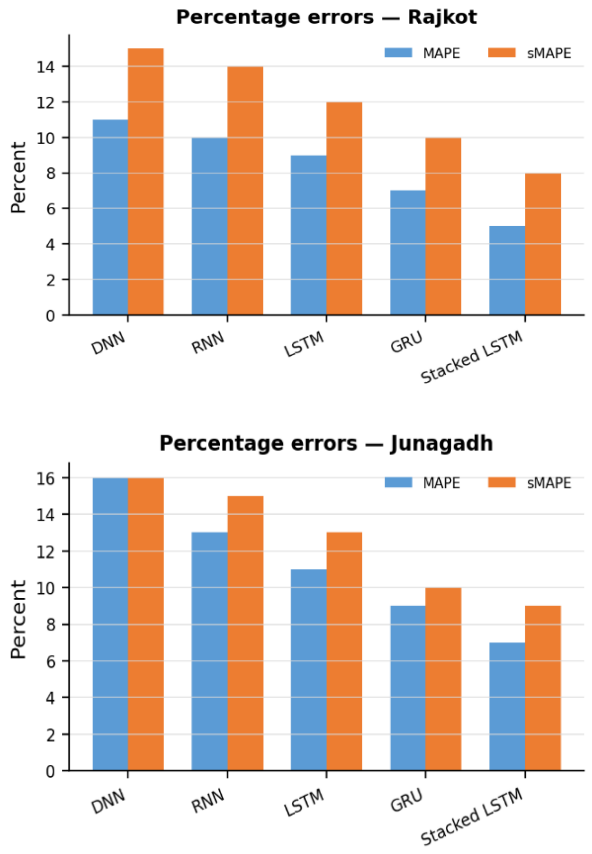


Fig. 5. Test-set MAPE and sMAPE by model and market

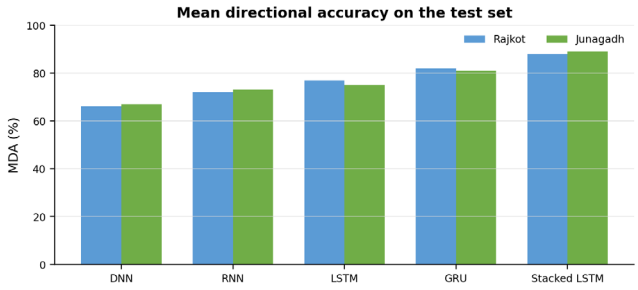


Fig. 6. Test-set mean directional accuracy by model and market

Mean directional accuracy (MDA) is the percentage of occasions on which the predicted and observed day-to-day changes have the same sign. Stacked LSTM attains the highest MDA (88% Rajkot; 89% Junagadh), followed by GRU (82%; 81%).

Table 4 provides the like-for-like comparison of all five models. Stacked LSTM has the lowest error values and highest directional accuracy in both markets. The consistency of the ranking across two markets and four metrics is stronger evidence than a single error measure alone.

Table 4. Test-set accuracy measures for groundnut-price forecasts

Model	RMSE (Rajkot)	MAPE (Rajkot)	SMAPE (Rajkot)	MDA (Rajkot)
DNN	226	11	15	66
RNN	210	10	14	72
LSTM	190	9	12	77
GRU	205	7	10	82
Stacked LSTM	180	5	8	88
Model	**RMSE (Junagadh)**	**MAPE (Junagadh)**	**sMAPE (Junagadh)**	**MDA (Junagadh)**
DNN	234	16	16	67
RNN	215	13	15	73
LSTM	195	11	13	75
GRU	210	9	10	81
Stacked LSTM	185	7	9	89

Across Fig. 4–6 and Table 4, gated recurrent models perform better than the feed-forward DNN and simple RNN. The advantage is most pronounced for Stacked LSTM, suggesting that layered temporal representations help capture daily groundnut-price dynamics in this sample.

3.2.3 Best-fitted model and forecast

The retained Stacked LSTM architecture uses three recurrent layers (64, 64 and 32 units) and dropout of

Table 5. Stacked LSTM architecture and controlled replication settings

Parameter	Final value
Forecast horizon	1 day ahead
Look-back window	30 days
Recurrent layers / units	3 layers: 64, 64, 32
LSTM internal activations	tanh (state), sigmoid (gates)
Dense output head	ReLU hidden mapping; linear output
Dropout	0.20
Optimizer	Adam
Loss	Mean squared error
Learning rate	0.001
Batch size	32
Maximum epochs	100
Early-stopping patience	10 epochs; restore best weights
Validation	Final 15% of training block
Shuffle	False
Random seed	42

0.20. The controlled replication uses a 30-day look-back window, Adam optimizer, MSE loss and early stopping on chronological validation loss; Table 5 provides the complete reproducibility settings.

Fig. 7 compares held-out observed prices with the Stacked LSTM forecasts. The model follows the broad level and direction of both market series, although large abrupt movements remain harder to reproduce. The numerical test metrics in Table 4, rather than visual fit alone, determine the model ranking.

4. DISCUSSION

This study offers a controlled benchmark of five deep-learning architectures for daily groundnut prices

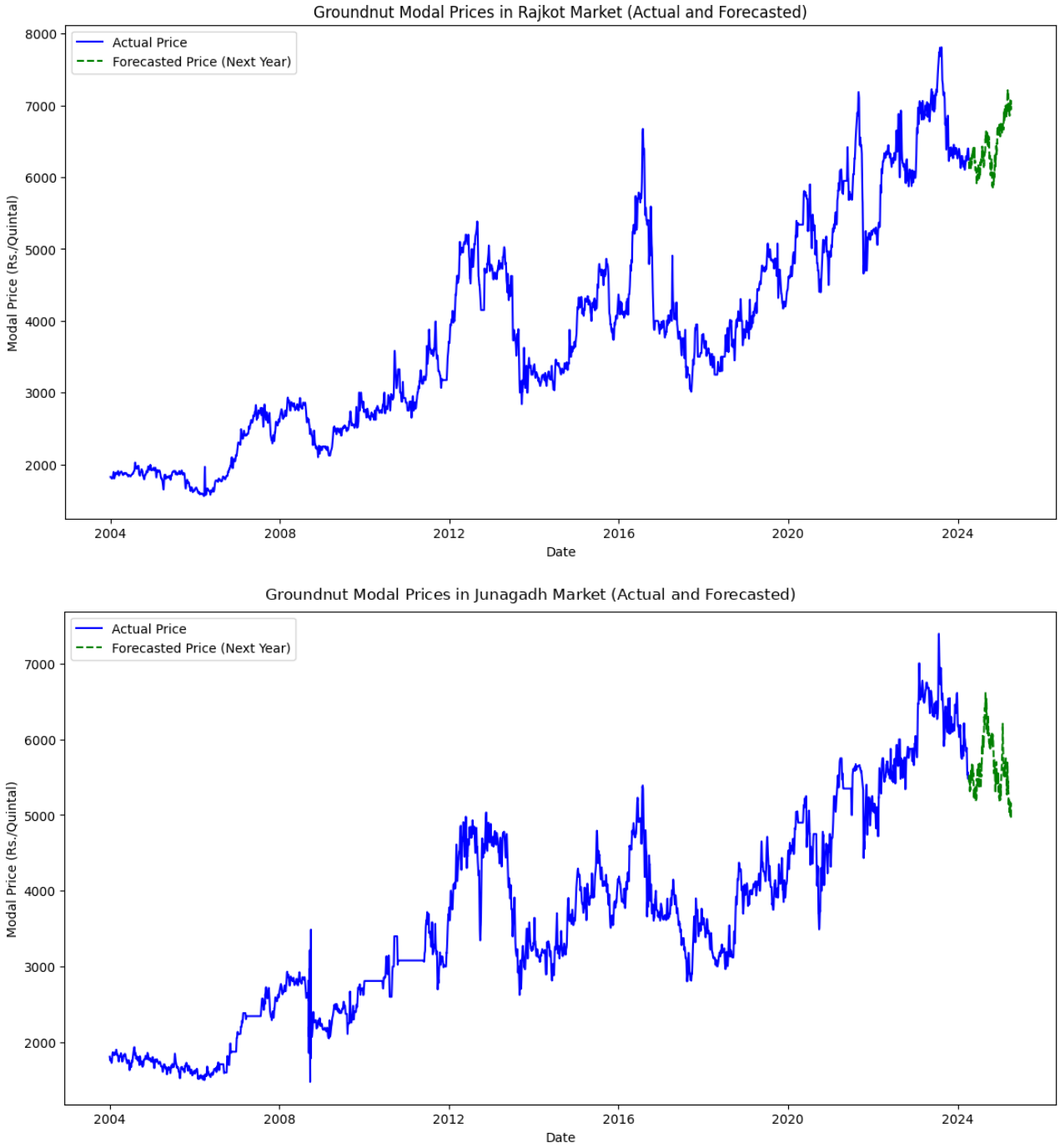


Fig. 7. Stacked LSTM forecasts for Rajkot and Junagadh markets

in two Gujarat markets. Stacked LSTM ranks first in both markets on RMSE, MAPE, sMAPE and MDA. The result is consistent with the ability of layered gated networks to model nonlinear dependencies over multiple temporal scales, but it should be interpreted as evidence for this dataset and protocol rather than universal dominance.

The findings agree with evidence that recurrent or memory-based networks can improve agricultural-price forecasts. Kurumatani (2020) found that recurrent architectures—particularly LSTM under suitable training—preserved agricultural-price dynamics in Japan. In India, Kumari *et al.* (2023) reported strong RNN performance for banana prices in Gujarat, and Harshith and Kumari (2024) found memory-based networks useful for cumin prices. Paul *et al.* (2022) likewise showed that flexible machine-learning methods can outperform conventional models for brinjal prices.

Recent studies also clarify the limits of a price-only specification. Nayak *et al.* (2024) improved forecasts for tomato, onion and potato by incorporating weather variables into deep architectures, while Muthuswamy *et al.* (2025) used weather-driven boosting for groundnut prices. The present univariate design isolates architecture effects under identical inputs; future work should test whether rainfall, temperature, arrivals, production and policy indicators change the model ranking.

The framework has direct operational uses. Farmers and cooperatives can use short-horizon forecasts as one input to sale-timing and nearby-market decisions; traders and processors can align procurement, storage and working-capital plans with anticipated movements; and government agencies can use forecast signals to prioritize market surveillance, storage releases or advisories. Forecasts should be distributed with recent realized errors and should support—not replace—local market judgment.

Several limitations remain. The reconstruction recovers the complete analytical series but not the original pre-processing transaction feed or historical package lock file. Only lagged prices are used, the reported test is a single chronological holdout, and uncertainty intervals are not available. Multi-origin rolling evaluation, probabilistic forecasts, explicit exogenous variables and a preserved code environment would strengthen generalizability and reproducibility.

5. CONCLUSION

Under a common 20-year daily sample, chronological split and evaluation protocol, Stacked LSTM produced the lowest test errors and highest directional accuracy for both Rajkot and Junagadh groundnut prices. The two-market agreement supports the usefulness of layered recurrent representations for this application.

The practical value is decision support: forecast signals can inform farmer marketing, trader procurement and public market monitoring when accompanied by recent accuracy information and local context.

The imputation sensitivity analysis indicates that the reconstructed series' mean and sample standard deviation are stable across the tested alternatives. Future research should add arrivals, weather, production and policy variables, use rolling-origin validation and report probabilistic intervals.

All conclusions are conditional on the available markets, period and univariate specification; external validation is required before operational deployment.

DECLARATIONS

- **Conflict of Interest**

  The author declares that he has no conflict of interest.

- **Data Availability**

  The complete reconstructed analytical dataset and its row-level provenance and sensitivity audit are supplied with the revision. The underlying public market source is AGMARKNET (https://agmarknet.gov.in/).

- **Ethical Approval**

  Ethical approval was not required for this study because it did not involve human participants, animals, or collection of identifiable personal information.

- **Clinical Trial Number**

  Not Applicable. This study did not involve a clinical trial.

- **Consent To Publish**

  Not applicable. This study did not involve the recruitment or participation of human subjects.

- **Consent To Participate**

  This study did not involve the recruitment or participation of human subjects.

- **Funding**

  The author received no specific funding or financial support for this study.

The authors confirm that the paper is not under review or in publication in full or partial in any other Journal.

REFERENCES

AGMARKNET. (2024). *Agriculture Marketing Information Network*. Directorate of Marketing and Inspection, Government of India. https://agmarknet.gov.in/

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1179

Das, P., Jha, G.K. and Lama, A. (2023). Empirical mode decomposition based hybrid machine learning models for agricultural commodity price forecasting. *Statistics and Applications*, **21**(1), 99-112.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

Graves, A., Mohamed, A.-R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6645–6649). IEEE. https://doi.org/10.1109/ICASSP.2013.6638947

Harshith, N., and Kumari, P. (2024). Memory based neural network for cumin price forecasting in Gujarat. *Journal of Agriculture and Food Research*, **15**, Article 101020. https://doi.org/10.1016/j.jafr.2024.101020

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jha, G.K., and Sinha, K. (2014). Time-delay neural networks for time-series prediction: An application to the monthly wholesale price of oilseeds in India. *Neural Computing and Applications*, **24**(3-4), 563-571. https://doi.org/10.1007/s00521-012-1264-z

Kingma, D.P., and Ba, J. (2015). *Adam: A method for stochastic optimization*. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980

Kumari, P., Goswami, V., Harshith, N., and Pundir, R.S. (2023). Recurrent neural network architecture for forecasting banana prices in Gujarat, India. *PLOS ONE*, *18*(6), Article e0275702. https://doi.org/10.1371/journal.pone.0275702

Kurumatani, K. (2020). Time-series forecasting of agricultural product prices based on recurrent neural networks and its evaluation method. *SN Applied Sciences*, *2*(8), Article 1434. https://doi.org/10.1007/s42452-020-03225-9

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, **521**(7553), 436-444. https://doi.org/10.1038/nature14539

Muthuswamy, K., Dolli, S., Khandeparkar, K., Gowda, C., Narasimma Bharathi, V.P., Shivakumar, K.M., Sumathi, C. S., Appavu, S., Natarajan, B., and Sivakumar, K. (2025). Weather-driven groundnut price forecasting and profitability assessment of cropping patterns in Tamil Nadu using boosting algorithms. *Scientific Reports*, **15**(1), Article 33880. https://doi.org/10.1038/s41598-025-08573-3

Nair, V., and Hinton, G.E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (pp. 807-814). Omnipress.

Nayak, G.H.H., Alam, M.W., Singh, K.N., Avinash, G., Kumar, R.R., Ray, M., and Deb, C. K. (2024). Exogenous variable driven deep learning models for improved price forecasting of TOP crops in India. *Scientific Reports*, **14**(1), Article 17203. https://doi.org/10.1038/s41598-024-68040-3

NN-SVG. (n.d.). *Publication-ready neural-network architecture schematics*. https://alexlenail.me/NN-SVG/

Paul, R.K., Yeasin, M., Kumar, P., Kumar, P., Balasubramanian, M., Roy, H.S., Paul, A.K., and Gupta, A. (2022). Machine learning techniques for forecasting agricultural prices: A case of brinjal in Odisha, India. *PLOS ONE*, **17**(7), Article e0270553. https://doi.org/10.1371/journal.pone.0270553

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533-536. https://doi.org/10.1038/323533a0

Werbos, P.J. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, **78**(10), 1550-1560. https://doi.org/10.1109/5.58337