http://doi.org/10.35784/iapgos.8032 *received: 07.11.2025 | revised: 02.01.2026 | accepted: 26.01.2026 | available online: 30.03.2026*

CROP PRICE FORECASTING USING A TEMPORAL FUSION TRANSFORMER FOR KRISHNA DISTRICT OF ANDHRA PRADESH

Dedeepya Manikonda[1], Ashutosh Satapathy[1], Keerthi Padamata[1], Jaswanthi Machcha[2], J. Chandrakant Badajena[3]

[1]Siddhartha Academy of Higher Education, Department of Computer Science and Engineering, Tumkur, India, [2]Amazon, Bangalore, India, [3]Odisha University of Technology and Research, School of Computer Science, Bhubaneswar, India

Abstract. *Indian farmers experience ongoing income volatility from fluctuating market prices, which erodes their financial health and long-term sustainability. To resolve this issue, a revamped Temporal Fusion Transformer (TFT), a state-of-the-art deep learning model is proposed for time-series forecasting with applications in agricultural price prediction. The TFT takes into account critical factors, including rainfall and temperature, previous price trends and market pressure in giving accurate, actionable forecasts to farmers. The model performed well when trained on a large database from Krishna district, Andhra Pradesh, India between January 2017 and September 2024. The TFT model achieved a RMSE of 99.13, MAPE of 2.16%, MAE of 72.08 and an accuracy rate of 93.24%. The system also allows the farmer to compare the forecast with the MSP and give them a very precise suggestion to maximise their revenue. It allows for farmers to take proactive decisions and supports an aware decision making approach by mitigating price volatility and stabilizing income.*

Keywords: crops price forecasting, climate and price trends, TFT, multi-parameter modelling, MSP

PROGNOZOWANIE CEN PLONÓW Z WYKORZYSTANIEM TEMPORAL FUSION TRANSFORMER DLA OKRĘGU KRISHNA W STANIE ANDHRA PRADESH

Streszczenie. *Indyjscy rolnicy borykają się z ciągłą zmiennością dochodów spowodowaną wahaniami cen rynkowych, co negatywnie wpływa na ich kondycję finansową i długoterminową stabilność. Aby rozwiązać ten problem, zaproponowano ulepszony model Temporal Fusion Transformer (TFT) – najnowocześniejszy model uczenia głębokiego – przeznaczony do prognozowania szeregów czasowych, który znajduje zastosowanie w przewidywaniu cen produktów rolnych. Model TFT uwzględnia kluczowe czynniki, w tym opady deszczu i temperaturę, wcześniejsze trendy cenowe oraz presję rynkową, dostarczając rolnikom dokładne i przydatne prognozy. Model sprawdził się podczas szkolenia na dużej bazie danych z dystryktu Krishna w stanie Andhra Pradesh w Indiach w okresie od stycznia 2017 r. do września 2024 r. Model TFT osiągnął RMSE na poziomie 99,13, MAPE na poziomie 2,16%, MAE na poziomie 72,08 oraz wskaźnik dokładności wynoszący 93,24%. System pozwala również rolnikom porównać prognozę z MSP i daje im bardzo precyzyjne sugestie dotyczące maksymalizacji przychodów. Umożliwia to rolnikom podejmowanie proaktywnych decyzji i wspiera świadome podejście do podejmowania decyzji poprzez łagodzenie zmienności cen i stabilizację dochodów.*

Słowa kluczowe: prognozowanie cen upraw, trendy klimatyczne i cenowe, TFT, modelowanie wieloparametrowe, MSP

Introduction

Agriculture holds the key to supporting the world population, a role that is further increasing with the increase in the world population. Farmers are the backbone of India's economy. However, they face many challenges, such as price volatility uncertainty driven by climatic factors, market demand changes, and transportation system inefficiencies [6, 8, 20, 26]. These induce income instability and render it difficult to decide the optimal time to harvest crops. Accurate price forecasting enables farmers to forecast market tendencies, plan harvesting efficiently, and minimize reliance on speculation [9]. Moreover, the comparison of predicted prices with the Minimum Support Price (MSP) empowers farmers to make sound choices, selling through government-managed markets when the expected price is less than the MSP supports income stability, while selling through private markets when predicted prices are equal to or greater than the MSP supports improved selling decisions [5]. This reduces financial risk, improves income stability, and strengthens farmers' livelihoods.

Machine learning and deep learning are presently disruptive technologies capable of reading big and complex data to find patterns that might not be apparent by conventional means. Machine learning builds models that learn from examples, whereas deep learning uses neural networks to find complex and temporal patterns [18]. These techniques have been used to provide estimations of seasonal, cyclical and non-seasonal price variations in the past markets for helping farmers make decisions over selling, storage and marketing decisions on produce with reduced risk [4]. The income generated from MSP directly contributes to financial returns of farmers and in turn stabilizes the farm economy [8].

The Temporal Fusion Transformer (TFT) is a deep learning model for explainable time series forecasting that integrates the power of transformer models and Recurrent Neural Networks (RNNs) to track short-range and long-range temporal dependencies in multifaceted datasets. The TFT automatically selects discriminative input features using gating mechanisms, variable selection networks, static covariate encoders and temporal processing layers. It does the opposite by focusing on vital time steps and variables, which leads to an increase in accuracy and interpretability. It also combines different data sources and automatically includes static covariates, and is thus more appropriate for multivariate time series forecasting [7, 28]. Through processing past prices, weather patterns, and market demand, TFT produces price forecasts that reflect trends, such as seasonality and irregular fluctuations.

1. Related work

Cheung et al. developed a three-dimensional Convolutional Neural Network (3D-CNN) with a clustering structure to forecast the crop prices, a promising forecasting accuracy has been achieved with a Mean Absolute Percentage Error (MAPE) of 0.083, a Root Mean Square Error (RMSE) of 40.39 and an MAE of 32.31 [3]. Xu et al. compared two variants of the Nonlinear AutoRegressive Neural Network (NARNN) and its augmented model with exogenous inputs (NARNN-X) for prediction of daily prices series of soybeans and soybean oil. Their findings revealed that the prediction accuracy of both models was stable and high, but NARNN-X had slightly better performance. In particular, the average Relative Root Mean Square Error (RRMSE) for soybeans and soybean oil are 1.701% and 1.777%, respectively on NARNN, while slightly lesser values of 1.695% and 1.775% were found on NARNN-X [23]. While Soni et al. implemented machine learning models, such as Decision Tree Regressor and Random Forest, utilizing publicly accessible agricultural data provided by data.gov.in. According to their outcomes, the predictive accuracy of the Decision Tree model was 92%, which proved to be superior to that of the Random Forest model, 88% accuracy [19].

Kurumatani suggested that Time Alignment of Time Point Forecast-Long Short-Term Memory (TATP-LSTM) model for the forecasting of agricultural prices based on National

Horticulture Board data, and obtained a 0.83 correlation for the price of tomatoes and a 0.98 correlation for onion prices in India [13]. Harshith et al. compared the LSTM, GRU, RNN, Deep Neural Network (DNN) and Stack-LSTM models in predicting agricultural prices based on Indian Agriculture Market Information Network data and achieved the superior performance by using Stack-LSTM crashing with RMSE of 3115.10, MAPE of 12.25% and Mean Directional Accuracy (MDA) as 0.51 [10]. Mahto et al. used the Autoregressive Integrated Moving Average (ARIMA) model to forecast sunflower seed prices in the Indian market, achieving a MAPE of 2.30% and a RMSPE of 3.44%, which indicates strong short-term forecasting performance. The results demonstrate that classical time-series models like ARIMA can effectively capture price trends without relying on deep learning approaches [12]. Purohit et al. tested Multiplicative-ARIMA-SVM on forecasting agricultural product prices and obtained 0.83 and 0.98 correlation values for tomatoes and onions, respectively [17].

Subhasree et al. created a Genetic Algorithm Neural Network (GANN) to predict agricultural prices based on Tirupur New Bus Stand Ulavar Market pricing data. They achieved a total accuracy of 89% overall [21]. In another study, Nassar et al. developed a hybrid model composed of an Attention based Convolution Neural Network and Long Short-Term Memory (ATT-CNN-LSTM), as well as Average Growth Model–Weather to Price (AGM W2P) and Weather to Yield (W2Y), compared against one another. Performance statistics consisted of 23.53 as an AGM-W2P score, a W2Y score value of 0.03, and an R-squared (R^2) value of 0.83 [16]. Yun et al. implemented a Bi-directional Depth-wise Separable Convolutional LSTM with Attention mechanism (Bi DSConvLSTM-Attention) for the prediction of grain futures price, having solid performance figures of RMSE equal to 5.61, MAE of 3.63, MAPE of 0.55, and R^2 equal to 0.9984 [27]. Additionally, Li et al. introduced the Time-Series Prediction and Causal Dynamics Regression (TPCDR) model to predict vegetable prices. The model recorded high accuracy on various levels: 99.62% in natural conditions, 99.82% in supply conditions, 91.69% for demand elements, and 93.56% for economic policy effects [14]. Elbasi et al. in 2023 made a comparative study with 15 machine learning algorithms for crop prediction. The Bayesian Network (BN) model achieved the highest accuracy of 99.59%, followed by Hoeffding Tree and Naive Bayes (NB) models with 99.46% accuracy [6].

Yang et al. introduced a new model called Fusion Transformer (FusFormer) for predicting Mooney Viscosity, which surpassed earlier transformer-based models by obtaining an RMSE of 3.37 and a Root Relative Squared Error (RRSE) of 0.321 [25]. Avinash et al. proposed a deep learning model constrained by Hidden Markov Models (HMM) for time-series forecasting. Their HM-GRU model was very effective in terms of predictive accuracy for the short-term forecasts. For example, the RMSE, MAE, and MAPE values for the 1-week prediction were 79.69, 61.72, and 4.30%, respectively. For the 4-week and 8-week forecasts, the RMSE, MAE, and MAPE values were 113.86, 91.29, 6.49%, and 157.07, 123.60, 9.10%, respectively. For long-term prediction at 12 weeks, they demonstrated that the HM-BiGRU yielded the best with RMSE, MAE, and MAPE values of 182.19, 151.52, and 11.18%, respectively [1]. Guo et al. proposed the Attention-based LSTM-ARIMA Back propagation Neural Network (AttLSTM ARIMA-BP) model for forecasting corn prices in Sichuan by incorporating the Attention Mechanism, LSTM, ARIMA, and Back propagation Neural Network (BPNN). The hybrid model had a highly accurate MAPE of 0.0043%, an MAE of 1.51, and an RMSE of 1.642 [9].

2. Proposed methodology

The crop price forecasting system aims at helping farmers to predict the prices of crops and find appropriate selling places in relation to MSP. It is designed in an intuitive and user-friendly manner for farmers to input crop information, and get accurate price prediction without stress. It also provides practical suggestions for decision making, so that farmers can maximise agricultural profits and sales strategy.

2.1. System architecture

Predicting vegetable and crop prices begins with acquiring the past price information, climate information which includes rainfall and temperature, market data. The data is cleaned, address missing values and prepared for analysis. The TFT model is utilized by defining its architecture and selecting relevant features. Subsequently, the model is trained on a portion of the data set on which it learns patterns and fine-tunes its internal parameters. The train model is then tested on unseen data to be able to quantify its generalisation power, and predictions are compared with ground truths as a mean for the validation of the performance. Finally, the model is deployed in real-time with a web-based application. Farmers can enter crop data, get price forecast and compare MSP to find out which selling point will fetch them maximum profit. The smart agriculture platform enables farmers to make data-driven decisions that improve profitability and sales strategy. Figure 1 depicts the suggested system architecture for crops and vegetable price prediction using TFT.

2.2. Architecture selection

Traditional forecasting models, such as statistical models like ARIMA and SARIMA, tree-based models like XGBoost and Random Forest, and deep learning models like LSTM, GRU, and CNN-LSTM, have been proven to work in many applications. They usually fail to incorporate short- and long-term dependencies in sophisticated, multi-input time series data. The modified Temporal Fusion Transformer (TFT) is employed as it can deal with sequential data and learn high-level temporal features Figure 2. TFT deviates from regular recurrent models by merging LSTMs with advanced attention mechanisms, enabling it to pinpoint significant features and minimize the effects of noise in agricultural data. In addition, TFT effectively models high-order temporal relationships and multi-dimensional inputs by combining recurrent layers to recognize short-term patterns with self-attention mechanisms to acknowledge long-term relationships, thus being a robust and reliable forecasting tool [11].

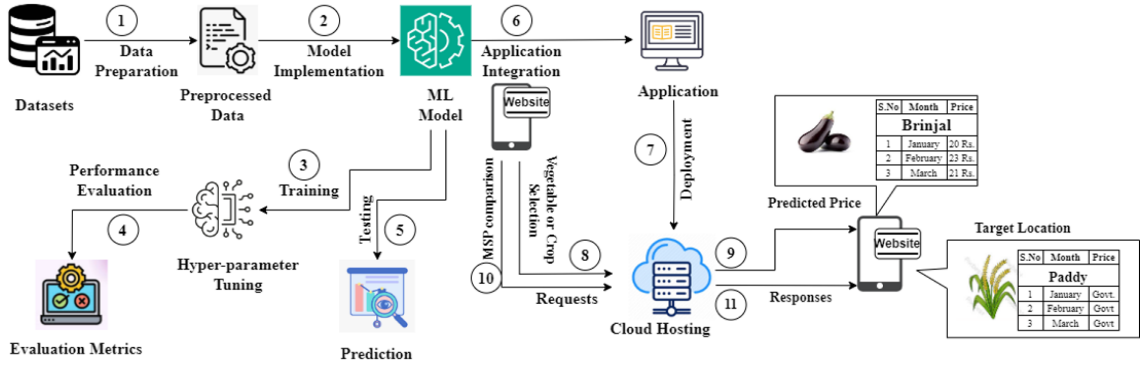

Fig. 1. Proposed system architecture for crops and vegetables

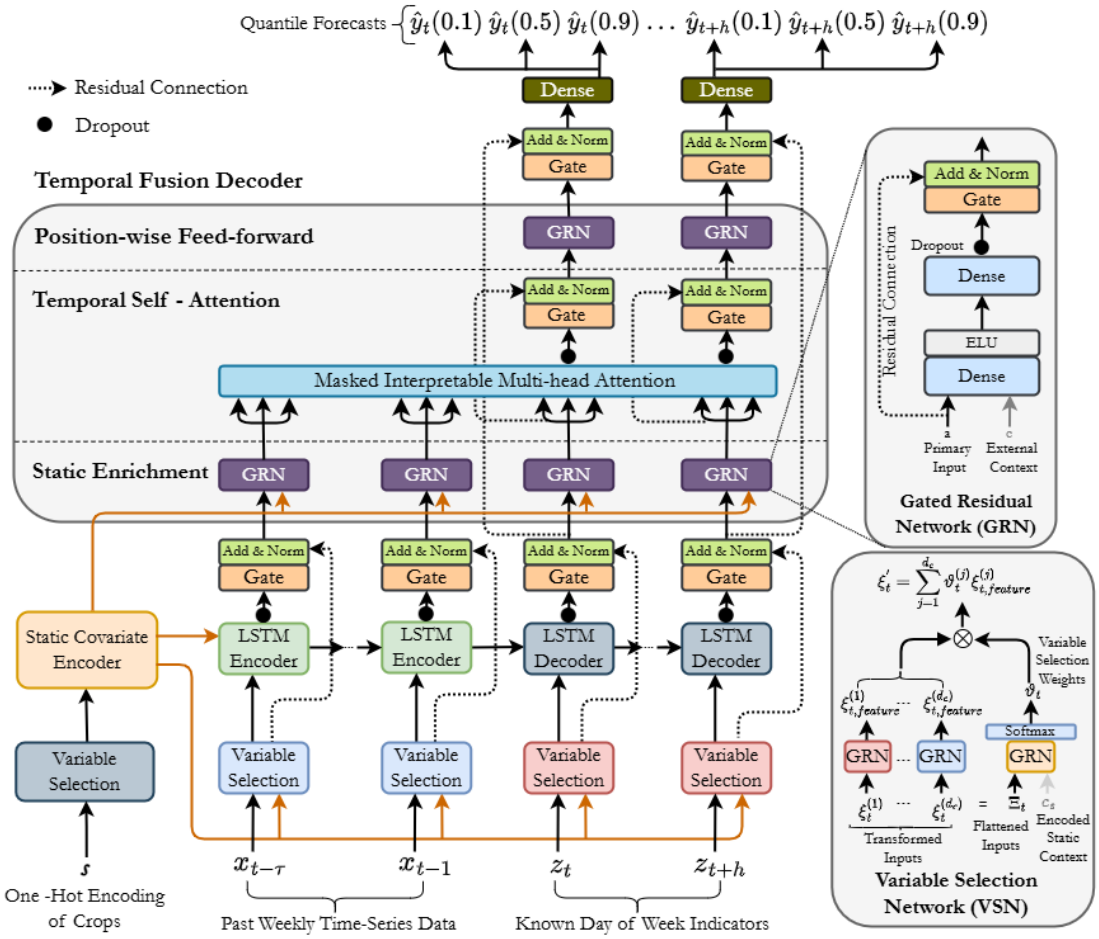

Fig. 2. TFT architecture for crop price prediction

2.2.1. *Data preparation*

a. Raw data

P_d: Daily market price (Rs.) of crop *i*.
R_d : Daily rainfall in mm.
T_d : Daily average temperature in °C.
z_d : One-hot encoded vectors for day *d*.

b. Data aggregation

Compute weekly wise average market price (P_w), rainfall (R_w) and temperature (T_w).

$$P_w = \frac{1}{7}\sum_{d=1}^{7} P_d \;,\; R_w = \frac{1}{7}\sum_{d=1}^{7} R_d \;,\; T_w = \frac{1}{7}\sum_{d=1}^{7} T_d \qquad (1)$$

c. Normalization

Z-score normalization scale daily price P_d over a week *j*, where $P_w^{(j)}$ and $\sigma_w^{(j)}$ the average market price and standard deviation of market price in week *j*.

$$P_{scaled} = \frac{P_d - P_w^{(j)}}{\sigma_w^{(j)}} \qquad (2)$$

d. Time series data

Weekly time-series data $x_t = [P_{scaled}, R_w, T_w] \in \mathbb{R}^3$, day-of-week vector $z_t \in \mathbb{R}^7$.

2.2.2. *Input specification*

a. Past Inputs: Historical time series for each crop $x_{t-\tau:t-1}$. τ is the lookback window, and *t* is the start time of the forecast.

$$x_{t-\tau:t-1} = \{x_{t-\tau}, \ldots, x_{t-1}\} \in \mathbb{R}^{\tau \times 3} \qquad (3)$$

b. Future Known Inputs: Pre-determined variables (e.g., day of week indicators), donated as $z_{t:t+h}$, where the forecast horizon is *h*.

$$z_{t:t+h} = \{z_t, \ldots, z_{t+h}\} \in \mathbb{R}^{h \times 7} \qquad (4)$$

c. Static Metadata: Time-invariant covariates, denoted as $s = \{s_1, s_2, \ldots, s_m\}$, where m is the number of static features (e.g., (m = 22) number of crops, s_1 = [1, 0, …, 0] for bitter gourd).

$$s = \{s_1, \ldots, s_m\} \in \mathbb{R}^m \qquad (5)$$

2.2.3. *Static context encoding*

Static metadata *s* is processed using a Gated Residual Network (GRN) to produce enriched static context $\boldsymbol{c_s} \in \mathbb{R}^{\boldsymbol{d_h}}$. It consists of two dense layers, an Exponential Linear Unit (ELU), Gate (*tanh* activation function), and Add and Norm layers.

$$e_s = \tanh(W_{sg} ELU(W_{s2} s + b_{s2}) + b_{sg}) \odot (W_{s1} ELU(W_{s2} s + b_{s2}) + b_{s1}) \qquad (6)$$

$$c_s = GRN_s(s) = LayerNorm(W_{s,res} s + e_s) \qquad (7)$$

$W_s, W_{s,res} \in \mathbb{R}^{d_h \times m}$, $W_{s1}, W_{sg} \in \mathbb{R}^{d_h \times d_h}$ and $b_{s1}, b_{s2}, b_{sg} \in \mathbb{R}^{d_h}$ are learnable weights and biases. $e_s \in \mathbb{R}^{d_h}$ is the gated output and d_h is dimensionality of the feature representations.

2.2.4. *Feature encoding*

Project past inputs $x_{t-\tau:t-1}$ to hidden continuous size (d_c) to get $x_{cont,t-\tau:t-1} \in \mathbb{R}^{\tau \times d_c}$. $W_{cont} \in \mathbb{R}^{d_c \times 3}$ and $b_{cont} \in \mathbb{R}^{d_c}$ are learnable weights and biases.

$$x_{cont,t-\tau:t-1} = W_{cont} x_{t-\tau:t-1} + b_{cont} \qquad (8)$$

Embed future known inputs $z_{t:t+h}$ to hidden continuous size (d_c) to get $z_{emb,t:t+h} \in \mathbb{R}^{h \times d_c}$. $W_{emb} \in \mathbb{R}^{d_c \times 7}$ and $b_{emb} \in \mathbb{R}^{d_c}$ are learnable weights and biases.

$$z_{emb,t:t+h} = W_{emb} z_{t:t+h} + b_{emb} \qquad (9)$$

2.2.5. *Variable Selection Networks (VSN)*

For each step *t* and input feature j (price, rainfall, temperature), variable selection weights $\boldsymbol{v_t^{(j)}} \in \mathbb{R}^{\boldsymbol{1}}$ are computed using SoftMax over weighted generating GRN output. $\boldsymbol{x_{cont,t}^{(j)}} \in \mathbb{R}^{\boldsymbol{1}}$ is the *j*-th projected feature at time *t*.

$$v_t^{(j)} = \frac{\exp(GRN_{j,weight}(x_{cont,t}^{(j)}, c_s))}{\sum_{k=1}^{d_c} \exp(GRN_{k,weight}(x_{cont,t}^{(k)}, c_s))} \quad (10)$$

$$e_{j,w} = \tanh\left(w_{jg,w} ELU\begin{pmatrix} W_{j2,w} x_{cont,t}^{(j)} + \\ W_{j3,w} c_s + \\ b_{j2,w} \end{pmatrix} + b_{jg,w}\right)$$
$$(W_{j1,w} ELU\begin{pmatrix} W_{j2,w} x_{cont,t}^{(j)} + \\ W_{j3,w} c_s \\ + b_{j2,w} \end{pmatrix} + b_{j1,w}) \quad (11)$$

$$GRN_{j,weight}(x_{cont,t}^{(j)}, c_s) = LayerNorm(x_{cont,t}^{(j)} + e_{j,w}) \quad (12)$$

The selected features are gated using a tanh activation function ($e_{j,w} \in \mathbb{R}^1$), added with the same features, and then normalized. $W_{j2,w} \in \mathbb{R}^{d_h \times 1}$, $W_{j3,w} \in \mathbb{R}^{d_h \times d_h}$, $W_{j1,w}$, $W_{jg,w} \in \mathbb{R}^{1 \times d_h}$, $c_s \in \mathbb{R}^{d_h}$, $b_{j2,w} \in \mathbb{R}^{d_h}$, and $b_{j1,w}, b_{jg,w} \in \mathbb{R}^1$ are learnable weights, static context, and biases.

$$e_{j,f} = \tanh\left(w_{jg,f} ELU\begin{pmatrix} W_{j2,f} x_{cont,t}^{(j)} + \\ W_{j3,f} c_s + \\ b_{j2,f} \end{pmatrix} + b_{jg,f}\right) \odot$$
$$(W_{j1,f} ELU\begin{pmatrix} W_{j2,f} x_{cont,t}^{(j)} + \\ W_{j3,f} c_s \\ + b_{j2,f} \end{pmatrix} + b_{j1,f}) \quad (13)$$

$$GRN_{j,feature}\left(x_{cont,t}^{(j)}, c_s\right) = LayerNorm(x_{cont,t}^{(j)} + e_{j,f}) \quad (14)$$

$$x_t' = \sum_{j=1}^{d_c} v_t^{(j)} GRN_{j,feature}\left(x_{cont,t}^{(j)}, c_s\right) \quad (15)$$

x_t' is the weighted combination of feature generated by GRN $GRN_{j,feature}\left(x_{cont,t}^{(j)}, c_s\right)$, yielding $x_{t-\tau:t-1}' \in \mathbb{R}^{\tau \times d_c}$. $W_{j2,f} \in \mathbb{R}^{d_h \times 1}$, $W_{j3,f} \in \mathbb{R}^{d_h \times d_h}$, $W_{j1,f}$, $W_{jg,f} \in \mathbb{R}^{d_c \times d_h}$, $b_{j2,f} \in \mathbb{R}^{d_h}$, and $b_{j1,f}$, $b_{jg,f} \in \mathbb{R}^{d_c}$ are learnable weights, and biases. $c_s \in \mathbb{R}^{d_h}$ and $e_{j,f} \in \mathbb{R}^{d_c}$ are the static context and feature generated by GRN.

$$u_t^{(j)} = \frac{\exp(GRN_{j,weight}(z_{emd,t}^{(j)}, c_s))}{\sum_{k=1}^{7} \exp(GRN_{k,weight}(z_{emd,t}^{(k)}, c_s))} \quad (16)$$

$$e_{j,w} = \tanh\left(w_{jg,w} ELU\begin{pmatrix} W_{j2,w} z_{emb,t}^{(j)} + \\ W_{j3,w} c_s + \\ b_{j2,w} \end{pmatrix} + b_{jg,w}\right)$$
$$(W_{j1,w} ELU\begin{pmatrix} W_{j2,w} z_{emb,t}^{(j)} + \\ W_{j3,w} c_s \\ + b_{j2,w} \end{pmatrix} + b_{j1,w}) \quad (17)$$

$$GRN_{j,weight}(z_{emb,t}^{(j)}, c_s) = LayerNorm(z_{emb,t}^{(j)} + e_{j,w}) \quad (18)$$

For each time step t and input feature j (day-of-week indicators), weights $u_t^{(j)} \in \mathbb{R}^1$ are computed using SoftMax over weighted generating GRN output. $z_{emd,t}^{(j)} \in \mathbb{R}^1$ is the j-th embedded feature at time t. $W_{j2,w} \in \mathbb{R}^{d_h \times 1}$, $W_{j3,w} \in \mathbb{R}^{d_h \times d_h}$, $W_{j1,w}$, $W_{jg,w} \in \mathbb{R}^{1 \times d_h}$, $c_s \in \mathbb{R}^{d_h}$, $b_{j2,w} \in \mathbb{R}^{d_h}$, and $b_{j1,w}, b_{jg,w} \in \mathbb{R}^1$ are learnable weights, static context, and biases.

$$e_{j,f} = \tanh\left(w_{jg,f} ELU\begin{pmatrix} W_{j2,f} z_{emb,t}^{(j)} + \\ W_{j3,f} c_s + \\ b_{j2,f} \end{pmatrix} + b_{jg,f}\right) \odot$$
$$(W_{j1,f} ELU\begin{pmatrix} W_{j2,f} z_{emb,t}^{(j)} + \\ W_{j3,f} c_s \\ + b_{j2,f} \end{pmatrix} + b_{j1,f}) \quad (19)$$

$$GRN_{j,feature}\left(z_{emb,t}^{(j)}, c_s\right) = LayerNorm(z_{emb,t}^{(j)} + e_{j,f}) \quad (20)$$

$$z_t' = \sum_{j=1}^{d_c} u_t^{(j)} GRN_{j,feature}\left(z_{emb,t}^{(j)}, c_s\right) \quad (21)$$

$W_{j2,f} \in \mathbb{R}^{d_h \times 1}$, $W_{j3,f} \in \mathbb{R}^{d_h \times d_h}$, $W_{j1,f}$, $W_{jg,f} \in \mathbb{R}^{d_c \times d_h}$, $b_{j2,f} \in \mathbb{R}^{d_h}$, and $b_{j1,f}$, $b_{jg,f} \in \mathbb{R}^{d_c}$ are learnable weights, and biases. $c_s \in \mathbb{R}^{d_h}$ and $e_{j,f} \in \mathbb{R}^{d_c}$ are the static context and feature generated by GRN. z_t' is the weighted combination of feature generated by GRN $GRN_{j,feature}\left(z_{emb,t}^{(j)}, c_s\right)$, yielding $z_{t:t+h}' \in \mathbb{R}^{h \times d_c}$.

2.2.6. Temporal processing with refinement

LSTM encoders encode a weighted combination of past inputs $x_{t-\tau:t-1}'$, and LSTM decoders decode a weighted combination of future inputs $z_{t:t+\tau}'$.

$$h_{t-\tau:t-1}, c_{t-1} = LSTM_{enc}(x_{t-\tau:t-1}') \quad (22)$$

$$h_{t:t+h}, c_{t:t+h} = LSTM_{dec}(z_{t:t+h}', h_{t-1}, c_{t-1}) \quad (23)$$

$h_{t-\tau:t-1} \in \mathbb{R}^{\tau \times d_h}$ are hidden states from τ LSTM encoders and $c_{t-1} \in \mathbb{R}^{d_h}$ is the cell state from the last LSTM encoder. d_h is dimensionality of the feature representations. $W_{ih} \in \mathbb{R}^{4.d_h \times d_c}$, $W_{hh} \in \mathbb{R}^{4.d_h \times d_h}$ and $b_h \in \mathbb{R}^{4.d_h}$ are input to hidden weights, hidden to hidden weights, and biases. $h_{t:t+h} \in \mathbb{R}^{h \times d_h}$ and $c_{t+t+h} \in \mathbb{R}^{d_h}$ are hidden states and cell states from h LSTM decoders. $W_{ih} \in \mathbb{R}^{4.d_h \times d_c}$, $W_{hh} \in \mathbb{R}^{4.d_h \times d_h}$ and $b_h \in \mathbb{R}^{4.d_h}$ are input to hidden weights, hidden to hidden weights, and biases.

$$g_{t-\tau:t-1} = \tanh(W_g h_{t-\tau:t-1} + b_g) \quad (24)$$

$$g_{t:t+h} = \tanh(W_g h_{t+t+h} + b_g) \quad (25)$$

$W_g \in \mathbb{R}^{d_h \times d_h}$ and $b_g \in \mathbb{R}^{d_h}$ are learnable weights and biases. $g_{t-\tau:t-1} \in \mathbb{R}^{\tau \times d_h}$ and $g_{t:t+h} \in \mathbb{R}^{h \times d_h}$ are modulated hidden states from LSTM encoders and decoders. Align $x_{t-\tau:t-1}'$ and $z_{t:t+h}'$ to the transformed hidden states $g_{t-\tau:t-1}$and $g_{t:t+h}$ dimension.

$$x_{t-\tau:t-1}'' = W_p x_{t-\tau:t-1}' + b_p \quad (26)$$

$$z_{t:t+h}'' = W_q z_{t:t+h}' + b_q \quad (27)$$

$W_p \in \mathbb{R}^{d_h \times d_c}$, $b_p \in \mathbb{R}^{d_h}$ and $x_{t-\tau:t-1}'' \in \mathbb{R}^{\tau \times d_h}$ are learnable weights, biases, and the past VSN output projection. Similarly, $W_q \in \mathbb{R}^{d_h \times d_c}$, $b_q \in \mathbb{R}^{d_h}$ and $z_{t+t+h}'' \in \mathbb{R}^{h \times d_h}$ are learnable weights, biases, and future known VSN output projection.

$$h_{t-\tau:t-1}' = x_{t-\tau:t-1}'' + g_{t-\tau:t-1} \odot h_{t-\tau:t-1} \quad (28)$$

$$h_{t-\tau:t-1}'' = LayerNorm(h_{t-\tau:t-1}') \quad (29)$$

$$h_{t:t+h}' = z_{t:t+h}' + g_{t:t+h} \odot h_{t:t+h} \quad (30)$$

$$h_{t:t+h}'' = LayerNorm(h_{t:t+h}') \quad (31)$$

$h_{t-\tau:t-1}' \in \mathbb{R}^{\tau \times d_h}$ and $h_{t:t+h}' \in \mathbb{R}^{h \times d_h}$ are the outputs after residual updation, where $\odot$ is the point-wise multiplication. $h_{t-\tau:t-1}'' \in \mathbb{R}^{\tau \times d_h}$ and $h_{t:t+h}'' \in \mathbb{R}^{h \times d_h}$ are the normalized outputs, where $LayerNorm(y)$ is $\frac{y-\mu}{\sqrt{\sigma^2+\epsilon}} \odot \gamma + \beta$. μ is mean, γ and β are the learnable parameters and ϵ is the tiny real number. $h_{t-\tau:t+h}''$ combines normalized past and future known output.

$$h_{t-\tau:t+h}'' = [h_{t-\tau:t-1}'', h_{t:t+h}''] \in \mathbb{R}^{(\tau+h) \times d_h} \quad (32)$$

2.2.7. Static enrichment

Integrate crop-specific context into temporal features. $W_{static,1}$, $W_{static,2}$, $W_{static,3}$, $W_{static,g} \in \mathbb{R}^{d_h \times d_h}$ and $b_{static,1}$, $b_{static,2}$, $b_{static,g} \in \mathbb{R}^{d_h}$ are the learnable weights and biases, respectively.

$$e_{static} = \tanh\left(W_{static,g} ELU\begin{pmatrix} W_{static,2} h_{t-\tau:t+h}'' \\ + W_{static,3} c_s \\ + b_{static,2} \end{pmatrix} + b_{static,g}\right)$$
$$\odot \left(W_{static,1} ELU\begin{pmatrix} W_{static,2} h_{t-\tau:t+h}'' \\ + W_{static,3} c_s \\ + b_{static,2} \end{pmatrix} + b_{static,1}\right) \quad (33)$$

$h_{t-\tau:t+h}''' \in \mathbb{R}^{(\tau+h) \times d_h}$ is the static enrichment of the normalized output $h_{t-\tau:t+h}''$. $e_{static} \in \mathbb{R}^{(\tau+h) \times d_h}$ is the gated output in the GRN Layer. $W_i \in \mathbb{R}^{d_h \times d_h}$ and $b_1 \in \mathbb{R}^{d_h}$ are the learnable parameters and biases, respectively.

$$h_{t-\tau:t+h}''' = GRN_{static}(h_{t-\tau:t+h}'', c_s) =$$
$$= LayerNorm(h_{t-\tau:t+h}'' + e_{static}) \quad (34)$$

2.2.8. Temporal self-attention

The multi-head attention operates on $h_{t-\tau:t+h}''' \in \mathbb{R}^{(\tau+h) \times d_h}$ to capture long-range dependencies, producing $a_{t-\tau:t+h} \in \mathbb{R}^{(\tau+h) \times d_h}$ and normalized residual output $a_{t-\tau:t+h}' \in \mathbb{R}^{(\tau+h) \times d_h}$.

$$a_{t-\tau:t+h} = MultiHead(h_{t-\tau:t+h}''') \quad (35)$$

$$MultiHead(h'''_{t-\tau:t+h}) = Attention(W_Q^i h'''_{t-\tau:t+h}, W_K^i h'''_{t-\tau:t+h}, W_V^i h'''_{t-\tau:t+h})W_o \quad (36)$$

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (37)$$

$$a'_{t-\tau:t+h} = LayerNorm(h'''_{t-\tau:t+h} + ELU(W_a a_{t-\tau:t+h} + b_a)) \quad (38)$$

W_Q^i, W_K^i, $W_V^i \in \mathbb{R}^{d_h \times \frac{d_h}{m}}$, W_0, $W_a \in \mathbb{R}^{d_h \times d_h}$ and $b_a \in \mathbb{R}^{d_h}$ are the learnable weights. *m* is the number of heads responsible for the independent attention mechanism. Query (Q) vectors to search for relevant information, Key (K) vectors to measure similarity with the query, and Value (V) vectors to provide the actual content, which is weighted and combined to produce the output.

2.2.9. Feed-forward transformation

It refines temporal representations with a residual connection from the post-LSTM temporal features. Apply GRN on the normalized residual output $a'_{t-\tau:t+h}$ to produce $f_{t-\tau:t+h} \in \mathbb{R}^{(\tau+h)\times d_h}$. $O_{t-\tau:t+h} \in \mathbb{R}^{(\tau+h)\times d_h}$ captures refined temporal features and serves as the final quantile forecasting layer.

$$e_{ff} = \tanh\left(W_{ff,g} ELU\begin{pmatrix} W_{ff,2} a'_{t-\tau:t+h} \\ +b_{ff,2} \end{pmatrix} + b_{ff,g}\right) \odot \left(W_{ff,1} ELU\begin{pmatrix} W_{ff,2} a'_{t-\tau:t+h} \\ +b_{ff,2} \end{pmatrix} + b_{ff,1}\right) \quad (39)$$

$$f_{t-\tau:t+h} = GRN_{ff}(a'_{t-\tau:t+h}) = LayerNorm(a'_{t-\tau:t+h} + e_{ff}) \quad (40)$$

$$O_{t-\tau:t+h} = LayerNorm(h''_{t-\tau:t+h} + ELU(W_0 f_{t-\tau:t+h} + b_0)) \quad (41)$$

$W_{ff,1}$,$W_{ff,2}$, $W_{ff,g} \in \mathbb{R}^{d_h \times d_h}$ and $b_{ff,1}$, $b_{ff,2}$, $b_{ff,g} \in \mathbb{R}^{d_h}$ are the learnable weights and biases, respectively. The residual connection from $h''_{t-\tau:t+h}$ skips static enrichment and attention layers, VSN-influenced temporal features (post-LSTM) to preserve raw temporal patterns (e.g., weekly price) before prediction.

2.2.10. Quantile forecasting

It predicts multiple percentiles (e.g., 10th, 50th, 90th) of future crop prices to quantify uncertainty.

$$\hat{y}_{t:t+h}(q) = W_d O_{t:t+h} + b_d \quad (42)$$

$O_{t:t+h} \in \mathbb{R}^{\tau \times d_h}$ and $\hat{y}_{t:t+h}(q) \in \mathbb{R}^{\tau}$ are output representations correspond to future time steps and weekly price forecasts per crop and quantile, respectively. $W_d \in \mathbb{R}^{1\times d_h}$ and $b_d \in \mathbb{R}^1$ are the learnable weights and biases.

2.2.11. Training and evaluation

Quantile loss $L_q(y,\hat{y})$ captures prediction errors across uncertainty levels and is calculated and averaged over crops, time steps, and quantiles.

$$L_q(y,\hat{y}) = max(q(y-\hat{y}),(q-1)(y-\hat{y})) \quad (43)$$

The TFT employs the Adam optimizer with a specified initial learning rate, batch size, number of epochs, and dropout regularization. It is evaluated using R^2, RMSE, MAE, and MAPE metrics. Performance is assessed on a test set for (m = 22) crops, comparing TFT forecasts to a seasonal ARIMA model.

2.2.12. TFT Interpretability

One of the advantages of TFT is its interpretability. The attention mechanisms and VSN permit the model to learn how to dynamically measure the relative importance of input features over time. The historical price behaviour was determined to play a dominant role in short-term predictions, while variability in aggregate variables such as temperature and rainfall played a stronger role during the seasonal transitions. This interpretability gives insight into the reasons of the price variations, which contributes to increase of transparency and trust in the model for practical decisions.

3. Results and discussion

3.1. Experimental setup

The TFT was deployed on a high-end system with an Intel Xeon CPU@2.00GHz, 32 GB DDR5 RAM, NVIDIA P100 GPU with 16GB memory, and Debian OS, to perform efficient computation. The 7.5 years of data for the 22 crops and vegetables were partitioned initially as 80% training and 20% testing sets, and subsequently another 60% training and 40% test split to check model performance one more time with chronological consistency while avoiding temporal leakage. Hyper parameter tuning with grid search was performed to refine important parameters, such as the learning rate at 0.001, batch size of 64, 50 epochs, hidden size of 32, attention head size of 4, dropout rate of 0.2, and hidden continuous size of 8. The systematic approach increases forecasting precision to make accurate price predictions and inform data-driven decisions in agricultural commodity markets.

3.2. Dataset preparation

A well-developed dataset of 7.5 yrs, ranging from January 2017 to July 2024 and covering 22 agricultural commodities like grains & vegetables from Krishna district in Andhra Pradesh, India was gathered to develop the crop/vegetable price forecasting model. Historical daily market price information was collected from the Patamata Rythu Bazar, in Vijayawada which is a primary agricultural local market. Meteorological data were collected from public and official safe sources to guarantee their solidity and consistency. Daily temperature data was recorded from Time and Date historical weather for Vijayawada from timeanddate.com and day wise rainfall measurements were obtained from the Andhra Pradesh Water Resources Information and Management System (APWRIMS) for the published monthly/annual rainfall statistics by India Meteorological Department (IMD).

Data were processed to ensure quality and reproducibility before modelling. Missing data were filled with interpolation, outliers were dropped out, numerical features are min-max normalized, categorical variables were one hot encoded and daily measures aggregated in weekly summaries. This preprocessing workflow guaranteed a uniform and well-organized dataset, which strengthens the robustness of the predictive model.

3.3. Results and analysis

The RMSE is calculated as the square root of mean squared differences between predictions and observations in terms of unit error. MAE is the average of the absolute errors between predicted and actual values, i.e., it represents the typical magnitude of a prediction error. MAPE calculates the mean of relative forecasting error in percentage, which is applicable to compare the relative accuracy of forecast estimates across data sets. RMSE, MAE are absolute errors, whereas MAPE takes into account the relative error. The R^2 measures the proportion of the variability in the dependent variable that is predictable from the independent variables, where a high value indicates a better fit. The result for the model based on two training/test splits of 80%–20% and 60%–40% is summarized in Table 1. For 80%-20% division the R^2 value is 0.85 which shows the good fit while RSME and MAPE, MAE are 99.13, 2.16% and 72.08 respectively. R^2 decreased to 0.79 and RMSE and MAE increased to 121.5 and 95.26 respectively as well as MAPE at a value of 2.5%, revealing a small decrease of predictive accuracy on the 60%-40% split. The 80%-20% division gave lower errors and a better model fit.

Table 1. Performance of TFT on [60%, 40%] and [80%, 20%] dataset splits for crop price prediction

Performance Metric	[Training – 60%, Testing – 40%]	[Training – 80%, Testing – 20%]
R^2	0.79	0.85
RMSE	121.5	99.13
MAPE	2.5%	2.16%
MAE	95.26	72.08

Figure 3 presents the real and forecasted prices of four crops-Groundnut, Bitter gourd, Cucumber, and Paddy-during 12 weeks. In the case of groundnuts, the forecasted prices are generally greater than the real prices. For instance, in Week 4, the real price was Rs. 5,972, while the forecasted price was Rs. 5,846. In the case of Bittergourd, the forecasts are near the actual prices in the initial weeks but are slightly higher in the later weeks. In Week 10, the actual price was Rs. 2,568, while the forecast price was Rs. 2,643. For Cucumber, the forecast prices are slightly less than in most weeks. In Week 11, for example, the actual price was Rs. 3,184, but the forecast price was Rs. 3,148. In the case of Paddy, the real prices rose from Rs. 3,247 during Week 1 to Rs. 3,794 during Week 12, while the forecasted prices were less than the actual values, reaching Rs. 3,520 during Week 12. The forecasts generally capture the trend, though there are differences, particularly for Groundnut and Paddy.

The observed and predicted prices of Groundnut, Bitter gourd, Cucumber, and Paddy for 12 weeks are shown in Figure 4. For Groundnut, the predicted prices are higher than the observed ones. For example, the actual price is Rs. 5,872, and the predicted price is Rs. 5,932 in week 4. In the case of Bittergourd, predictions are close to the actual prices in the early weeks but slightly higher beyond that time. For example, the true price is Rs. 2,568 and the forecasted price is Rs. 2,599 in week 10. For Cucumber, actual prices are in general higher than predicted ones, and the difference can be observed from Week 4, where the actual price is Rs. 3,223, but its prediction is slightly lower at Rs. 3,274. True prices for Paddy are Rs. 3,247 in Week 1 and increased to Rs. 3,794 in Week 12, with most predicted higher values. For example, the prediction is Rs. 3,751 compared to the true of Rs. 3,794 for week 12. Overall, the predicted ones are well in line with the prices observed. However, sometimes there is a certain deviation in some weeks.

Table 2 is a weekly comparison of crop prices between actual market prices, forecasted prices, and the MSP to identify the potentially more favourable selling option for farmers. For Paddy, the forecasted price always surpasses the actual price and the MSP, with the preferred option being in the local market. Conversely, the projected price of Groundnut is below the MSP for all but one week, which indicates that it would be more profitable to sell to the government. The forecasted prices for both crops follow the projected prices very closely, with the projected values being marginally higher, reflecting an optimistic bias in price forecasting. The analysis indicates alignment with the proposed decision strategy, such that Paddy is better suited for sale in local markets. At the same time, Groundnut is more suitable for government procurement.

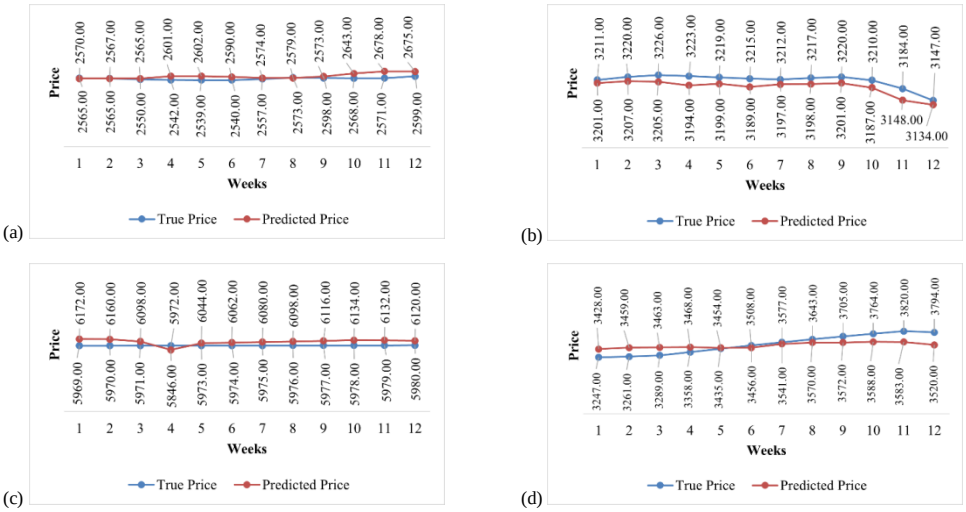

Fig. 3. Predicted crop prices using TFT compared to actual prices from January to March 2024 with an 80% train and 20% test data: (a) bitter gourd, (b) cucumber, (c) groundnut, (d) paddy

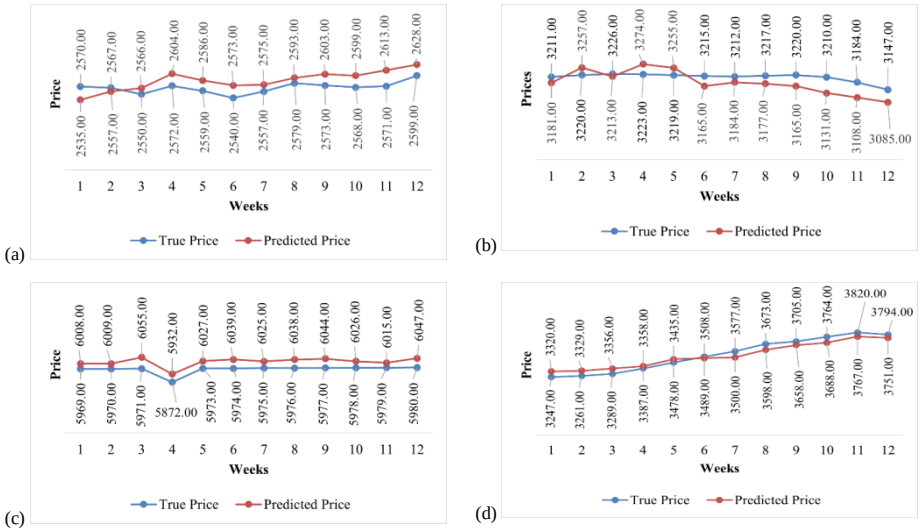

Fig. 4. Predicted crop prices (Rs.) using TFT compared to actual prices (Rs.) from January to March 2024 with a 60% train and 40% test data: (a) bitter gourd, (b) cucumber, (c) groundnut, (d) paddy

Table 2. Best-selling location based on predicted price and MSP

Week No.	Crop	Actual Price (Rs.)	Predicted Price (Rs.)	MSP (Rs.)	Best Selling Location	Decision
1	Paddy	3247	3428	2183	Local Market	✓
1	Groundnut	5969	6172	6377	Government	✓
2	Paddy	3261	3459	2183	Local Market	✓
2	Groundnut	5970	6160	6377	Government	✓
3	Paddy	3289	3463	2183	Local Market	✓
3	Groundnut	5971	6098	6377	Government	✓
4	Paddy	3289	3468	2183	Local Market	✓
4	Groundnut	5972	5846	6377	Government	✓

Table 3 compares crop price prediction models and the TFT's performance with 15 other existing methods across different datasets. Cheung et al. utilized 3D-CNN for the crop price prediction collected from the Investing.com dataset and obtained MAPE of 0.083, RMSE of 40.39, and MAE of 32.31 [3]. Xu et al. implemented NARNN and NARNN-X on the FAO dataset and obtained RRMSE values of 1.701% and 1.695%, respectively, for soybeans, and 1.777% and 1.775%, respectively, for soybean oil [23]. Soni et al. implemented Decision Tree Regressor and Random Forest on crop price data gathered from the government website data.gov.in, where Decision Tree Regressor obtained an accuracy of 92%, and Random Forest obtained an accuracy of 88% [19]. Kurumatani tested the TATP-LSTM on the Japanese Government Agricultural Price Movement dataset and obtained an RMSPE of 0.3464 and MAPE of 0.2645 [13]. Purohit et al. tested Multiplicative-ARIMA-SVM on forecasting agricultural product prices and obtained 0.83 and 0.98 correlation values for tomatoes and onions, respectively [17]. Harshith et al. used Stack-LSTM in the Indian Agriculture Market Information Network dataset achieving 3115.10 RMSE, 12.25% MAPE and 0.51 MDA [10]. Mahto et al. applied ARIMA to agricultural commodity prices collected from the data.gov.in and obtained 2.30% MAPE and 3.44% RMSPE [12]. Subhasree et al. used GANN on Tirupur New Bus Stand Ulavar Market data for vegetable price forecasting with an accuracy of 89% [21]. Nassar et al. implemented ATT-CNN-LSTM for Strawberry yield and farm pricing prediction using data accumulated from the California Strawberry Commission website, for which AGM-W2P was 23.53, W2Y was 0.03, and R^2 was 0.83 [16]. Yun et al. used grain price data from Investing.com and Kaggle and achieved future price prediction with an RMSE of 5.61, an MAE of 3.63, an MAPE of 0.55, and an R^2 of 0.9984 using Bi-DSConvLSTM-Attention [27]. Li et al. applied TPCDR on Vegetable Market Data in China, and the accuracy results turned out to be 88.6% (Natural Environment), 99.82% (Supply), 91.69% (Demand), and 93.56% (Economic Policy) [14]. Elbasi et al. used Bayesian Belief Network, Naïve Bayes, and Hoeffding Tree on the Kaggle crop recommendation dataset with accuracies of 99.59%, 99.46%, and 99.46%, respectively [6]. Yang et al. used FusFormer on the Mooney Viscosity Prediction Dataset, and the outcome was an RMSE of 3.37 and RRSE of 0.321 [25]. Avinash et al. used HM-GRU and HM-BiGRU on Agmarknet data, whose RMSE, MAE, and MAPE varied by week, i.e., week 1 with RMSE of 79.69, MAE of 61.72, and MAPE of 4.30% [1]. Guo et al. used AttLSTM-ARIMA-BP in China's agricultural big data website with MAPE of 0.0043, MAE of 1.51and RMSE of 1.642 [9]. Lastly, our study employed TFT on APSPDS, Rythu Baazar, and Vijayawada data with an RMSE of 99.13, MAPE of 2.16%, MAE of 72.08, and accuracy of 93.24%.

Harshith et al. applied Stack-LSTM in the Indian Agriculture Market Information Network dataset with 3115.10 RMSE, 12.25% MAPE, and 0.51 MDA [10]. Mahto et al. applied ARIMA to agricultural commodity prices collected from the data.gov.in and obtained 2.30% MAPE and 3.44% RMSPE [12]. Subhasree et al. used GANN on Tirupur New Bus Stand Ulavar Market data for vegetable price forecasting with an accuracy of 89% [21]. Guo et al. employed AttLSTM-ARIMA-BP on China's agricultural big data website, with MAPE of 0.0043, MAE of 1.51, and RMSE of 1.642 [9].

Table 3. Performance comparison between TFT and existing implementations for crop price forecasting

Literature	Model/ Algorithm	Dataset	Performance
Cheung et al. [3]	3D-CNN	Data collected from Investing.com	MAPE: 0.083, RMSE: 40.39, MAE: 32.31
Xu et al. [23]	NARNN, NARNN-X	Food and Agriculture Organization (FAO)	**Soybeans** NARNN: RRMSE – 1.701%, NARNN-X: RRMSE – 1.675% **Soybean Oil** NARNN: RRMSE – 1.777%, NARNN-X: RRMSE – 1.775%
Soni et al. [19]	Decision Tree Regressor, Random Forest	India Government Website data.gov.in	**Decision Tree Regressor:** Accuracy – 92% **Random Forest:** Accuracy – 88%
Kurumatani [13]	TATP - LSTM	Japanese Government Agricultural Price Movement Dataset	RMSPE – 0.3464, MAPE – 0.2645
Purohit et al. [17]	Multiplicative-ARIMA-SVM	National Horticulture Board, India	**ARIMA-SVM:** correlation: 0.83 (tomatoes), correlation: 0.98 (onions)
Harshith et al. [10]	STACK-LSTM	Indian Agriculture Market Information Network	RMSE: 3115.10, MAPE: 12.25%, MDA: 0.51
Mahto et al. [12]	ARIMA	India Government Website data.gov.in	MAPE: 2.30%, RMSPE: 3.44%
Subhasree et al. [21]	Genetic Algorithm Neural Network (GANN)	Tirupur New Bus Stand Ulavar Market	Accuracy: 89%
Nassar et al. [16]	ATT-CNN-LSTM	California Irrigation Management Information System	AGM-W2P: 23.53, W2Y: 0.03, R^2: 0.83
Yun et al. [27]	Bi-DSConvLSTM-Attention	Data collected from Investing.com and Kaggle	RMSE: 5.61, MAE: 3.63, MAPE: 0.55, R^2: 0.9984
Li et al. [14]	TPCDR	China's Vegetable Market Data	Accuracy: 88.6% (Natural Environment), 99.82% (Supply), 91.69% (Demand), 93.56% (Economic Policy)
Elbasi et al. [6]	Bayesian Belief Network, Naïve Bayes and Hoeffding Tree	Global Crop Market Data	**Bayesian Belief Network:** Accuracy – 99.59%, **Naïve Bayes and Hoeffding Tree:** Accuracy – 99.46%
Yang et al. [25]	FusFormer	Mooney Viscosity Prediction Dataset	RMSE: 3.37, MAPE: 0.321
Avinash et al. [1]	HMM-GRU, HMM-BiGRU	Agmarknet	**HM-GRU:** ***Week 1:*** RMSE – 79.69, MAE – 61.72, MAPE – 4.30%, ***Week 4:*** RMSE – 113.86, MAE – 91.29, MAPE – 6.49%, ***Week 8:*** RMSE – 157.07, MAE – 123.60, MAPE – 9.10%, **HM-BiGRU:** ***Week 12:*** RMSE – 182.19, MAE – 151.52, MAPE – 11.18% (Long-term prediction)
Guo et al. [9]	AttLSTM-ARIMA-BP	China's agricultural big data website	MAPE: 0.0043, MAE: 1.51, RMSE: 1.642
Our Work	TFT	APSPDS, Local Rythu Baazar, Vijayawada	RMSE: 99.13, MAPE: 2.16%, MAE: 72.08, R^2: 0.85, Accuracy: 93.24%

3.4. Prototype implementation

A website has been carefully crafted with a high emphasis on functionality and user experience to generate real-time crop price forecasts to facilitate accessibility and ease of use for farmers, traders, and policymakers. Figure 5 provides a homepage overview highlighting market prices that can be expected to facilitate decision-making. The site also features an MSP comparison tool so that users can assess whether marketing to local markets or government buyers would yield the greatest profit.

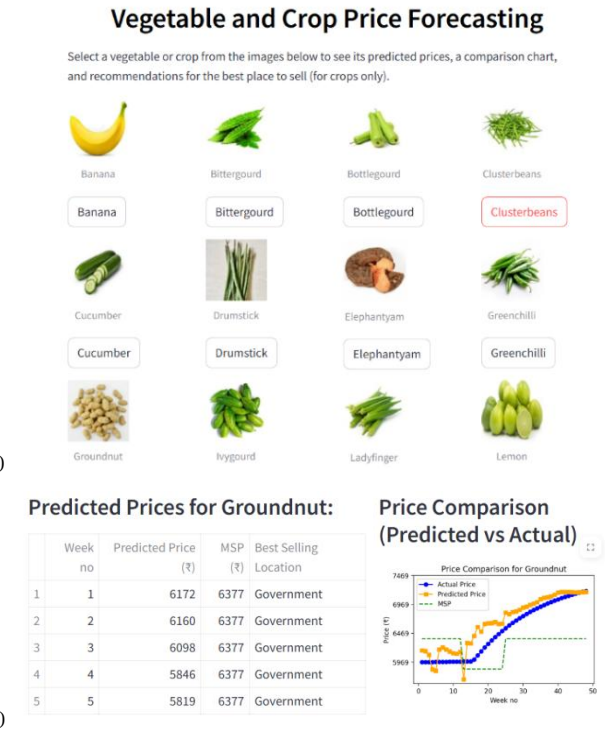

Fig. 5. Vegetable and crops prediction application; (a) homepage, (b) price comparison and selling locations

4. Conclusion

The variations in the market prices of agri-horticultural commodities have remained a challenge to the farmers of Krishna District, Andhra Pradesh and often lower their profit as well decision. A TFT model for forecasting the prices of 22 vegetables was developed using historical local Rythu Bazaar market price data along with meteorological parameters, viz., rainfall and temperature in this study. The developed model performed well with the accuracy of 93.24%, R^2 of 0.85, RMSE of 99.13, MAPE of 2.16%, and MAE of 72.08, which is very competitive relative to the existing studies in terms of accuracy and reliability. An important feature of the system is the MSP comparison with local market rates so that farmers can choose to sell their produce in nearby markets or to government procurement agencies if it results into better return. In addition to the correct price prediction, the model is helping farmers with better crop activities positioned so that it minimizes risk in fluctuating markets. Considering its scalability, the system can easily be ported to other regions and expanded to incorporate new crops or influencing factors and thus provides an extensible approach for price prediction in agriculture.

References

[1] Avinash, G., Ramasubramanian, V., Ray, M., Paul, R. K., Godara, S., Nayak, G. H. H., Kumar, R. R., Manjunatha, B., Dahiya, S., & Iquebal, M. A. (2024). Hidden Markov guided Deep Learning models for forecasting highly volatile agricultural commodity prices. *Applied Soft Computing*, *158*, 111557. https://doi.org/10.1016/j.asoc.2024.111557

[2] Bernal, E. A., Yang, S., Herbst, K., & Venuto, C. S. (2024). Comparing machine learning and deep learning models to predict cognition progression in Parkinson's disease. *Clinical and Translational Science*, *17*(11), e70066. https://doi.org/10.1111/cts.70066

[3] Cheung, L., Wang, Y., Lau, A. S. M., & Chan, R. M. C. (2023). Using a novel clustered 3D-CNN model for improving crop future price prediction. *Knowledge-Based Systems*, *260*, 110133. https://doi.org/10.1016/j.knosys.2022.110133

[4] Chintapalli, P., & Tang, C. S. (2022). The implications of crop minimum support price in the presence of myopic and strategic farmers. *European Journal of Operational Research*, *300*(1), 336–349. https://doi.org/10.1016/j.ejor.2021.09.034

[5] Das, R. (2020). Minimum support price in India: What determines farmers' access? *Agricultural Economics Research Review*, *33*(1), 61–69. https://doi.org/10.5958/0974-0279.2020.00007.5

[6] Elbasi, E., Zaki, C., Topcu, A. E., Abdelbaki, W., Zreikat, A. I., Cina, E., Shdefat, A., & Saker, L. (2023). Crop Prediction Model Using Machine Learning Algorithms. *Applied Sciences*, *13*(16), 9288. https://doi.org/10.3390/app13169288

[7] Fayer, G., Lima, L., Miranda, F., Santos, J., Campos, R., Bignoto, V., … Goliatt, L. (2023). A Temporal Fusion Transformer Deep Learning Model for Long-Term Streamflow Forecasting: A Case Study in the Funil Reservoir, Southeast Brazil. *Knowledge-Based Engineering and Sciences*, *4*(2), 73–88. https://kbes.journals.publicknowledgeproject.org/index.php/kbes/article/view/7983

[8] Fouquet, D., Grotz, C., Sawin, J., & Vassilakos, N. (2005). Reflections on a possible unified EU financial support scheme for renewable energy systems (RES): A comparison of minimum-price and quota systems and an analysis of market conditions. *Brussels and Washington DC*, *26*(5).

[9] Guo, Y., Tang, D., Tang, W., Yang, S., Tang, Q., Feng, Y., & Zhang, F. (2022). Agricultural Price Prediction Based on Combined Forecasting Model under Spatial-Temporal Influencing Factors. *Sustainability*, *14*(17), 10483. https://doi.org/10.3390/su141710483

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

[11] Kauser, N. T. P., Ashwitha, M. R., & Sultana, S. (2024). Enhancing crop yield prediction through machine learning techniques. In *47th Series Student Project Programme (SPP)*, Bengaluru, India. https://www.kscst.org.in/spp/47_series/47s_spp/Exhibition%20Projects/235_47S_MSC_0133.pdf

[12] KumarMahto, A., Biswas, R., & Alam, M. A. (2019). Short Term Forecasting of Agriculture Commodity Price by Using ARIMA: Based on Indian Market. In M. Singh, P. K. Gupta, V. Tyagi, J. Flusser, T. Ören, & R. Kashyap (Eds), *Advances in Computing and Data Sciences* (Vol. 1045, pp. 452–461). Springer Singapore. https://doi.org/10.1007/978-981-13-9939-8_40

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

[14] Li, Y., Yao, J., Song, J., Feng, Y., Dong, H., Zhao, J., Lian, Y., Shi, F., & Xia, J. (2024). Investigation of causal public opinion indexes for price fluctuation in vegetable marketing. *Computers and Electrical Engineering*, *116*, 109227. https://doi.org/10.1016/j.compeleceng.2024.109227

[15] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

[16] Nassar, L., Okwuchi, I. E., Saad, M., Karray, F., Ponnambalam, K., & Agrawal, P. (2020). Prediction of Strawberry Yield and Farm Price Utilizing Deep Learning. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–7. https://doi.org/10.1109/IJCNN48605.2020.9206998

[17] Purohit, S. K., Panigrahi, S., Sethy, P. K., & Behera, S. K. (2021). Time Series Forecasting of Price of Agricultural Products Using Hybrid Methods. *Applied Artificial Intelligence*, *35*(15), 1388–1406. https://doi.org/10.1080/08839514.2021.1981659

[18] Sengupta, S., Basak, S., Saikia, P., Paul, S., Tsalavoutis, V., Atiah, F., Ravi, V., & Peters, A. (2020). A review of deep learning with special emphasis on architectures, applications and recent trends. *Knowledge-Based Systems*, *194*, 105596. https://doi.org/10.1016/j.knosys.2020.105596

[19] Soni, N., & Raut, J. (2022). Crop price prediction using machine learning techniques. *International Advanced Research Journal in Science, Engineering and Technology*, *9*(5), 757–762.

[20] Stulec, I., Petljak, K., & Bakovic, T. (2016). Effectiveness of weather derivatives as a hedge against the weather risk in agriculture. *Agricultural Economics (Zemědělská Ekonomika)*, *62*(8), 356–362. https://doi.org/10.17221/188/2015-AGRICECON

[21] Subhasree, M., & Arun Priya, C. (2016). Forecasting vegetable price using time series data. *International Journal of Advanced Research*, *4*(2), 535–541.
[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (Version 7). arXiv. https://doi.org/10.48550/ARXIV.1706.03762
[23] Xu, X., & Zhang, Y. (2022). Soybean and Soybean Oil Price Forecasting through the Nonlinear Autoregressive Neural Network (NARNN) and NARNN with Exogenous Inputs (NARNN–X). *Intelligent Systems with Applications*, *13*, 200061. https://doi.org/10.1016/j.iswa.2022.200061
[24] Yan, W., Cai, Y., Lin, F., & Ambaw, D. T. (2021). The Impacts of Trade Restrictions on World Agricultural Price Volatility during the COVID-19 Pandemic. *China & World Economy*, *29*(6), 139–158. https://doi.org/10.1111/cwe.12398
[25] Yang, Y., & Lu, J. (2022). A Fusion Transformer for Multivariable Time Series Forecasting: The Mooney Viscosity Prediction Case. *Entropy*, *24*(4), 528. https://doi.org/10.3390/e24040528
[26] Yin, H., Jin, D., Gu, Y. H., Park, C. J., Han, S. K., & Yoo, S. J. (2020). STL-ATTLSTM: Vegetable Price Forecasting Using STL and Attention Mechanism-Based LSTM. *Agriculture*, *10*(12), 612. https://doi.org/10.3390/agriculture10120612
[27] Yun, B., Lai, J., Ma, Y., & Zheng, Y. (2024). Research on Grain Futures Price Prediction Based on a Bi-DSConvLSTM-Attention Model. *Systems*, *12*(6), 204. https://doi.org/10.3390/systems12060204
[28] Zheng, P., Zhou, H., Liu, J., & Nakanishi, Y. (2023). Interpretable building energy consumption forecasting using spectral clustering algorithm and temporal fusion transformers architecture. *Applied Energy*, *349*, 121607. https://doi.org/10.1016/j.apenergy.2023.121607

B.Tech. Dedeepya Manikonda
e-mail: dedeepyamanikonda@gmail.com

Dedeepya Manikonda is persuing B.Tech in Computer Science and Engineering from Siddhartha Academy of Higher Education (Deemed to be University), Vijayawada. She has an interest in the application of machine learning to solve real life problems, emphasizing on agriculture and sustainability. Her project on modelling agricultural price predictions using the Temporal Fusion Transformer exemplifies this vision. She has a practical exposure in the internships that she done under MVC Development and API Development, with certifications on AI, Deep Learning, Linux, IoT and Azure AI. Her love for learning and creativity make her a bright star as a "contributor" in social technologies.

https://orcid.org/0009-0006-8562-0509

Ph.D. Ashutosh Satapathy
e-mail: ashutosh.satapathy1990@gmail.com

Ashutosh Satapathy is currently working in the Department of Computer Science and Engineering as an assistant professor at Siddhartha Academy of Higher Education, Deemed to be University, Vijayawada. He completed his Ph.D. at Vellore Institute of Technology, India, and his M. Tech. in Information Security and Computer Forensics at SRM University, India. During his Ph.D. program, he completed a research internship at Information and Communications Research Laboratories, ITRI-Taiwan. Prior to this, he worked as a lecturer at Odisha University of Technology and Research (formerly College of Engineering and Technology, Bhubaneswar). He has published 23 research articles, including international journal articles, book chapters, and conference proceedings, all indexed in SCIE (WoS) or Scopus.

https://orcid.org/0000-0002-2531-4079

B.Tech. Keerthi Padamata
e-mail: padamatakeerthi2004@gmail.com

Keerthi Padamata is pursuing B.Tech in CSE from Siddhartha Academy of Higher Education (Deemed to be University), Vijayawada, India and an Amazon Future Engineer Scholar. She has recently done an intern at Amazon as software development engineer, Bangalore. Her research interests are on software systems, time-series forecasting and deep learning with application to agricultural price prediction. She is currently working on data-driven models for decision support in agri-food and socio-economic contexts.

https://orcid.org/0009-0003-7958-6621

B.Tech. Jaswanthi Machcha
e-mail: jaswanthi1804@gmail.com

Jaswanthi Machcha is software development engineer at Amazon works on building and scaling reliable software systems. She finished her B.Tech. in computer science and engineering from Velagapudi Ramakrishna Siddhartha Engineering College. Her technical interests lie in distributed systems, cloud computing, backend programming, and performance. Through her work she's built considerable expertise on designing APIs, building scalable service applications and maintaining large production systems.

https://orcid.org/0009-0005-1112-2083

Ph.D. J. Chandrakanta Badajena
e-mail: chand.cet@gmail.com

J. Chandrakanta Badajena is currently working as an assistant professor in the Department of Information Technology at Odisha University of Technology and Research, Bhubaneswar, Odisha, India. He obtained his Ph.D. degree in computer science engineering and application from Utkal University, Odisha, India. He has more than 15 years of teaching experience at both undergraduate and postgraduate levels. His research interests encompass brain computer interaction, emotion recognition, human computer interaction, computer and network security, cognitive radio networks, IoT and cloud computing. He has published extensively in international journals on topics including brain computer interfaces, emotion recognition computer network and security and IoT based systems.

https://orcid.org/0000-0002-3184-1908