Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 12(125)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111100298

Tomato price forecasting using a stacked LSTM network with global scaling CNN and RC layer

G Sumaiya Farzana[1] and N Prakash[2*]
Department of Computer Science and Engineering, B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India[1]
Department of Information Technology, B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India[2]

Received: 15-March-2024; Revised: 12-April-2025; Accepted: 14-April-2025

Abstract
Accurate price prediction is essential across various sectors such as real estate, e-commerce, and finance. Traditional methods often struggle with high-dimensional data and nonlinear patterns. Recent advancements in machine learning (ML), particularly deep learning (DL) techniques such as convolutional neural networks (CNN) and collaborative models like StackNet, offer promising solutions. The proposed approach leverages advanced DL techniques, specifically the global scaling convolutional neural network (GS-CNN) and the StackNet rapid curing (RC) Layer. In this framework, long short-term memory (LSTM) models are organized in a stacked configuration to reduce kernel size and enable effective information flow between layers, thereby achieving optimal predictions with lower error margins for tomato price forecasting. The performance of the proposed model is evaluated using standard error metrics, including mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and the coefficient of determination (R^2). The experimental results indicate that the model achieves an MSE of 0.0046, RMSE of 0.06, MAPE of 0.2753, MAE of 0.05, and an R^2 value of 0.83. These outcomes confirm the effectiveness and adaptability of the proposed approach in accurately forecasting tomato prices with minimal error rates. However, limited knowledge of underlying market dynamics may still constrain prediction accuracy. While the proposed model demonstrates low error rates, further improvement is possible by incorporating larger and more diverse datasets to enhance predictive performance.

Keywords
Price prediction, Deep learning, StackNet-RC, Global scaling CNN, Tomato forecasting, Stacked LSTM.

1.Introduction
Inflation has emerged as a leading concern among Indian policymakers and citizens over the last decade [1]. As agriculture is the backbone of the country's development, data mining and decision making regarding these commodities take up a major role in appropriate price forecasting [2]. Agricultural product cost plays a significant part in the horticultural markets. In India, among vegetables, tomatoes have a large supply and significant price variations [3]. As tomatoes are grown both indoors and outdoors throughout the year, it is crucial to settle the tomato inventory and the costs involved [4]. Moreover, forecasting vegetable prices is challenging due to numerous factors, such as weather and crop production rates. These also have both nonlinear and a non-stationary category of characteristics [5]. Tomatoes are indispensable vegetables in food, encompassing higher nutritional values [6].

On the other hand, India is a vast country, having agriculture as its backbone, sharing an overall Gaussian differential privacy (GDP) rate of 20.2% from 2020-2021 [7]. However, these ranges have declined due to various reasons such as excess rain, drops occurring in the market, and many others. Among all these factors, the prediction of prices and related commodities is one of the major issues among farmers [8].

The accurate price prediction of agricultural goods is one of the quintessential aspects of ensuring the proper economic functioning worldwide [9]. Recent research reveals that the application of deep learning

*Author for correspondence

(DL)-based tasks to agricultural price forecasting is comparatively scarce [10]. Moreover, the effectual prediction involves the tasks of lack of experience among the farmers involved in farming, as well as in the instability and the impact upon the price flow from the agricultural and the economic factors [11]. The view on the unstable nature of the market and the difficulties involved in obtaining the agricultural sectors. These challenges may result in the people involved in agriculture earning comparatively lower profit rates and discouraging themselves from being involved in agricultural practices [12]. Thus, an accurate forecast and prediction of crop growth rates are related to both scheduling and planting, determining the machine harvest time, and predicting the crop yield [13]. The computational tools and, the forecasting practices and the prices for fresh produce have been accomplished using traditional time series and several artificial intelligence (AI) based approaches [14]. Concurrently, building effective and high-precision prediction models for fresh produce (FP) amid the market price is crucial in aspects protecting retailers from overpriced FPs [15].

The agricultural sector relying on the Indian economy is in need of more technological support for both their development and growth [16]. Effective price prediction of agricultural products and commodities helps ensure the farmers with good returns and recovery of their investments [17]. Thus, several studies have intended to make use of machine learning (ML) models to deal with successful modeling over the time series data. However, making use of ML models is scarce in India for vegetable price forecasting [18]. The suggested approach intended to make use of historical data in view of accomplishing the price of vegetables [19]. However, it is one of the essential aspects to acknowledge the models are subjected to hold some of the limitations regarding the underlying view of assumptions. Thus, the regression-based model has been used as a part of the ML-based technique, along with the long short-term memory(LSTM) model, to predict the crop price [20]. Whereby random forest (RF) has been accomplished in the suggested approaches to forecasting the results of various crops in the areas of Thailand [21].

Though several approaches are involved in price prediction and forecasting, the study of vegetable prices has limitations and pitfalls, such as higher error rates, false predictions, and less ability to handle multiple and varying rates of data aiming to produce the right prediction. Thus, to address these challenges and limitations, the proposed study aims to achieve effective prediction using a robust DL-based approach, incorporating a global scaling convolutional neural network (GS-CNN) within the convolutional layer (CL) and integrating layered memory networks within the StackNet-RC layers. These are used in aspects of resolving the two vital issues such as the optimization of the kernel size and in the effectual output retrieval from one LSTM layer to the following layer. The overall approach intending to perform the price prediction of tomatoes is evaluated using the performance metrics encompassing the mean squared error (MSE), root mean squared error (RMSE), R-squared (R^2) rates, and mean absolute error (MAE).

The primary objective of this study is to accurately forecast future tomato prices, which is achieved through the following key steps:

1. To collect daily tomato price data and perform essential pre-processing techniques to handle missing values, normalize features, and prepare the data for effective model learning.
2. To develop the proposed StackNet-RC model integrated with a GS-CNN, and train it using an 80:20 train-test split to ensure robust learning and generalization.
3. To evaluate the model's performance using regression-based metrics such as R^2, RMSE, MSE, and MAE, aiming to achieve high prediction accuracy with minimal error rates.

The remainder of the paper is organized into four sections. Section 2 discusses existing models used for predicting tomato and other vegetable prices, employing various datasets with ML and DL approaches. Section 3 presents the comprehensive methodology adopted in the study, along with the corresponding algorithms. Section 4 provides a detailed analysis of the results and outcomes obtained from the implementation. Finally, Section 5 concludes the study and outlines potential directions for future work.

2.Literature review

The suggested approach has been using various forecasting models such as the box-jerkins-based regression model, and ML models for predicting the retail stage for some of the selective vegetables. Upon comparison, the support vector regression (SVR) and the LSTM models have been procuring more effectual results than other forecasting models [22]. Web-crawler have been used in the collection of large-scale data, whereas autoregressive integrated moving average (ARIMA) models have been used in aspects of forecasting horticultural products. These models have been used in forecasting tomato, cucumber, and eggplant prices for both long and short-term periods [23]. Whereas the optimized neural network (NN) model has been used in the price forecasting of vegetables, where the model is oriented to the ordered weighted averaging (OWA) technique. These are also combined with the functions of optimization, using generalized neural network (GRNN) and the basis function recurrent neural network (RNN). This, in turn, reduces the model construction process and enhances the accuracy [24].

Due to many such nonlinear capabilities, ML models such as artificial neural network (ANN), support vector machine (SVM), and RF have been employed for the aspects of time-series data-based forecasting [25]. It has been demonstrated that these models are superior to the conventional econometric models involved in forecasting agricultural product prices [26]. Accurate crop price forecasting approaches are of vital need in enabling supply chain planners and the other associated government bodies to take appropriate actions by estimating various market factors such as demand and supply [27]. Whereby onion price prediction has been done using the ML-based approaches encompassing the SVM, RF, Naïve Bayes (NB), and the NN. The ranges of prices from low to high have been estimated using the suggested approach [28]. Followed by the ARIMA model has been used in the onion price predictions. The data have been used from the years 2003-2017, where the cost of price predicted were in the ranges from 2956-1651. From January-June 2018 [29].ARIMA model with the ANN has been combined to predict the price of onion in the Banglore market from Jan –June 2018. The effectual outcomes of the model have been less applicable for onion price forecasting with an RMSE rate of 3.79 [30].

The existing model [31] has constructed a four-layered CNN followed by a max pooling layer(PL). The model has been divided into validation and training datasets in the ratio of 20:80, has been trained under 105 epochs, and has achieved greater accuracy. Likewise, In [32], for predicting the tomato price in Varanasi, India, the study has developed models, namely generalized AutoRegressive conditional heteroskedasticity (GARCH) (1,1), ANN (14-8-1), and ARIMA (3,1,4). Amongst these methods, the ANN (14-8-1) approach has been found to be the finest predicting method. To analyze and predict the association of tomato arrivals and also cost among two major markets in the Kurnool, average pooling (AP), ARIMA approach has been applied to predict daily data that has been gathered from the selected markets over 3 years from Jan 2019 to Dec 2021. The outcome has revealed that there has been a 1% level of significant negative association between the arrival of tomatoes and price in the AP [33]. Furthermore, [34] has employed the ML method, particularly the LSTM method, to forecast the price; LSTM has been examined utilizing the weekly field cost of tomatoes for the past 10 years from 2011- 2022. The outcome of the research has shown that LSTM has acquired less error value than the other traditional models.

Additionally, to predict the tomato price, data has been derived from six various regions in the northwestern Peloponnese, created 33 various dataset inputs, and developed a decision support system (DSS) along with K-nearest neighbors (KNN), which has offered the best for the created dataset. The results have found that the implementation of DSS in real-time data has been a valuable tool for prediction[35]. Similarly, to predict the normal cost of tomatoes, the SVM method, along with polynomial feature alteration, has been made. Minimum and maximum pricing attributes have been applied, and to prove the performance as well as consistency of the model, the dataset has been standardized, and to evaluate the performance of the method, MSE and also R^2 score metrics have been employed. The outcome has shown that the model has the potential to predict the average price of a tomato when employed with polynomial feature transformation [36]. For this reason, [37] has presented an approach to construct a strong crop price forecast method by considering different characteristics like historical cost along with market arrival quantity of crop, history of weather data which has affected the production of the crop, and then transportation and quality of data associated to features gained through the performing statistical analysis. Two various time sequences have been utilized to represent the experimental results, and the outcome has shown that the model has better accuracy metrics than the other models [38].

Concomitantly, in [39] to demonstrate the correlation between the price sequence of tomatoes in chosen markets known as agricultural produce market committee (APMC) in Haryana, multi-dimensional time sequence methods like vector autoregressive (VAR) and then vector autoregressive integrated moving average (VARMA) have been employed. Performance of the methods has been compared utilizing Relative Deviation (%), mean absolute percentage error (MAPE) and then standard error prediction (SEP) for various predictive horizons like 1,3,6,9 and 12 months. The results have shown that the model has attained better than the VAR method for predicting the price of tomato. In addition, the existing research [40] has utilized the Central Statics Agency data to attain statistics regarding Indonesia's statistics on tomato plants from 2015-2020, and these data have been solved utilizing the Fletcher-Reeves algorithm through the architectural methods like 2-35-1, 2-30-1, and 2-10-1. The results have proved that model 2-10-1 has the finest architecture, and it has been utilized to calculate the accuracy of the Fletcher-Reeves method along with the MSE training set at 0.00008463. Another view is that [41] has provided a hybrid method that has been combined with ARIMA, SVR, and linear statistical analysis for data of time sequence. Similarly, [42] has planned a CNN-LSTM model for strawberry price prediction. The analysis of the results has revealed that model 3 has shown enhanced results with a 13.37% difference from the normal observation than models 1 and 2.

Furthermore, the prior research [43] has aimed to explore different ML algorithms like gradient boosting machine (GBM), RF, GRNN, and SVR to predict the wholesale price of brinjal in 17 major cities of Odisha, India. An empirical comparison of the predictive accuracy of various methods has been analyzed and has shown that the GRNN has performed better than the other models. Likewise, the monthly price of recant in Kerala has been predicted utilizing the time series as well as the ML model. ML models such as seasonal autoregressive integrated moving average (SARIMA), LSTM, and Holt-Winter's Seasonal model have been utilized, and then the performance of these models has been calculated based on the RMSE value on the dataset with cost from 2007 to 2017. The outcome has exhibited that the LSTM model has been found to be the best model than the other methods [44]. Additionally, [45] has aimed to design a wheat price forecasting model, which utilizes the weather, price, production, and then consumption trends for the cost of wheat, which has been taken over the past few years, and this has been examined utilizing LSTM technique and has shown significant improvement than the other conventional ML methods.

The review highlights various models used in vegetable price forecasting, with ML techniques like SVR, LSTM, ANN, and RF outperforming traditional econometric methods. ARIMA and hybrid models have also been applied, though often with higher error margins. Recent approaches integrate NNs with optimization techniques and DSSs for improved accuracy. Studies show LSTM and ANN consistently achieve superior results in forecasting tomato and other vegetable prices across diverse regions and datasets.

3.Proposed methodology

Although vegetable prices have a significant impact on the population, their volatile and dynamic nature makes accurate price prediction challenging. This inconsistency creates difficulties in forecasting future prices reliably. Nevertheless, vegetable price prediction remains a crucial task, enabling the general public to anticipate price trends in advance. There are many such types of research focusing on improved forecasting models, which are more accurate by using both the statistical and computational dorms of approach, such as ML and DL. The existing and several ML-based approaches is intended to show comparatively lower performance rates, being less accurate with the dynamic changing nature of prices, unable to deal with time series datasets, and also less adaptable to the fluctuating price rates prediction and forecasting. The overall flow of the projected approach is shown in *Figure 1*.

Figure 1 represents the workflow for predicting tomato prices and begins with a dataset that contains the daily price of tomatoes. The first step involves preprocessing the data, which prepares it by cleaning, normalizing, or transforming it into a suitable format for analysis and modeling. Following this, the data is divided into two sets. The training data, comprising 80% of the dataset, is used to train the predictive models, while the test data, making up the remaining 20%, is utilized to evaluate the model's performance. The prediction model itself may involve a DL or ML architecture with multiple layers and structure, as suggested by the figure's detailed computational design. Finally, the trained model generates predictions and future tomato's process. This structured approach emphasizes systematic data processing and model training to ensure robust and reliable predictions.

The complete design of the proposed framework is illustrated in *Figure 1*. The proposed model utilizes two distinct approaches within the CNN architecture to predict and forecast tomato prices. The process begins with retrieving the daily tomato price dataset, followed by effective data preprocessing techniques, including the removal of missing values and the transformation of raw data into structured and machine-readable formats. The dataset is processed in aspects of looking for the missing values, removing the noise and other redundant attributes associated with the data and the other redundant values associated with the data. After pre-processing of data, the next step is splitting the data, here the data is split into two sub-set, training and testing. Where 80% of the data is trained in the learning model, and the remaining 20% is utilized to evaluate the final performance of the model. The pre-processed and refined data are fed into the proposed regression model, which integrates CNN and the StackNet-RC layer. This design aims to minimize kernel sizes and facilitate effective, layer-by-layer input propagation through the stack-wise arranged LSTM layers. These components are primarily responsible for generating accurate predictions with minimal error rates. The models employed in the proposed approach are discussed in the following subsections.

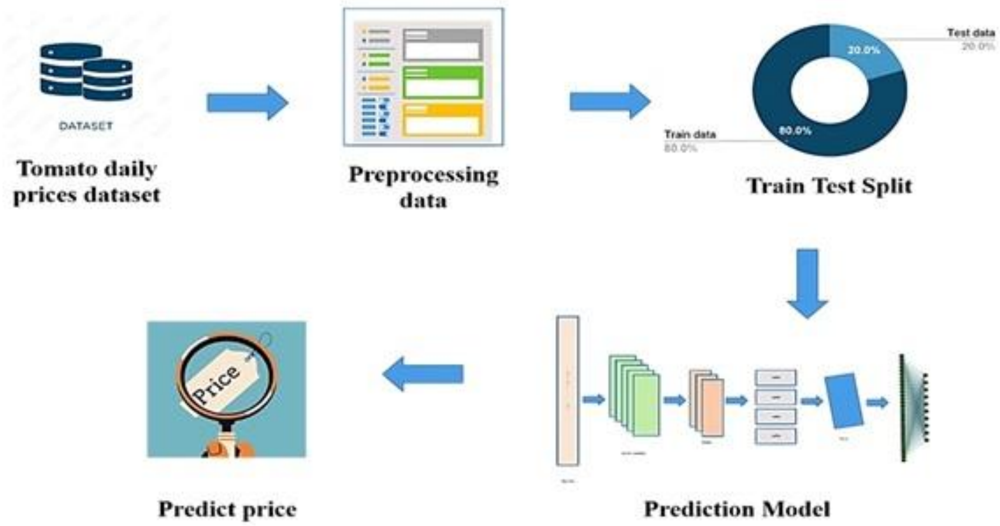

Figure 1 Architectural illustration of the projected approach

3.1Pre-processing

Noise reduction techniques are more important to analyze the price data efficiently because they assist in mitigating short-term fluctuations as well as revealing long-term trends. One of the common techniques is the smoothing method, which includes averaging the cost data over a particular number of days. This improves the clarity of trends by filtering the noise. Another important method is outlier detection and removal, which particularly utilizes the Z-score technique. By measuring the Z-scores for every price point, analysts can categorize and then eliminate outliers that alter the data. Furthermore, data transformation, like a logarithmic scale to price data, is utilized to stabilize the variance and also to mitigate the influence of extreme values. Thus, these techniques all together offer a more accurate analysis of price movement.

Noise denotes variances or random errors that are calculated in variables. It is obtained from different sources like calculation errors or inappropriate information. This may lead to inaccuracies during investigations. The overall results can be improved significantly by managing the missing values by filling them in statistical calculations such as median or mean or by removing the records with missing values.

3.2CNN

CNNs are a fundamental DL architecture widely used in solving optimization problems, particularly in image processing and pattern recognition tasks. A typical CNN consists of the following key components:

- Convolutional layer (CL)
- Pooling layer (PL)
- Flatten layer (FL)
- Fully connected layer (FC)
- Activation function (AF)

The test image, across various pixel size ranges, uses several filters to capture essential features. The CL is specifically designed for processing 1D sequence data; this layer contains 32 filters, meaning it extracts 32 distinct feature maps from the input. Then, the global scaling layer is a custom layer that employs a global scaling operation, and the reshape layer is utilized to restructure the output from the prior layer. Finally, the dense layer is a FC layer that provides the final prediction. The PL tends to reduce the overall dimensionality of the image features, which are extracted using the process of either Max pooling (MP) or AP approaches. In the case of MP, the maximal range of values is considered, whereas, in aspects of AP, the average range of values is considered among the filter region. The FL is intended to convert the output of the preceding layer to a 1D array, which is from the input of the FC layer. Using the feature vector array, the FC layer tends to perform the process of classification, and the overall output is then provided to the output layer. After the FC, AF is processed, this utilizes the rectified linear unit (ReLU) AF, and the kernel size is set to 3, which means that every filter looks at 3 continuous steps in the input data.

3.2.1GS-CNN

CNN is able to capture the important features from the provided input in an automated approach, especially from the image inputs, which are compared to the multi-layered perceptron. The effectual performance and the accuracy of the CNN among image recognition are highly adaptable to the other conventional approaches. One of the major challenges associated with CNN is the number of images and their kernel size, which results in a huge range of training phases. Additionally, the process of hyper-parameter tuning is also an inevitable approach for obtaining optimal performance outcomes. The overall architectural workflow of the proposed StackNet-RC integrated with GS-CNN is illustrated in *Figure 2*.

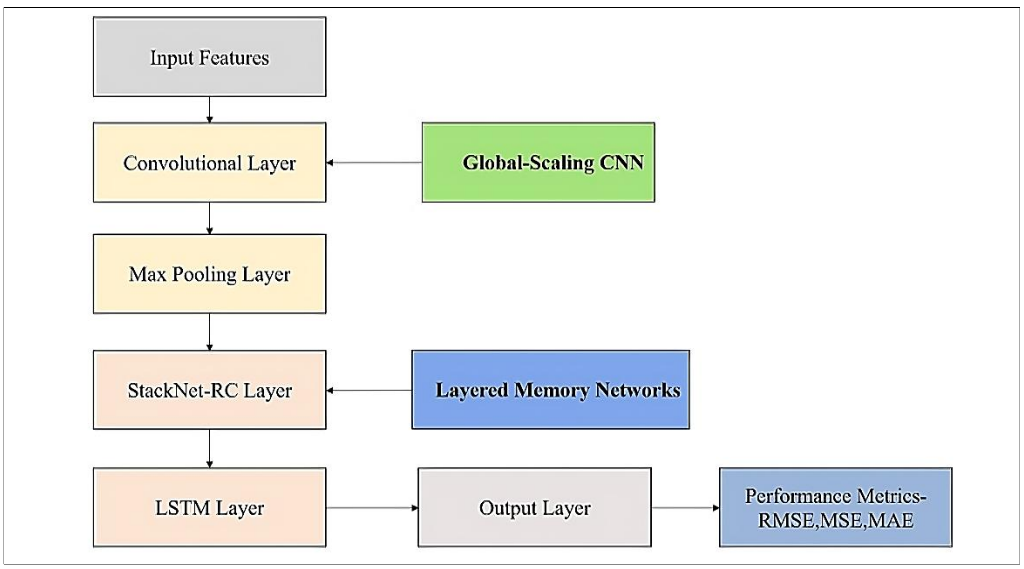

Figure 2 Working on the proposed StackNet-RC with GS-CNN

The proposed StackNet-RC model, illustrated in *Figure 2* and integrated with a GS-CNN, processes input features to generate predictions evaluated using performance metrics such as RMSE, MSE, and MAE. The workflow begins with raw input data, which is scaled and analyzed through CL to extract relevant features. A max PL subsequently reduces the dimensionality by retaining the most significant values, thereby enhancing computational efficiency. At the core of the architecture, a stacked layer integrates stacking and recursive computation through a layered memory network, effectively capturing historical patterns. The incorporated LSTM layer models long-term dependencies in sequential data, allowing the system to learn complex temporal relationships. Finally, the output layer produces predictions, which are assessed using standard evaluation metrics to determine the model's accuracy and robustness.

In the first two ranges of CL, the kernel sizes are from a prime range of 1 to$N$, being varied from one layer to the other layer. In the third CL, the final layer conducts the convolutional operation with a kernel size of either 1 or 2. GS-CNN is utilized for numerous reasons. This integrates a global scaling mechanism that permits the method to learn a scaling factor for inputs dynamically, which helps the features while minimizing the noise. Then, the CL efficiently captures the local design in the data. Through combining convolutional operations along with global scaling, GS-CNN improves the process of feature extraction. This technique also enhances the generalization by decreasing the over fitting of particular features. Thus, it enhances the ability of the method to generalize to unseen data. Then, the StackNet-RC layer reduces the size of kernels in CLs, permitting the capture of more effective local features and also decreasing the computational complications, making the model more effective. In the combination of GS-CNN along with the StackNet-RC layer the proposed method aims to improve the accuracy as well as effectiveness of tomato price predictions. The GS-CNNs, with their corresponding fields, are able to wrap all the possible integers rather than the odd numbers. The selection of kernel size is one of the critical aspects of the CNN model. Most of the current 1D-CNN models are in view of treating the kernel size as the hyper-parameter, which makes use of the grid search in detecting the optimal kernel size range. However, most of the time series are with multiple salient signals, which are in need of multiple kernel sizes involved in capturing them correspondingly. This makes the combination of multiple kernel sizes result in the unexpected rise of exponential increase in the computational burden for a grid search. Moreover, the kernel size should be considered as a part of the learning process rather than the process of searching the hyper parameter. The working algorithm of the GS-CNN is tabulated in Algorithm-I.

Algorithm-I-GS-CNN

Input: feature
Output: prediction
Data: training data
// function for convolution
1 Function Conv(activation, weights):
2 $c^l \longleftarrow s^{l-1} * Wg^l$
3 conv $\longleftarrow c^l$
4 return conv
$L(c) = L(c^1) + L(c^2) - 1,$
$c(K) = 1, c(K^1) = \min(K^1, K^2),$
for $\forall X, K, \exists$ K1, K2 makes $\mathrm{Conv}(X, K) = \mathrm{Cove}(\mathrm{Conv}(X, K1), K2)$
// function for max pooling
5 Function Pool(activation(l thlayer)):
6 $s^l_{xy} \longleftarrow \max_{(i=0,\dots,n,j=0,\dots,n)} s^{l-1}{}_{(x+i)(y+i)}$
7 return s^l_{xy}
// function for the fully connected layer
8 Function FC(activation, learnable parameters):
9 $c^l \longleftarrow Wg^l \cdot s^{l-1}$
10 return c^l
// Main program calls function
11 Function Main(input feature):
12 Conv $\longleftarrow$ input image
13 Pool $\longleftarrow$ Conv
14 FC $\longleftarrow$ Pool
15 Output $\longleftarrow$ FC
16 return Output

Thus, to overcome the limitations of computational complexity and the computational burden, the model wrapping all the scales of 1D-CNN, where the optimal size range of kernel is selected in an automated manner during the learning phase. This is considered to be the GS-CNN. These are adapted in aspects of learning the multiple kernel sizes on an effective basis, which are also easy to implement by making minor modifications to the current 1D-CNN in the CL layer. Using all of the prime numbers, which are less than or equal to the ranges of $N = data\ length/4$, are considered to be the effectual and considerable kernel size. Using the setting, half of the input signal is long enough to make a catch over the positional data, which are in a relative position to each of the edges.

Learning rate: It is a hyper parameter that decides the size of each step at every iteration when approaching a minimum loss function during the training process. It controls the method and how much to be altered in response to the expected error for every updated weight of the method. The learning rate of the present method starts at the highest value and then reduces over time; this assists the method to converge more effectively as it meets an optimal solution.

Dropout Rate: This is a regularization method that is utilized to avoid over fitting in NNs. During the process of training, at every update, this dropout sets a fraction of the input units to zero to make the method more efficient. The dropout rate of the present model is 0.2. That is, 20% of the neurons are dropped during the training process. This process helps to decrease the over fitting.

Batch Size: The batch size denotes the number of training that is utilized in one reiteration. The batch size of the weight updated model after each training is denoted by 1, and this process leads to a more precise gradient estimate.

Number of Epochs: Epoch denotes one complete process of both forward as well as backward of every training through the method. The number of epochs describes the total number of learning algorithms that have worked throughout the entire training dataset. The present model has trained for 10 epochs, which means that the training data has been passed 10 times throughout the model.

3.3LSTM

LSTM networks are an extension of RNNs with an inherent capability to forecast time series data. This recurrent layer is widely used for sequence prediction and, in the proposed model, consists of 64 units configured to return sequences. LSTMs are designed to store long-term temporal dependencies and effectively map input sequences to corresponding outputs. Unlike traditional perceptron's, the LSTM architecture includes memory cells and gating mechanisms such as input, forget, and output gates—which regulate the flow of information. These gates enable the network to retain relevant data over extended time intervals. *Figure 3* illustrates the architecture of the LSTM model.

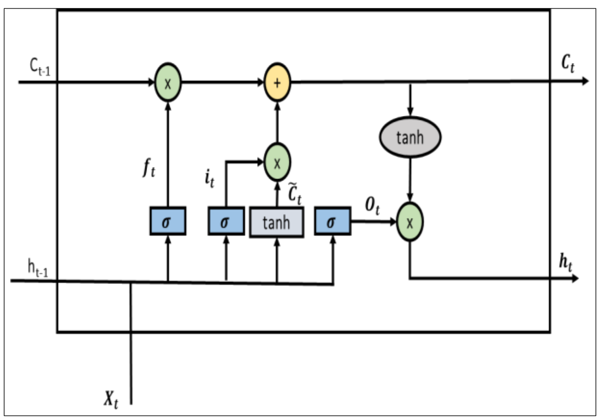

Figure 3 Architecture of LSTM

This model incorporates input, forget, and output gates along with internal cell memory, which collectively regulate the flow of information. The forget gate determines which information should be discarded from the cell state, using a sigmoid AF to compute retention. The input gate controls the addition of new information into the cell state by processing incoming inputs and gate activations. The output gate regulates the information to be passed on to the next layer by applying both sigmoid and tanh AFs, thereby producing the final output of the LSTM cell. The AF used in this layer is ReLU. With 64 units, the layer extracts 32 distinct feature maps from the input data.

Thus, the overall function of the LSTM is as follows:
(1) Computing the output of LSTM using forward learning.
(2) Error computation amid the overall resultant data and the provided input data from each layer.
(3) The computed errors are propagated in reverse through the network using back propagation, allowing each gate to be individually updated during training.
(4) Relying on the ranges of error, the values of weight in each of the gates are updated using the approach used for optimization.

These procedure are carried out and are repeated for a provided iteration until the ideal range of values of weights and the biases are attained.

3.3.1Bi –LSTM

Bidirectional long short-term memory (Bi-LSTM) is a variant of the LSTM model. This is a special network structure that is formed by superimposing two LSTM layers. The LSTMs simultaneously train the input data at the same time. These two LSTM layers vary in that one-layer inputs the data in a positive temporal order, whereas the other layer processes it in a reverse temporal order. *Figure 4* depicts the architecture of the Bi-LSTM model.

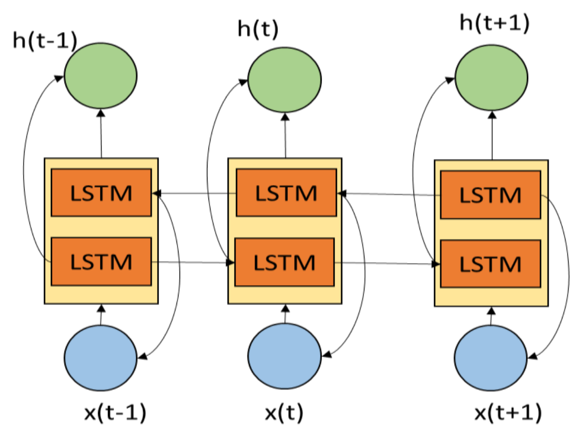

Figure 4 Overall architecture of Bi-LSTM

Bi-LSTM consists of two parallel LSTM layers: a forward layer and a backward layer. The forward layer processes the sequence in chronological order, capturing past context, while the backward layer processes the sequence in reverse, capturing future context. Together, they allow the model to learn information from both directions. The output sequence is generated by combining the outputs from the forward pass (ascending in time) and the backward pass (descending in time).

This bidirectional structure enhances the temporal modeling capability of the recurrent network by incorporating dependencies from both past and future time steps.

3.3.2GRU architecture

Gated recurrent units (GRUs) are considered more efficient and effective than LSTMs, as they simplify the LSTM architecture by reducing computational complexity. GRUs achieve this by combining the forget and input gates into a single update gate, enabling faster training while still addressing the issue of long-term dependencies (LTD). This streamlined structure enhances performance in sequence modeling tasks while preserving the core benefits of LSTM networks. Comparing this with LSTM, GRU performs faster as well as applies low memory consumption as it can capture long-term dependencies. For these reasons, the GRU is considered more effective than the LSTM method. *Figure 5* illustrates the architecture of the GRU model.

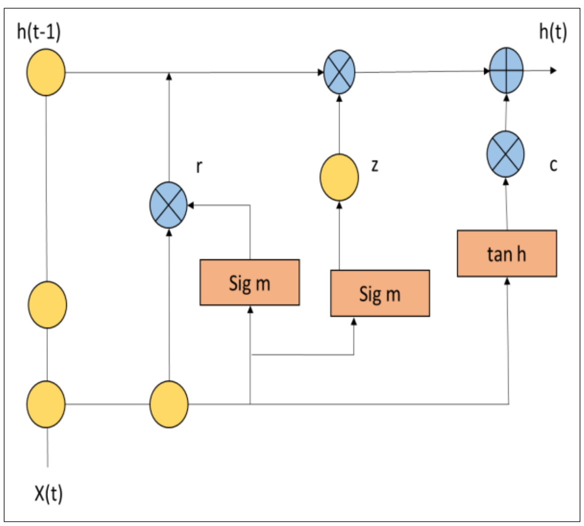

Figure 5 Architecture of GRU model

Figure 5 represents the architecture of the GRU model, which is an NN designed for sequential data processing, such as time series analysis and natural language processing. It utilizes inputs at each time step, denotes as x(t), and maintains a hidden state where h(t-1) is a previous state and h(t) is the updated state. Key components included the reset gate(r), which uses a sigmoid AF to determine how much of the previous hidden state to forget, and the update gate(z), which controls the extent to which the new candidate state (c)replaces the previous state.

The candidate activation combines the reset gate and current input using a tanh AF for non-linearity.

The final hidden state is a weighted combination of the previous state and candidate state, influenced by the update gate. This architecture allows GRUs to efficiently model long-term dependencies while simplifying the structure compared to LSTM units, resulting in faster training and reducing over fitting risk.

GRU gate is similar to LSTM with a forget gate, but it has fewer parameters; even though it contains fewer parameters than the LSTM method, it does not have an output gate. The process of GRU is found to be more efficient for NLP language. Initially, the series is read by the reset gate, and then the input sequence is updated by applying the update gate only if the input is considered to be necessary. This passes the information and then waits for the upcoming input. In this method, the cells consist of input as well as forget gate. In which the input and the forget gate are controlled by utilizing a gate controller.

3.3.3StackNet-RC layers

The StackNet-RC is the functionality of standard LSTM, which is utilized to stack numerous LSTM layers on top of each other. This method improves the ability to understand the complicated patterns as well as the relationships inside the data. It obtains the output of the max PL and then processes it further through capturing more temporal dependencies. Whereas in the temporal feature extraction, the LSTM layers are temporal dependencies and then patterns through utilizing sequential data. This understands the temporal features for predicting or for sequence categorization by stacking the LSTM methods. This StackNet-RC method can learn various levels of abstraction from the input data. This process utilizes the output of the max PL to accomplish better performance metrics results.

The aspects of interaction among the modality information are less considered, resulting in the final multimodal features being obtained as unenviable forms of information, which are irrelevant to the tasks of detection. Thus, the StackNet-RC layers have been used in aspects of many such applications. Here, if the modality of the information is not measured, the output modality of the features that are obtained is with large amounts of redundant data, which are also irrelevant to the detection of the task. The adaptive algorithm for StackNet-RC layer is tabulated in Algorithm-II.

Algorithm-II-StackNet-RC Layer
Function LSTM (xt, et−1) returns et
Local Variables:
- ft, it, ot, Ct: intermediate LSTM gate outputs
- All variables are in R^n
Model Weight Matrices:
- Wf, Wi, Wo, Wc ∈ R^{nxm}
Model Bias Vectors:
- bf, bi, bo, bc ∈ R^n
Gate Computations:
ft = sigmoid(Wf·xt + Uf·et−1 + bf)
it = sigmoid(Wi·xt + Ui·et−1 + bi)
ot = sigmoid(Wo·xt + Uo·et−1 + bo)
Attention Matrices:
A1 = $\omega_v \cdot (\omega_t)^T$
A2 = $\omega_t \cdot (\omega_v)^T$
Attention Weights:
M1 = softmax(A1)
M2 = softmax(A2)
Contextual Representations (O1, O2 ∈ R^{uxdm}):
O1 = M1 · ω_t
O2 = M2 · ω_v
Context Vectors:
C1 = O1 ⨀ ω_v
C2 = O2 ⨀ ω_t
Combined Context Matrix:
Cmatv = $(C1)^T \oplus (C2)^T$ // ⊕ denotes concatenation
Cell State Update:
Cmatv = ft ⨀ Cmatv_prev + it ⨀ tanh(Wc·xt + Uc·et−1 + bc)
Output:
et = ot ⨀ tanh(Cmatv)

Thus, a stacked framework is needed. Thus, the stacked form of RC layers has been used in the projected approach. Here, the input provided in the time into the initial layer of LSTM, which are with the values of feature metrics, are multiplied in aspects of obtaining stacked RC layers. This is shown in Equation 1.

$$A1 = \omega v \cdot (\omega t)T, A2 = \omega t \cdot (\omega v)^T \quad (1)$$

Using the Line-by-line SoftMax function, the distribution of attention is also estimated, where $M_1(i,j)$ are indicated as the correlation amid $the\ i$ and j feature vectors. The greater the value, the stronger the range of interactions seen among the features and among the essential feature information. This is provided using Equation 2.

$$M1 = \mathrm{softmax}(A1), M2 = \mathrm{softmax}(A2) \quad (2)$$

Finally, the attention mechanism (AM) has been multiplied using the feature matrix in aspects of obtaining the final AM matrix. This is obtained by Equation 3.

$$O1, O2 \in Ru \times dm . O1 = M1 \cdot \omega t, O2 = M2 \cdot \omega v \quad (3)$$

Finally, the multiplication of gate mechanisms is used in aspects of obtaining the mutual form of AM information matric among each of the individual modalities as per Equation 4.

$$C1 = O1 . \omega v, C2 = O2 . \omega t \quad (4)$$

Thus, transporting the O_1 and O_2 to the desired dimensions and concatenating them are fused to obtain the information representation. Here, the ⊕ represents the concatenation of the operation vectors. This is obtained by using Equation 5

$$Cmatv = (C1)T \oplus (C2)T \quad (5)$$

The mentioned process is repeated until the final values of the LSTM in the projected model are involved in calculating the resultant value. One of the main benefits of the StackNet-RC layers is that each layer of the model, in aspects of time series, which are then passed to the following block till the final accrued lock is involved in computing the output. Thus, the concatenation of the CNN among the CL layer with the advantage of GS and the StackNet-RC layers with the layered form of memory network endures in obtaining effectual outcomes with an effective prediction of tomato price with considerably lower ranges of error rates.

4.Results and discussion

This section presents the outcomes of the proposed approach, including a detailed description of the dataset used, the overall experimental results, and the Exploratory Data Analysis (EDA). These elements are discussed to demonstrate the effectiveness and robustness of the proposed model.

4.1Dataset collection

The data used in this study were obtained from the Tomato Daily Price dataset, which covers the period from 2013 to 2021. The dataset contains daily records of tomato prices, including minimum, maximum, and average price values for each day. Specifically, the maximum price indicates the highest price of tomatoes on a given day, the minimum price reflects the lowest, and the average price represents the mean value for that day. These price metrics are essential for analyzing trends and patterns in tomato pricing over time, facilitating accurate future price predictions.

The dataset was sourced from the following link: https://www.kaggle.com/datasets/ramkrijal/tomato-daily-prices.

4.2Performance metrics

For the evaluation, the following performance metrics have been utilized:

RMSE

It is used to compute the average variance amongst values in the classification and actual values. The RMSE formula is presented in Equation 6.

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{N} (\text{actual} - \text{predicted})^2} \;/N \quad (6)$$

RMSE measures the average magnitude of errors between predicted and actual values. It penalizes larger errors more heavily due to squaring the differences. A lower RMSE indicates better model performance. N represents the total number of observations or data points in the dataset.

MAE

It is denoted as the average of the absolute dissimilarity between predicted and actual values. Equation 7 shows the calculation of MAE.

$$\mathrm{MAE} = \sum_{i=1}^{n} |y_i - x_i| \,/n \quad (7)$$

MAE calculates the average of the absolute differences between predicted (y_i) and actual (x_i) values. Unlike RMSE, it treats all errors equally without squaring them. n refers to the total number of observations or data points in the dataset.

R-squared (R^2)

The R-squared is the numerical calculation that indicates the proposition of variance for the dependent variable with the independent variable.

MSE

MSE is a metric used to evaluate the accuracy of predictions by measuring the average squared difference between actual values and predicted values. A lower MSE indicates better model performance, as values closer to zero represent minimal error. The formula for MSE is shown in Equation 8

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - Y_i)^2 \quad (8)$$

MSE calculates the average squared difference between actual (Yi) and predicted (Yi') values. Like RMSE, it penalizes larger errors more heavily but does not take the square root.

4.3EDA

EDA is employed to examine the dataset using various methodologies for effective data visualization. Additionally, EDA helps in identifying patterns and validating assumptions inherent in the data. These objectives are achieved through graphical representations and statistical summaries. Overall, EDA plays a crucial role in gaining a comprehensive understanding of the dataset.

Figures 6 and *7*, which present the box plot and violin plot of average annual tomato prices, illustrate significant fluctuations in tomato pricing over the years. The visualizations reveal that the average price of tomatoes peaked in 2019, reaching a maximum of approximately 120. In contrast, the lowest average prices were observed in 2021, with rates dropping to around 50. Additionally, the period from 2015 to 2017 exhibited relatively stable price trends, with minimal fluctuations noted in the plots.

Figures 8 and *9* provide valuable insights into tomato price trends using a heat map and scatter plots. The heat map visualizes trends across specific parameters from 2013 to 2021, with months ranging from January to December. Color intensity varies from dark purple (indicating lower values around 20) to bright orange (indicating higher values above 70). Notably, higher price values are concentrated in specific months, such as May to July in 2016 and September to December in 2021, while darker regions in 2013 and 2018 indicate consistently lower prices.

In contrast, the scatter plot displays minimum tomato prices by month, where bubble sizes represent the corresponding maximum price range, varying from 20 to 120 units. Minimum prices tend to cluster between 20 and 60 units, with broader ranges appearing particularly in June and December, reflecting greater price volatility. The analysis suggests that the patterns observed in the heat map may be influenced by seasonal variations or annual market conditions, as also supported by the scatter plot. Periods of high activity in both plots point to potential seasonal peaks in demand or fluctuations in supply affecting tomato pricing.

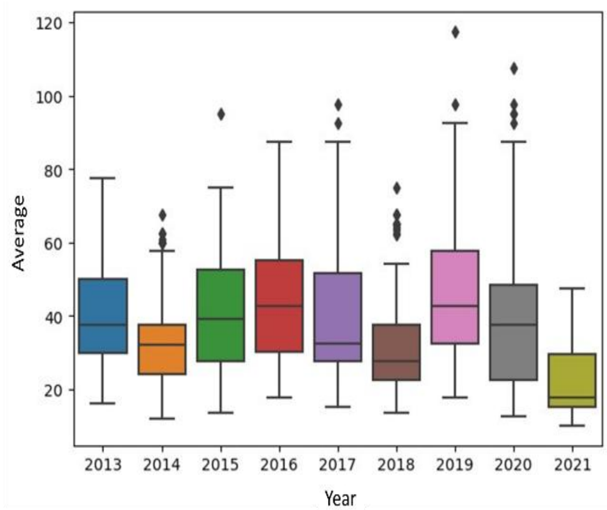

Figure 6 Box plot for average price of tomato each year

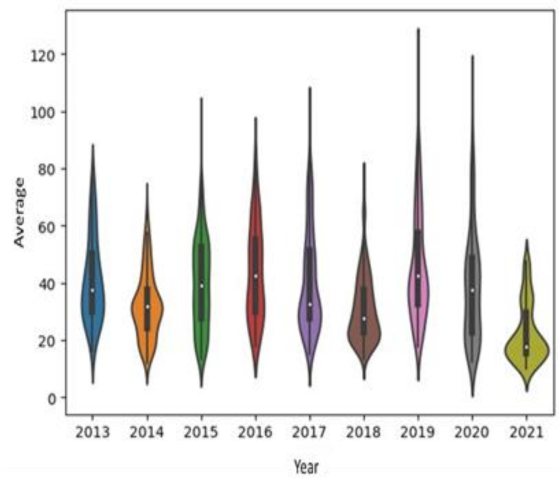

Figure 7 Violin plot for average price of tomato each year

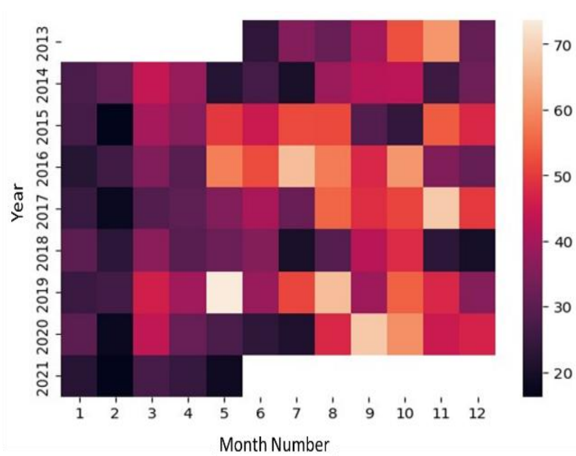

Figure 8 Heat map of the proposed approach

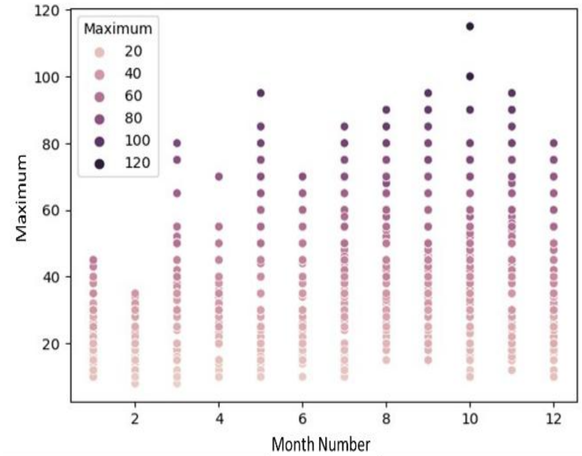

Figure 9 Scatter plot for minimum price of tomato each month

Figures 10 and *11* illustrate tomato price trends using a cluster map and a month-wise analysis of minimum prices. The cluster map groups similar price fluctuations across different years and time periods, highlighting patterns in pricing behavior. Additionally, the minimum price ranges are analyzed on a monthly basis. The graphs clearly indicate that tomato prices exhibit continuous fluctuations, with noticeably higher minimum price values observed in the 8^{th} (August) and 10^{th} (October) months. A significant drop in minimum prices is evident in the 12^{th} month (December), reflecting a drastic decline during that period.

Figures 12 and *13* illustrate the box plot and the maximum price ranges of tomatoes, respectively. *Figure 12* shows that tomato prices exhibit continuous fluctuations, highlighting the minimum price range along with their corresponding probability distributions. *Figure 13* presents the maximum monthly tomato prices, revealing that the price typically starts around ₹40 and peaks at ₹55. The highest maximum price ranges are observed between the 8^{th} and 10^{th} months. In contrast, during the 12^{th} month (December), the maximum price is relatively lower, ranging only between ₹40 and ₹45.

4.4Results based on error rates

The models are evaluated based on their error rates, and the results are presented accordingly. *Table 1* compares the performance of four predictive models LSTM, Bi-LSTM, GRU, and the proposed model using three evaluation metrics: MAE, MSE, and RMSE. The proposed model demonstrates superior performance, achieving the lowest error values: MAE of 0.546, MSE of 0.00463, and RMSE of 0.06807, indicating highly accurate predictions with minimal error. In contrast, the LSTM model yields the highest MAE (0.0953) and RMSE (0.1293), suggesting lower predictive accuracy. While the Bi-LSTM and GRU models show moderate improvements over LSTM, they still fall short of the proposed model's performance. These results highlight the effectiveness of the proposed model in capturing complex data patterns more accurately than traditional architectures.

Table 2 presents the performance of four predictive models LSTM, Bi-LSTM, GRU, and the proposed model using two evaluation metrics: MAPE and R^2. Among the models, the LSTM achieves the best performance, with the lowest MAPE of 0.2272 and a high R^2 value of 0.9632, indicating strong predictive accuracy and a good fit to the data. In contrast, the proposed model reports a higher MAPE of 0.2753 and a lower R^2 value of 0.83857, suggesting reduced effectiveness in capturing the underlying data patterns. Notably, one model shows an R^2 value exceeding 1, which is atypical and may indicate over fitting or potential data anomalies. Overall, based on these metrics, the LSTM model emerges as the most reliable and accurate predictor.

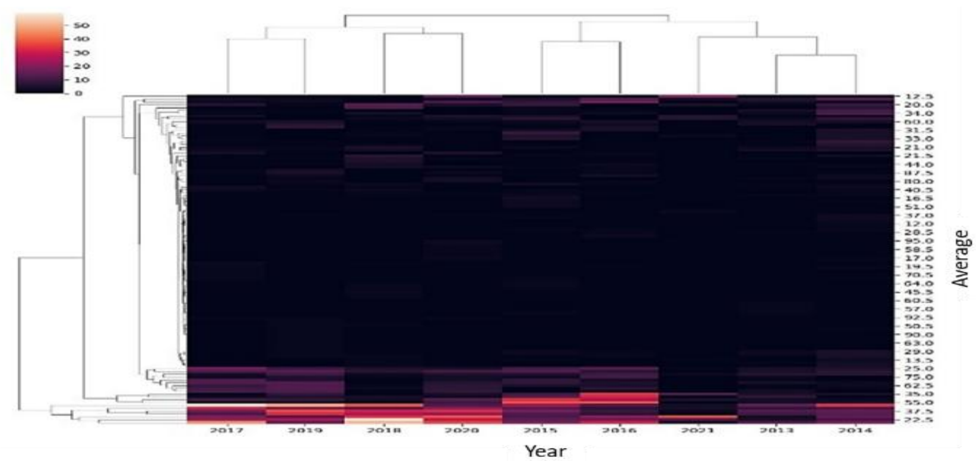

Figure 10 Cluster map of the projected approach

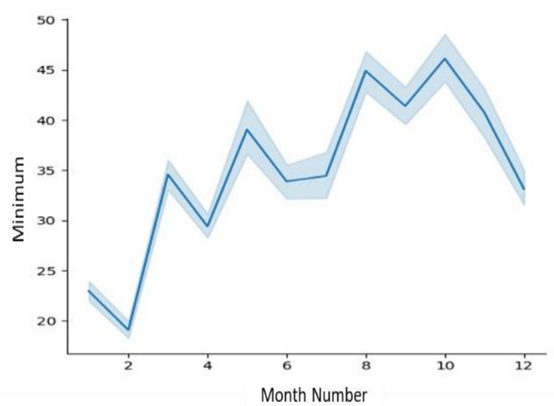

Figure 11 Minimum price graph of tomato on month-wise estimation

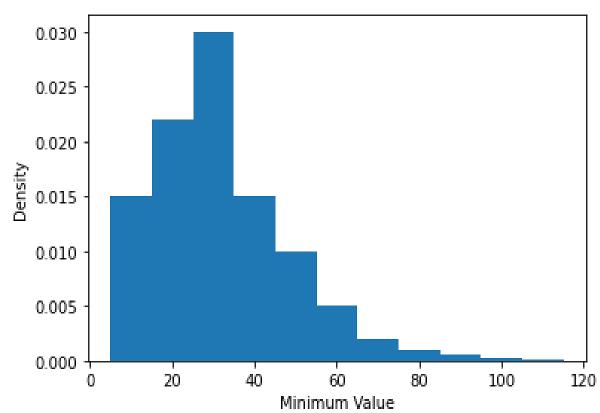

Figure 12 Relational plot function of the projected approach with respect to tomato price

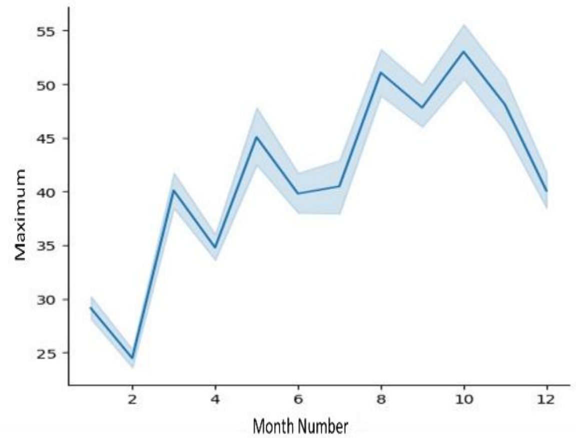

Figure 13 Maximum price plot of tomato in month wise

Table 1 Performance comparison of predictive models based on MAE, MSE, and RMSE

Model	MAE	MSE	RMSE
LSTM	0.0943	0.0796	0.1293
Bi-LSTM	0.0742	0.0756	0.1783
GRU	0.0919	0.0814	0.1349
Proposed StackNet-RC layer model	0.0546	0.00463	0.06807

Table 2 Performance comparison of predictive models based on MAPE and R²

Model	MAPE	R-Square
LSTM	0.2272	0.9632
Bi-LSTM	0.2744	1.0632
GRU	0.2643	0.9342
Proposed StackNet-RC layer Model	0.2753	0.83857

Limitations

The primary challenge associated with this model is its complexity, which can complicate both the training process and the interpretation of results. Moreover, employing such a sophisticated architecture increases the risk of overfitting, where the model may learn noise instead of meaningful patterns during training. Additional implementation challenges include the need for extensive data collection and preprocessing, the requirement for improved interpretability, access to advanced computational resources, and the necessity for continuous monitoring. A complete list of abbreviations is listed in *Appendix I.*

5.Conclusion

Agricultural products and their prices play a vital role in the functioning of the agricultural market. Vegetables, in particular, are subject to significant supply fluctuations and associated price volatility. These fluctuations are often influenced by meteorological changes and other dynamic factors, making price stabilization a major challenge. As a result, such volatility can have a substantial impact on the national economy. Although the government implements various measures to stabilize the supply of vegetables, recurring disruptions caused by weather, supply chain inefficiencies, and market dynamics remain critical issues. To address these challenges, studying market behavior and accurately forecasting vegetable prices are essential. The proposed study introduces an effective DL-based approach that leverages the strengths of CNN and LSTM networks, enhanced through the integration of a GS-CNN and a layered StackNet-RC architecture. It combines the GS-CNN with the StackNet-RC layer to generate accurate predictions for tomato prices. The StackNet architecture allows the stacking of multiple LSTM layers, each capable of learning sequential patterns in the data and passing refined outputs through the network. Meanwhile, the GS-CNN improves feature extraction through convolutional operations, enhancing the model's ability to learn complex patterns efficiently. By integrating these techniques, the proposed model effectively captures both short-term and long-term trends while optimizing kernel sizes and input propagation across layers. This results in highly accurate predictions with minimal error. The performance metrics confirm the model's effectiveness, with MSE values as low as approximately 0.00, RMSE at 0.06, MAE at 0.05, and an R^2 score of 0.83. These results affirm the model's capability and adaptability in forecasting tomato prices with high accuracy. However, forecasting agricultural prices—especially for commodities like tomatoes—requires a deep understanding of complex interactions among various factors such as production volumes, supply chain dynamics, and consumer behavior. Incomplete knowledge of these variables can limit prediction accuracy. Despite the strong performance of the proposed model, it has certain limitations. One of the main challenges lies in the model's complexity, which can make it difficult to train and interpret. Additionally, the use of a highly complex architecture increases the risk of over fitting, where the model may learn noise instead of meaningful patterns during training. Challenges in adopting this model also include data collection and preprocessing requirements, the need for improved interpretability, access to high-performance computational resources, and continuous monitoring. Future work can address these limitations by incorporating additional factors affecting commodity prices and utilizing larger, more diverse datasets to further enhance model performance. Although the proposed model has achieved low error rates, its effectiveness can be further improved in large-scale applications involving broader datasets.

Acknowledgment
None.

Conflicts of interest
The authors have no conflicts of interest to declare.

Data availability
The dataset used for this research is publicly available and was sourced from the following link: https://www.kaggle.com/datasets/ramkrijal/tomato-daily-prices.

Author contribution statement
G Sumaiya Farzana: Conceptualization, methodology, data curation, formal analysis, investigation, writing original draft. **N Prakash:** Conceptualization, methodology, data curation, investigation, validation, supervision and reviewing the draft

References
[1] Haybron DM. Well-being policy: consensus hallmarks and cultural variation. In beyond classical liberalism 2024 (pp. 205-21). Routledge India.
[2] Deepalakshmi R, Devi SP, Revathy JS, Shalini TG. Scheming a new algorithm for dynamic price prediction of vegetable commodities using statistical price prediction technique. International Journal of Computational Complexity and Intelligent Algorithms. 2019; 1(2):117-28.
[3] RL M, Mishra AK. Forecasting spot prices of agricultural commodities in India: application of deep-learning models. Intelligent Systems in Accounting, Finance and Management. 2021; 28(1):72-83.
[4] Pujari VM. A machine learning approach for tomato crop yield and price prediction. Journal of Advanced Zoology. 2023; 44: 517-23.
[5] Yin H, Jin D, Gu YH, Park CJ, Han SK, Yoo SJ. STL-ATTLSTM: vegetable price forecasting using STL and attention mechanism-based LSTM. Agriculture. 2020; 10(12):1-17.
[6] Ge J, Zhao L, Yu Z, Liu H, Zhang L, Gong X. Prediction of greenhouse tomato crop evapotranspiration using XGBoost machine learning model. Plants. 2022; 11(15):1-17.
[7] Gulati A, Juneja R. Transforming Indian agriculture. Indian Agriculture Towards. 2022; 2030:9-37.
[8] Prajwal CY, Chandan KS, Likith S, Poorna PMS, Santhosh B. Crop price prediction using deep learning. In ICT infrastructure and computing: proceedings of ICT4SD 2022 (pp. 297-302). Springer Nature Singapore.
[9] Dong S, Mo K. Analysis of factors for vegetable price fluctuation in Lu'an. In exploring the financial landscape in the digital age 2024 (pp. 456-62). CRC Press.
[10] Murugesan R, Mishra E, Krishnan AH. Forecasting agricultural commodities prices using deep learning-based models: basic LSTM, Bi-LSTM, stacked LSTM, CNN LSTM, and convolutional LSTM. International Journal of Sustainable Agricultural Management and Informatics. 2022; 8(3):242-77.
[11] Yuan CZ, Ling SK. Long short-term memory model based agriculture commodity price prediction application. In proceedings of the 2nd international conference on information technology and computer communications 2020 (pp. 43-9). ACM.
[12] Banerjee T, Sinha S, Choudhury P. Long term and short term forecasting of horticultural produce based on the LSTM network model. Applied Intelligence. 2022:1-31.
[13] Yue Y, Li JH, Fan LF, Zhang LL, Zhao PF, Zhou Q, et al. Prediction of maize growth stages based on deep learning. Computers and Electronics in Agriculture. 2020; 172:105351.
[14] Chaudhary M, Gastli MS, Nassar L, Karray F. Deep learning approaches for forecasting strawberry yields and prices using satellite images and station-based soil parameters. Proceedings of the AAAI spring symposium on combining machine learning and knowledge engineering (AAAI-MAKE) 2021 (pp. 1-12). CEUR.
[15] Okwuchi I. Machine learning based models for fresh produce yield and price forecasting for strawberry fruit. Master's Thesis, University of Waterloo. 2020.
[16] Murugesan R, Mishra E, Krishnan AH. Forecasting agricultural commodities prices using deep learning-based models: basic LSTM, bi-LSTM, stacked LSTM, CNN LSTM, and convolutional LSTM. International Journal of Sustainable Agricultural Management and Informatics. 2022; 8(3):242-77.
[17] Kankar M, Kumar MA. Price prediction of agricultural products using deep learning. In advanced machine intelligence and signal processing 2022 (pp. 505-18). Springer Nature Singapore.
[18] Mahto AK, Alam MA, Biswas R, Ahmed J, Alam SI. Short-term forecasting of agriculture commodities in context of Indian market for sustainable agriculture by using the artificial neural network. Journal of Food Quality. 2021; 2021(1):1-13.
[19] Abewickrama G. Analyzing and predicting the market price of fruits and vegetables. BSc. Dissertation, Informatics Institute of Technology, Delhi, India. 2022.
[20] Banerjee S, Mondal AC. An ingenious method for estimating future crop prices that emphasises machine learning and deep learning models. International

Journal of Information Technology. 2023; 15(8):4291-313.
[21] Kaewchada S, Ruang-on S, Kuhapong U, Songsri-in K. Random forest model for forecasting vegetable prices: a case study in Nakhon Si Thammarat Province, Thailand. International Journal of Electrical and Computer Engineering. 2023; 13(5):5265-72.
[22] Priyadarshi R, Panigrahi A, Routroy S, Garg GK. Demand forecasting at retail stage for selected vegetables: a performance analysis. Journal of Modelling in Management. 2019; 14(4):1042-63.
[23] Weng Y, Wang X, Hua J, Wang H, Kang M, Wang FY. Forecasting horticultural products price using ARIMA model and neural network based on a large-scale data set collected by web crawler. IEEE Transactions on Computational Social Systems. 2019; 6(3):547-53.
[24] Li B, Ding J, Yin Z, Li K, Zhao X, Zhang L. Optimized neural network combined model based on the induced ordered weighted averaging operator for vegetable price forecasting. Expert Systems with Applications. 2021; 168:114232.
[25] Lu H, Ma X, Huang K, Azimi M. Carbon trading volume and price forecasting in China using multiple machine learning models. Journal of Cleaner Production. 2020; 249:119386.
[26] Yang W, Wang J, Niu T, Du P. A novel system for multi-step electricity price forecasting for electricity market management. Applied Soft Computing. 2020; 88:106029.
[27] Bhavani M, Mounika P. A novel model selection framework for forecasting agricultural commodity prices using time series features and forecast horizons. International Journal of Scientific Research in Science and Technology. 2022; 9(5):134-44.
[28] Babu KS, Mallikharjuna RK. Onion price prediction using machine learning approaches. In proceedings of international conference on computational intelligence and data engineering: ICCIDE 2021 (pp. 175-89). Springer Nature Singapore.
[29] Areef M, Rajeswari S, Vani N, Naidu GM. Price behaviour and forecasting of onion prices in Kurnool market, Andhra Pradesh State. Economic Affairs. 2020; 65(1):43-50.
[30] Areef M, Rajeswari S, Vani N, Naidu GM. Forecasting of onion prices in Bangalore market: an application of time series models. Indian Journal of Agricultural Economics. 2020; 75(2):217-27.
[31] Baser P, Saini JR, Kotecha K. Tomconv: an improved CNN model for diagnosis of diseases in tomato plant leaves. Procedia Computer Science. 2023; 218:1825-33.
[32] Badal PS, Goyal A, Kamalvanshi V, Kumar P, Mondal B. Tomato price forecasting-a comparison between ARIMA, GARCH and ANN. Agro-Economist. 2023; 10(1):89-96.
[33] Reddy GT, Murthy BR, Umar SN, Prakash KK. Forecasting models for arrivals and prices of tomato at selected markets in Kurnool district of Andhra Pradesh. International Journal of Agricultural and Statistical Sciences. 2022; 18(1):1695-701.
[34] Khanum R. Forecasting of tomato prices using long short term memory. Indian Journal of Agricultural Marketing. 2023; 37(3spl):179-80.
[35] Kasimatis CN, Psomakelis E, Katsenios N, Katsenios G, Papatheodorou M, Vlachakis D, et al. Implementation of a decision support system for prediction of the total soluble solids of industrial tomato using machine learning models. Computers and Electronics in Agriculture. 2022; 193:106688.
[36] Belouz K, Nourani A, Zereg S, Bencheikh A. Prediction of greenhouse tomato yield using artificial neural networks combined with sensitivity analysis. Scientia Horticulturae. 2022; 293:110666.
[37] Chen W, Zhang H, Mehlawat MK, Jia L. Mean–variance portfolio optimization using machine learning-based stock price prediction. Applied Soft Computing. 2021; 100:106943.
[38] Jain A, Marvaniya S, Godbole S, Munigala V. Towards context-based model selection for improved crop price forecasting. In proceedings of the 5th joint international conference on data science & management of data (9th ACM IKDD CODS and 27th COMAD) 2022 (pp. 195-203). ACM.
[39] Tao KF, Min AN, Wei SU, Fang LJ, Lin HU, Shan CS. A dataset of the statistics on tomato transaction price in the wholesale market in Beijing-Tianjin-Hebei region from 2012 to 2019. Journal of Agricultural Big Data. 2023; 5(2):62-7.
[40] Haikal R, Wahyudi M, Pujiastuti L, Solikhun S. Fletcher-reeves algorithm for predicting the quantity of production tomato plants in Indonesia. International Journal of Mechanical Computational and Manufacturing Research. 2022; 11(3):109-20.
[41] Murugesan G, Radha B. An extrapolative model for price prediction of crops using hybrid ensemble learning techniques. International Journal of Advanced Technology and Engineering Exploration. 2023; 10(98):1-20.
[42] Gastli MS, Nassar L, Karray F. Deep learning models for strawberry yield and price forecasting using satellite images. In international conference on systems, man, and cybernetics (SMC) 2021 (pp. 1790-6). IEEE.
[43] Paul RK, Yeasin M, Kumar P, Kumar P, Balasubramanian M, Roy HS, et al. Machine learning techniques for forecasting agricultural prices: a case of brinjal in Odisha, India. Plos One. 2022; 17(7):1-17.
[44] Sabu KM, Kumar TM. Predictive analytics in agriculture: forecasting prices of arecanuts in Kerala. Procedia Computer Science. 2020; 171:699-708.
[45] Nayak GH, Alam MW, Avinash G, Singh KN, Ray M, Kumar RR. N-BEATS deep learning architecture for agricultural commodity price forecasting. Potato Research. 2024:1-21.

G Sumaiya Farzana received her Bachelor of Technology in Information Technology from Bhajarang Engineering College, affiliated with Anna University, Chennai. She completed her Master's degree in Computer Science and Engineering at B. S. Abdur Rahman Crescent Institute of Science and Technology, where she is currently pursuing her Ph.D. Her research interests include Machine Learning, Deep Learning, Data Analytics, and Cloud Computing.
Email: sumaiyaf310@gmail.com

Dr. N Prakash is currently serving as a Professor in the Department of Information Technology at B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. He holds a B.E. in Computer Science and Engineering from Bharathidasan University, Trichy, and an M.E. in Software Engineering from the College of Engineering, Guindy, Anna University, Chennai. He earned his Ph.D. from the School of Information and Communication Engineering at Anna University, Chennai. Dr. Prakash has published over 20 articles in refereed journals, authored two books, contributed to five book chapters, and presented in more than 20 conferences. His research interests include Data Science, Machine Learning, and Software Engineering.
Email: prakash@creat.edu.in

Appendix I

S. No.	Abbreviation	Description
1	AI	Artificial Intelligence
2	AF	Activation Function
3	AP	Average Pooling
4	ARIMA	Autoregressive Integrated Moving Average
5	BiLSTM	Bidirectional Long Short-Term Memory
6	CL	Convolutional Layer
7	CNN	Convolutional Neural Network
8	DL	Deep Learning
9	DSS	Decision Support System
10	FL	Flatten Layer
11	FC	Fully Connected
12	GRU	Gated Recurrent Unit
13	GRNN	Generalized Neural Network
14	GS-CNN	Global Scaling Convolutional Neural Network
15	LSTM	Long Short-Term Memory
16	MAE	Mean Absolute Error
17	MAPE	Mean Absolute Percentage Error
18	ML	Machine Learning
19	MSE	Mean Squared Error
20	NN	Neural Network
21	OWA	Ordered Weighted Averaging
22	PL	Pooling Layer
23	RC	Rapid Curing
24	RMSE	Root Mean Squared Error
25	SARIMA	Seasonal Autoregressive Integrated Moving Average
26	SEP	Standard Error Prediction
27	SVM	Support Vector Machine
28	SVR	Support Vector Regression
29	VAR	Vector Autoregressive
30	VARMA	Vector Autoregressive Integrated Moving Average