10.48047/jocaaa.2024.33.07.52

Forecasting Agricultural Commodity Prices Using Multilayer Perceptron Neural Networks

Rubi Kambo[1,2], Jyothi Pillai[3], and Sunita Soni[1]

[1] Department of Computer Science and Engineering,
Bhilai Institute of Technology Durg, Chhattisgarh, India
[2] Department of Computer Science and Engineering,
Rungta College of Engineering and Technology, Bhilai, Chhattisgarh, India
[3] Department of Computer Applications,
Bhilai Institute of Technology, Durg, Chhattisgarh, India
rubi.kambo7@gmail.com
jyothi.pillai@bitdurg.ac.in
sunita.soni@bitdurg.ac.in

Abstract. Transitions in the supply and demand for a good cause price changes, which in turn impact customers' disposable incomes. If a precise or somewhat accurate forecast can be made of the price of the commodity, then the risk associated with price fluctuations may be managed. In this research, a model is employed using MLP to accurately anticipate agricultural commodity prices. The model is trained using univariate time series price data. Daily tomato prices in the Raipur area of Chhattisgarh were used to test and refine the model. A computation and evaluation of the Root Mean Square Error (RMSE) of the model are carried out. As an analysis, it gives an innovative approach to computing the cost of agricultural products. The findings of the modeling process illuminate the degree to which the proposed model accurately predicts outcomes as well as the efficiency of the early warning system.

Keywords: Agriculture Commodity, Tomato, MLP, Price Forecasting

1 Introduction

The agricultural and ancillary industries in India are helping to boost the country's GDP. A positive growth rate of 3.6% was expected in agriculture and associated sectors in 2020–21. This was made possible by a combination of factors, including a good monsoon and a number of government efforts aimed at increasing financing availability, improving investments, building market facilities, promoting the growth of infrastructure in the agricultural sector and boosting the availability of high-quality inputs to the industry.

Timely execution of the Atma Nirbhar Bharat (ANB) Abhiyan and other growth-promoting initiatives (ANB and other programmes are described in the appropriate areas) has further assisted agriculture in reaching an enhanced growth rate of 3.9% in 2021–22. Agriculture is one sector that has grown rapidly

10.48047/jocaaa.2024.33.07.52

during the last two years. With a 3.6% increase in 2020–21 and a 3.9% increase in 2021–22, the sector that provides jobs for the vast majority of the population significantly boosted the country's Gross Value Added (GVA). The primary drivers of industry expansion have been growth in allied businesses such as dairying, fishing, and agriculture [1].

The price of agricultural commodities has an impact on the economy of our nation. The commodity price is determined by the demand for the crop. There are a number of variables that could affect the price of a crop, including supply and demand, the weather, and the time of year. Knowing the expected market price for a crop well in advance of its commercial release would be of great benefit to both consumers and producers alike. The use of deep neural networks to forecast crop prices is a relatively new method that might lead to the creation of a useful tool for long-term commodity price forecasting and monitoring.

In order to extract the underlying truth or pattern from the current commodity market price, innovative methods are being utilized in the agriculture industry today. To estimate future commodity or agricultural prices, analysts use historical data to examine patterns in the past. Time series data may be useful for examining how the cost of a crop or product has changed over time. Time series data may be used to estimate future crop prices based on observations made at many intervals (daily, weekly, etc) [2]. Researchers uncovered a plethora of papers using neural networks and deep learning strategies for agricultural commodity price prediction. Researchers utilising the multilayer feed forward network and comparing it to ARIMA for US Live Cattle data have been the subject of many papers on price prediction methods [3].

The other contributors also employed the genetic approach in the network for learning and provided ambiguous inputs and weights. Not only for the commodities sector but also for the boiler industry, they also applied the case-based reasoning technique, and discovered this strategy performs better than linear regression or the regression tree approach [4].

Integrating ARIMA and NN in various ways, such as backpropagation with equal weight. For assessment, they computed MSE, MAPE, and MAE and indicated that ANN shows superior benefit [5,6]. In predicting the price of sugar, a few researchers employed a hybrid model such as ANN with the Kalman filter. This data was acquired from the Brazilian and Indian markets [7] not only for vegetable commodities but also for predicting the prices of oil seeds like soybean and rapeseed mustard, others in the literature tested out the ANN-based model using data from these crops. They focused on the turning point as a component that is helpful in making predictions.[9]. Develop a short-term prediction system for five vegetables and use data from the Beijing market. They employed the exponential smoothing and moving average approaches, and certain volatile factors were assessed via a Multivariate Regressor MR. Using both linear and nonlinear techniques, such as VECM and SVR, is a novel approach for predicting future prices of cotton and maize in China [10]. This method was intended to help in price forecasting. Machine learning approaches like radial basis, backpropagation, and genetic-based networks are also used for forecasting the price

10.48047/jocaaa.2024.33.07.52

of vegetables for the next day. However, genetic-based networks show a higher level of accuracy than the other approaches [11]. The researchers used the data on agricultural products that were sold on the Brazilian market and then applied the ANN model to their multivariate analysis. It was discovered that a large number of other academics were working on the price prediction of the agriculture commodity [12].

ELMs perform better than other models in terms of error metrics, based on computational research. A unique technique that includes preprocessing steps, lag choices, and the application of ELMs, the analysis addresses a gap in the area of coffee price prediction [13].

Price asymmetry exists between real interest rates and agricultural commodities. The negative correlations between real interest rates and agricultural commodity prices have been found statistically significant [14]. The dimensions such as user, item, and time capture temporal patterns and input these dimensions in CNN-based model for predicting user preference over a particular context on these dimensions. And suggested that user, item, and time correlation must be analyzed for prediction [15]. Two dimensions such as user and item for the prediction suggested two-stage deep-learning-based recommendation helps in prediction [16]. As Time series data are nonlinear in nature so fuzzy-based RBNF and Auto-regressive model with exogenous variables results in effective accuracy and performance than a simple model [17]. Price fluctuation affects the demand and supply of any commodity so a dual input-based mechanism in a short-term memory-based model was suggested by the author and they recommend using data such as meteorological, past price, and trading volume and applying the model for predicting the price of two vegetables such as cabbage and radish for south Korean market [18]. LSTM networks predict carry trade profits better than regression models.

Factors such as demand, supply, population growth, consumer preference, production, and transportation cost affect the price so all these factors must be analyzed during the commodity price forecasting. Other determinants include natural disasters, public opinion, exchange rates, and urban-rural consumption indices and also suggested models based on RBF for price forecasting [19]. The Internet of Things (IoT) boosts agricultural efficacy, and a cutting-edge MLP method improved precision and productivity by 99% when predicting sugar production in IoT agriculture [20].

Predicting the price of commodities, notably oil, is a popular area of research. Multiple researchers use a variety of techniques, for example, Gaussian process estimation and vine copula regression, to predict oil rates employing the benchmark S&P 500 inventory information. The combination of Vine copula regression and NLPCA is the most effective approach for reliably forecasting oil prices [21].

An MLP neural network was found to have lower MSE and be useful for decision-making regarding future banana prices, despite the authors' preference for an ARIMA (4,1,2) (1,0,1) model. Understanding the previous pricing is critical for understanding and estimating future changes in banana prices, suggesting that research efforts can improve future price estimates [22].

10.48047/jocaaa.2024.33.07.52

The study finds that MLP and NARX models beat ARIMA in Korean power generation SMP that is system marginal price prediction, with certain hidden layer units performing best. All predictions rank ARIMA, MLP, and NARX by forecast error [23]. Suggested MLP model to predict the dropout rate of students during the COVID-19 pandemic [24]. MLP-based forecasting is done not only in the field of agriculture but also in power load forecasting taking into account atmospheric elements for accuracy. The suggested approach forecasts electricity load better than existing models, proving its reliability and effectiveness [25].

Theoretical research identifies that low Forecasting error helps in reducing the cost of operation and risk suggested a two-stage load forecasting model, combining LSTM and MLP neural networks, which outperforms other models in accuracy. Sequence to sequence module is suggested effective in the early hours of predictions, and the MLP further enhances results, reducing the MAPE to 2% [26]. Determine neurons that remain hidden in neural networks and a new strategy to Elman networks for wind speed prediction, offering a simple, effective response [27].

Food-energy price relationships are analyzed using the vector autoregressive-moving-average algorithms with coherent time intervals.

Partial Wavelet Coherence (PWC) and Multiple Wavelet Coherence (MWC) to identify coherency across time and frequency domains for five agricultural commodities namely global crude oil, soybeans, corn, wheat, and and prices for sugar instead of return and volatility correlations in model-free wavelet analysis [38]. Sentiments related to commodities along with factors such as economic, social, and environmental were addressed [33] and their impact on price forecasting is discussed in [37].

The MLP is a kind of ANN that is often used for Machine Learning applications such as price prediction. Other examples include the Synthetic Neural Network (SNN). The use of an MLP model for predicting vegetable prices may have a wide variety of outcomes.

Improved accuracy : They are able to catch intricate patterns and correlations in the data, MLP models are particularly effective at forecasting the price of vegetables. In order to provide more accurate predictions about vegetable prices, an MLP model may be trained on historical data to understand trends, seasonality, and other characteristics that impact prices.

Rapid Decision Making : Prudent choices may be made in the vegetable industry with the support of precise price estimates by farmers, wholesalers, and retailers. With real-time information on prospective price changes, they may adjust production, inventory management, buying, and pricing strategies as needed.

Contingency Planning : A number of market participants might benefit from a vegetable price prediction in terms of risk management. Farmers may decide

10.48047/jocaaa.2024.33.07.52

when the optimum time is to harvest and sell their crops, mitigating risk from price fluctuations. Wholesalers and retailers may structure their pricing and buying methods to mitigate the consequences of price volatility.

Risk Management : With reliable pricing projections, the vegetable supply chain may be optimized. By anticipating price changes, stakeholders may better plan distribution channels, warehouse space, and shipping operations. This optimization has the potential to lessen costs, cut down on waste, and ensure customers have consistent access to fresh veggies.

Market Disclosability : Models that anticipate future prices of vegetables may contribute to the openness of markets by providing more information about fluctuations in prices. The ability of market participants to make well-informed choices based on market circumstances, supply-demand dynamics, and other factors is enhanced when they have access to reliable price estimates.

Strategy Formulation : Models that estimate the future value of vegetables may help governments and policymakers examine market trends, measure the impact of existing restrictions, and develop effective responses. By analysing price shifts and the factors that underlie them, policymakers have the ability to institute measures that will bring about price stability, provide assistance to farmers, and ensure the safety of the food supply.

It's worth noting, however, that no model is foolproof and that variables like weather, pests, illnesses, geopolitical events, and consumer preferences make it hard to reliably estimate vegetable prices. Furthermore, the performance of MLP models is heavily influenced by the quality and accessibility of the data. As a result, while MLP-based price prediction models may have beneficial effects, they should be employed as only one tool among many in the vegetable industry's decision-making processes. In this article, an MLP-based exploratory study on agriculture commodity prices was performed on univariate price data, and experiments were carried out. This paper is divided into several parts: Part II discusses the related work, where different linear and nonlinear models were found to be used for forecasting in different application areas like electricity price and load forecasting; Part III includes the data preparation or data set used for the experiment and discusses the existing methodology or architecture used in an experiment; Part IV displays the experimental result performed on price data; and Part V concludes with some conclusions and future work. The results and experiments are summarised in Section V, and the future directions of the study are considered.

2 System Modelling

In this investigation, the experiment was carried out using Python. Python Scipy is a prerequisite for this purpose. In addition, TensorFlow is used as a backend

10.48047/jocaaa.2024.33.07.52

with Keras 2.0. To conduct the research, modules like Pandas, Numpy, and Matplotlib were used.

2.1 Predicting the cost of vegetables using a non-linear model

1. Agriculture product wholesale cost: The daily commodity prices for vegetables at the wholesale level are described in the vegetable price dataset. Experimentation is performed using data collected over one year. The government's daily report on prices and arrivals may be seen at https://www.nhb.gov.in /OnlineClient/MISDailyReport.aspx. In this experiment, the price of a commodity, such as a tomato, from a certain website to do forecasting-related research. New product variants, such as local and hybrid varieties, have entered the marketplace. The native kind of tomatoes that are grown in the Raipur area of Chhattisgarh state were used for this experiment, and the price of those tomatoes was taken into consideration. We only used the middle price of tomatoes that was published on the website, although data for minimum, maximum, and median prices, as well as retail prices, were all available. The utilization of data and calculations done on the data were used to forecast the price of veggies for the next day. Using the Pandas and Matplotlib libraries, a series of data are imported and plotted in Python. Below is a line plot of the data series, which was generated for one month of commodity prices as shown in Fig 2. The graph demonstrates rising price tendencies.
2. Proposed forecasting model based on MLP: The subsequent phase was to go through the model setup, and many test harnesses were carried out. Multiple tasks, including but not limited to the following, are required for the experiment.
   a) Data Split:
   The daily commodity price data set is partitioned into two subsets a training set and a testing set. As previously mentioned, a dataset spanning one year was gathered. The training set included seven months of data from a whole year, whereas the test set consisted of the remaining three months. To determine the pricing of tomato commodities, data were gathered beginning in January 2018 and continuing through December 2018. The first set, covering January through September, was used for training, while the second set, covering October through December, served as test data. Actual observed data is pulled from the website and input into the model for the experiment. The model is then trained using the data, and it predicts the value of the commodity one day in advance. Forex, actual data from the previous year or daily data of the past year utilized in the model, such as 31 December 2018, etc., were used to forecast the price in January 2019. The total number of days of historical data included in the model is hyperparameterized by the timestamp.
   b) Methods for Model Configuration and Assessment:
   Assembly of Models: The MLP-based model was used to conduct a rolling window analysis, often known as the walk-forward approach. In rolling win-

10.48047/jocaaa.2024.33.07.52

dow analysis, historical data is accumulated over time and used to make projections about the market's future worth. The process of forecasting serves to increase one's awareness of future trends. It's useful for spotting trouble spots in performance. For the duration of the investigation, the "roll" dimension in the rolling window remains constant. In the course of this research, a variety of timesteps, including 1, 3, 5, and 7, were used. This implies that in the test dataset, each step walked one, three, five, or seven steps at a time. In this research, an MLP-based model was used to create the prediction for each of the four anticipated timesteps. The test set's actual or observed values were used to predict the following day's value, taking into account the various time steps. Models are used in the agricultural industry to anticipate the cost of a certain product based on data on its expected daily arrival and current market price. On the test dataset, every conceivable prediction was computed, the results were gathered, and an error score was computed for the model; this score, in turn, summarises the model's performance. To measure the effectiveness of the model, root-mean-squared error (RMSE) was used. Preparing and transforming data for experimentation is necessary before fitting the model to time series data. The skeleton of a simple forecasting model is shown in Fig. 1.

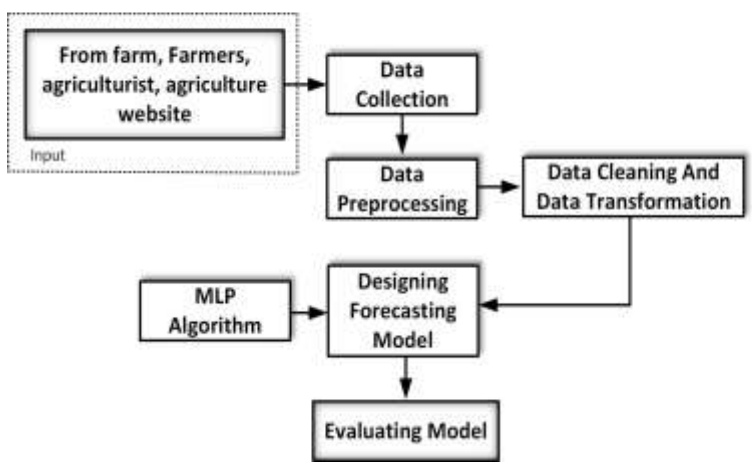


Fig. 1. Basic Model Architecture for Forecasting

3 Research Methodology

In this paper, an innovative MLP-based model for predicting the price of vegetables is suggested. In Section 3.1, the procedure for selecting data is presented. Section 3.2, provided a comprehensive model layout. Finally, Section 3.3 explains the experimental methodology and the measures used in the study's assessment.

10.48047/jocaaa.2024.33.07.52

3.1 Methods for Choosing and Preparing Data

To assess the accuracy of our proposed MLP methodology, the research looks at the daily modal prices of several commodities grown in the Raipur district of Chhattisgarh. These crops include tomatoes. Annual city statistics (covering around 230–240 calendar days) are gathered from the official government website. How often studies are conducted is determined by the timeliness of available data and the characteristics of the explanatory factors. The short-term lagging effects on the pricing of vegetables are collected monthly. The study mostly focused on predicting one-month using data from one year due to the nature of the algorithm. Therefore, samples ranging from January 1, 2018, all the way through December 31, 2018, are employed for the data splitting. The model is fitted using the first eight months of data (e.g., 01/01/2018 to 30/8/2018), and its prediction strength is then tested with the remaining samples (e.g., 01/09/2018 to 31/12/2018). In a nutshell, we have 240 samples set aside for training, and between 90 and 100 samples have been set out for testing. Take note that there are 235 data samples included in one year's worth of research.

1. The Wholesale Cost of Agricultural Products: The daily prices of commodities entering the wholesale market are described in the Vegetable Price Dataset. The experiment uses data collected over one year. The following URL is where the daily price and arrival data report is retrieved from the government website: https://www.nhb.gov.in/OnlineClient/MISDailyReport.

   The purpose of this experiment is to gather data for forecasting-related research by downloading the price of a commodity like a tomato from a certain website. Different kinds of products made their way into the market, such as the local and hybrid kinds of products. In this study, the cost of the locally grown tomato variety is used. Daily minimum, maximum, and median prices, as well as wholesale and retail prices, were all shown on the website, although in this experiment only considered the median price of a commodity. Vegetable price projections for the next day were calculated using historical data. The data is loaded in series format as shown in Fig 2.

10.48047/jocaaa.2024.33.07.52

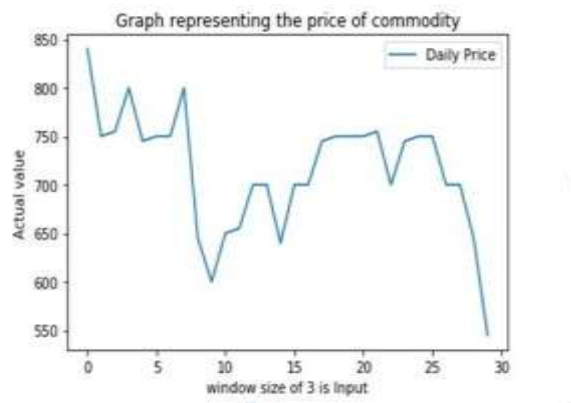


Fig. 2. Graph depicts the plot of day and price of a commodity for one month

3.2 MLP based Model Design

Step one in putting the MLP-based model into action is to gather and record all relevant trade and pricing data, either in a database or a file. Step three entails loading and pre-processing this data.

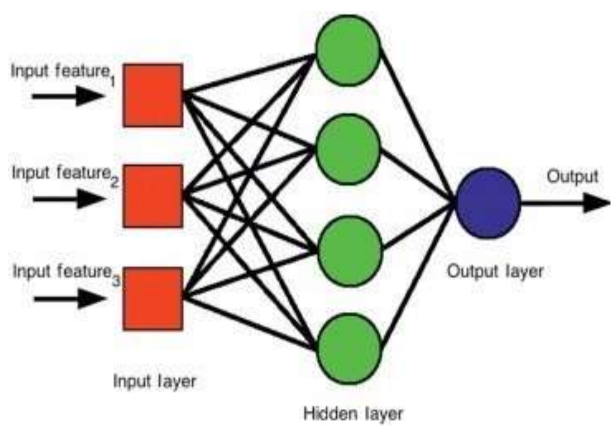


Fig. 3. MLP Model with Three Layer along with window size ws=3

Data Preprocessing The first stage is called ”data preprocessing,” and it entails performing transformations on the acquired data. When there is a data point that does not exist, the previous day’s value is used as a stand-in. During

10.48047/jocaaa.2024.33.07.52

this trial, three distinct transformations were carried out. To get rid of the rising trend in the data, the time series must be ”stationarized” by employing a differencing lag value of 1. Second, employed an Input-Output data structure to convert the time series data into a supervised learning problem for experimentation and then fed that data into a multi-layer perceptron model. The value that was gathered during the time step before this one was used as an input to predict the output during this time step. Normalization is the last step in transforming the data to a given scale from the range -1 to 1. To facilitate quicker learning and convergence, neural network-based models need normalized data. Therefore, use the min-max scalar for each variable across the entire training set to standardize them. The formula for scaling:

$$X_{sc} = \frac{X - X_{min}}{X_{min} - X_{max}} \quad (1)$$

Since just the quantity measures must be taken into account, such a rescaling does not alter the economic or internal meaning of feature variables. On the test samples, the same transformation is carried out in compliance. All of these transformations had to be reversed to the original scale before the error score could be calculated after the experiment.

Model Design MLP-based models were built for agriculture commodities. For training and testing, 80 After downloading price data for training and testing, this study makes use of the MLP model, whereby the model is defined by comprising numerous layers of input nodes that are directly coupled to one another. Between the input and output layers, a directed graph is built at the point of intersection. MLP is a deep learning method that uses backpropagation for training the network. A multilayer perceptron-based model can be applied at any time series data for forecasting. The MLP-based model used in this study predicts the price of the next day using only the previous day’s price data of agriculture commodities. Previous timestamps are used as input. A data vector, or window size, is used for predicting the price of the next day. The MLP model consists of multiple layers of interconnected artificial neurons. Each neuron receives input from the previous layer, applies a nonlinear activation function, and passes the output to the next layer. MLP’s hidden layers let it detect complex correlations in the data, while the output layer gives the final predicted values. As can be seen in Fig. 3., the MLP model has three distinct layers: an input layer, a hidden layer, and a prediction layer.

Layer-by-layer explanations and design considerations:

1. Input Layer: The first layer of the model is the input layer, sometimes called the feature layer, and it is responsible for receiving data, in this case prices. The number of neurons in the input layer determines the size of the input vector. Here, t is the timestamp, and $X = (X_1, X_2, X_3, \quad , X_t)$ Where t is the timestamp.

10.48047/jocaaa.2024.33.07.52

2. Hidden Layer: The second layer, ”hidden,” processes the incoming data. At this layer, which is known as the computational layer, the input is processed. This research begins with a single hidden layer since it is known to be sufficient for approximating the mapping of any continuous function. The selection of neurons in the hidden layer was conducted empirically according to a thumb rule. Overtraining occurs when the hidden layer has too many neurons, whereas a lack of neurons might have a negative impact on the model’s overall performance. After the learning process has been completed successfully, the number of neurons may decrease. In comparison to a newly trained model, the results of a network that has been trained several times are far more favorable.

   Activation Function : The Rectified Linear Unit (ReLU) function is a prominent activation function that is utilized in a variety of machine learning models, including those used for price forecasting. The primary purpose of the ReLU function is to introduce non-linearity into the model. This makes it possible for the model to discover intricate linkages and patterns in the data, which is essential for accurate price predictions.

   RELU helps predict price movements because:

   In a neural network, a neuron (or node) is said to be activated (i.e., fire) if the weighted sum of its inputs meets a threshold value set by the activation function. The activation function of the ReLU is defined as follows:

$$ReLU(x) = maximum(0, x) \tag{2}$$

   If the value being inputted is positive, the output will be that value; otherwise, it will be 0.

   When compared to other activation functions like sigmoid and tanh, the vanishing gradient issue is something that may slow down the training of deep networks; however, ReLU can help alleviate that.

   In the domain of price forecasting, the ReLU function is generally used as the activation function in neural network models like feedforward neural networks or deep learning designs like convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

   The model becomes non-linear as a result of the use of this straightforward thresholding. When compared to other activation functions, such as sigmoid or tanh, the primary benefit of ReLU is that it helps alleviate the vanishing gradient issue, which is a challenge that may arise during the training of deep neural networks.

   Non Linearity: Complexity and non-linearity are hallmarks of the price data and financial markets. These non-linear correlations may be more accurately captured by the neural network with the assistance of the ReLU function. The ability of the model to mimic complex patterns in the data is made possible by the ReLU algorithm, which prevents neuronal firing unless the input is positive.

   Sparse Activation: The ReLU function has several fascinating properties, one of which is that it brings sparsity into activations. This is known as sparse

10.48047/jocaaa.2024.33.07.52

activation. As a consequence of the fact that it is only active when the input is positive, certain neurons may stay inactive while the model is being trained. This gives the model the ability to concentrate on the characteristics that are most important to it while ignoring the noise.

Effective Computation: The ReLU function may be computed in an effective manner, which is essential when working with huge datasets and deep architecture.

The ReLU activation function is often used as the activation function in the hidden layers of neural network model architectures for use in price prediction. In addition to this, its derivative is either 0 or 1, which helps to simplify the process of backpropagation when it is being used in training.

3. Prediction Layer: The third and final layer of the network is the prediction layer, which makes forecasts about the commodity's price for the next day. The combined output of the input and hidden layers is used to generate the final result.

The use of a thumb rule enabled the selection of the hidden layer's number of neurons.

TR:1 The amount of neurons in the hidden layers should be determined by the size of the input and output layers.

TR2: The number of hidden neurons should be equal to two-thirds the size of the input layer and the output layer combined.

TR:3 The number of hidden neurons needs to be much less than twice the magnitude of the input layer.

Alternatively, one may use a heuristic technique like cross-validation to judge performance.

During the research, several models for MLP were considered, including the one that is shown in Fig.4., with 1IL:1HL:1OL proportions.

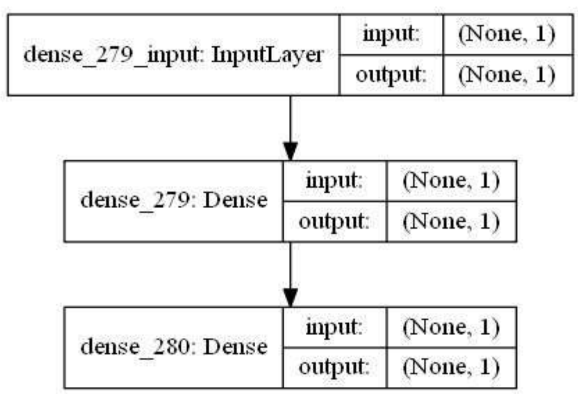


Fig. 4. MLP Model with ws=1

10.48047/jocaaa.2024.33.07.52

Model Training The MLP's output layer provides the final predicted values, while its hidden layers are responsible for enabling the model to identify intricate correlations in the data. The MLP model is constructed with the assistance of hyperparameters and an appropriate architecture. Optimization methods like backpropagation are used to iteratively alter the weights of the neurons in the model based on the data from the training dataset.

1. Backpropagation: The price disparity between what was predicted and what happened should be reduced as a goal. One of the most crucial algorithms for training ANNs is called backpropagation. To do this, it repeatedly adjusts the weights and biases of the neurons, allowing the network to "learn" from the input data. Backpropagation is a method that works by recursively propagating the error of the network's output across the layers to obtain the gradient of the loss function concerning each weight and bias. This is done to identify how the loss function is affected by each weight and bias. Backpropagation is a prevalent method for training artificial neural networks. The neurons in the network are given the ability to learn from labeled training data by adjusting the weights and biases of connections within the network. To describe how backpropagation works on the inside, consider its outlined phases:
   Backward Propagation:- The backward propagation of errors through the weights of the network and bias in the opposite direction is the most essential component of backpropagation. At first, the gradients of the loss function relative to the network's output need to be calculated. After the gradient is calculated, it is sent back through the network's layers to see whether any changes were made. Each neuron's slope is calculated using its activation function's derivative and the gradient of the neurons it connects to in the layers below it. In calculus, this technique is known as the rule of chaining.
   Forward Propagation:- When the neural network is accessed, the input data propagates through the layers in a forward fashion. To produce an output, each neuron in a layer must first receive signals from the layer above, and then process the weighted sum of those signals using an activation mechanism.
   Calculation of the Loss:- After defining a loss measure, the forward propagation process compares the network's output against the intended output. The loss function calculates the dissimilarity between the predicted and observed output.
   Adjustments made to the weight and bias:- An Update on Weight and Bias: Once the gradients of the weights and biases are known, the weights and biases are modified to minimize the loss. In most cases, this is achieved by the use of an optimization technique such as gradient descent, which makes bias and learning rate-scaled weight adjustments in the opposite direction of the slope.
   Until the network's efficiency stabilizes or reaches an adequate level, the aforementioned steps 1–5 are repeated a predetermined number of times over a defined amount of time using numerous different training samples. Using the slopes determined by backpropagation, the artificial neural network can

10.48047/jocaaa.2024.33.07.52

fine-tune its weights and biases to improve its predictive abilities and reduce the loss function. To get the lowest possible error in training data, weights are also modified throughout the learning phase. The layer-specific hyperparameter settings that were employed were... When predicting a single actual value, it is sufficient to have a single neuron in the last fully connected layer, also known as the prediction layer.

3.3 Pseudo Code

1. Input: Vegetable Price Data Set.
2. Output: Price Prediction for next day.
3. Data Collection.
4. Data Loading.
5. Data Preparation:
   (a) Transformation TS data into Stationary by differencing of order 1.
   (b) Transformation TS data into supervised form by setting lag=1 or window size=1,3,5 or 7.
   (c) Scaling data into Range from[-1,1] by min-max scalar.
6. Partitioning the Input price data into two parts i.e 70% for training and 30% for testing.
7. After preprocessing, apply the MLP-based TS model to the Price data of the commodity.
8. for training, Repeat for each training vector pair(x_p, t_p), evaluate output (y_p) when (x_p) is Input. if (y_p) $\neq$ (t_p) form a new weight vector according to the below-given formula: (W_{np}) = (W_p) + α ($t_p - y_p$).(x_p) weights are propagated back to the starting point of a network by the below-given equation

$$\delta w_p(t_p) = -e(\partial E/\partial w_p(t_p)) + \alpha\delta(w_p(t_p)) \quad (3)$$

   In each iteration, the weighted sum is forwarded through all layers and MSE is calculated. MSE is calculated as a loss by the formula,

$$MSE = \frac{1}{N} \Sigma_{i=1}^{N} \left(y_{pi} - \hat{y}_{pi} \right)^2 \quad (4)$$

   where N= no. of datapoints,
   Y_{pi}= observed values,
   Y_{pi} = predicted values.
   And Neurons are activated by RELU,
   if input > 0 :
   return input
   else
   return 0 function g(z) using max() over a set of z,0, and Input.

$$g(z) = max\{0, z\} \quad (5)$$

9. Test the model with 30% of the rest price data.

10.48047/jocaaa.2024.33.07.52

10. Forecast the next day price.
11. Invert the scaling.
12. Invert the Differencing
13. Store the forecast result into the predicted variable.
14. Lastly, performance measure on RMSE

$$RMSE = \sqrt{(forecast\ value - observed\ values)} \tag{6}$$

3.4 Training Procedure

Procedure for Training: While backpropagation has been used to train multilayer perceptron (MLP) models, the optimizer known as Adam has been more popular as an optimization approach. Blending ideas from RMSProp (Root Mean Square Propagation) and AdaGrad (Adaptive Gradient Approach), it is an optimization method with a dynamic rate of learning. The fundamental goal of the Adam optimizer is to dynamically adjust the MLP's learning rate for each parameter based on the magnitude of their respective prior slopes. Backpropagation uses a record of the squared values of prior gradients and an exponentially decreasing average of all those values to adjust the MLP's parameters. For each parameter, the approach primarily uses two estimates: the first-moment estimate (mean) and the second-moment estimate (uncentered variance). The algorithm's main estimates for each parameter are their first-moment estimates (mean) and second-moment estimates (uncentered variance). During the first stage of training, the estimated values are set to zero by initializing the vectors that make up those values to zero. Throughout each iteration of the backpropagation approach, the Adam optimization model produces these estimates by taking into account the slope of the parameter at the time as well as the rates at which the mean and uncentered variance degrade. The Adam optimizer is a well-known method of optimizing the training of multilayer perceptron (MLP) models via backpropagation. It combines ideas from the Adaptive Gradient Approach (AdaGrad) and the Root Mean Square Propagation (RMSProp) algorithms to optimize the learning rate in a dynamic environment. To adjust its settings, the Adam optimizer takes these estimates into account with its learning rate throughout its update phase. The following procedures are carried out: The MLP's parameter slope may be determined with backpropagation. Improve the current first-moment estimate (mean) by incorporating the decay rate of the mean with the current gradient. The second-moment estimate (uncentered variance) is updated by multiplying it by the decay rate and adding the squared value of the current gradient. Estimate the mean and uncentred variance using zero-centered initialization bias corrections. Modifying the parameters of the MLP is accomplished by performing the following calculation: multiplying the learning rate by the bias-corrected mean estimate, then dividing this result by the square root of the bias-corrected uncentered variance estimate. Due to adaptive learning rate adjustments and the incorporation of momentum-like behavior through the first-moment estimate, the Adam optimizer is both effective and efficient when training MLP models. It aids in the control of sparse gradients, boosts convergence, and provides stability

10.48047/jocaaa.2024.33.07.52

while choosing hyperparameters. An often-used optimization strategy in MLP backpropagation is the Adam optimizer, which controls the learning rate and improves training performance.

All of the above analyses were carried out on one month's worth of tomato pricing data from the Raipur area shown in Fig.2. Using the aforementioned rules of thumb, the study's authors estimated RMSE for use in model training. Different scenarios involving the MLP input layer neuron generation depending on the window size or rolling window are considered.

Case 1: The annual price of tomatoes in the Raipur area is taken into consideration. In this scenario, the batch size is 4, and the experiment is run 20 times with a value of 1 for the lag feature variable and a value of 1 for the hidden layer neurons. The activation function is set to RELU, the loss is the mean squared error, and ADAM is used to optimize the model. The epochs used here are 50, 100, and 500.The RMSE score was evaluated across different training epochs (50, 100, and 500) using a lag value of 1. The comparative performance of these configurations is shown in Fig. 5., which presents a summary of RMSE error scores.

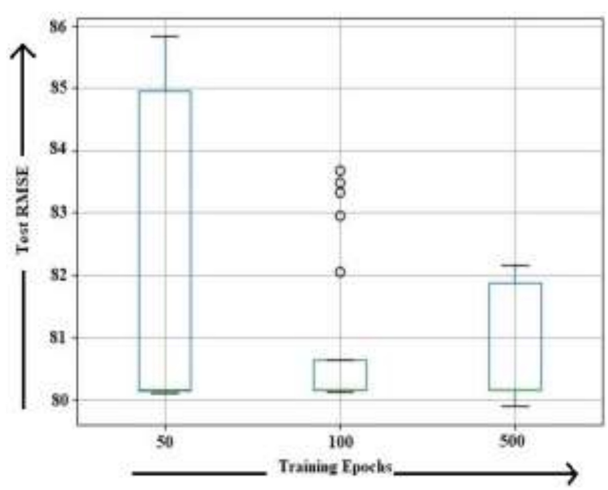


Fig. 5. RMSE Error Score Summary Plot with lag 1 considering epochs 50,100, 500

The RMSE plot shows that 100 epochs yield the lowest and most stable RMSE, indicating optimal model performance, as depicted in Fig. 5.

Case 2: Consider case 2, the Annual Raipur Tomato Price. Here, the batch size is 4, there are 20 iterations of the experiment, and both the value of the lag feature variable and the number of neurons in the hidden layer are set to 1. In addition, the activation function is set to Relu from the outset, the MSE is

10.48047/jocaaa.2024.33.07.52

utilized as the loss, and ADAM is used as the optimizer. epochs equals [50, 100, 500, 1000].

Case 3: Diagnostic MLP with Variable Epochs, Hidden layer neurons, and Window Sizes.

a) Iterate ten times for a total of one thousand epochs, using a batch size of four and a hidden layer neuron count of one, with a lag value of one for ws1, ws3,ws5, and ws7.

b) Iterate ten times for a total of one thousand epochs, using a batch size of four and a hidden layer neuron count of two, with a lag value of one for ws3,ws5,ws7.

c) Iterate ten times for a total of one thousand epochs, using a batch size of four and a hidden layer neuron count of five, with a lag value of one for ws3, ws5, and ws7.

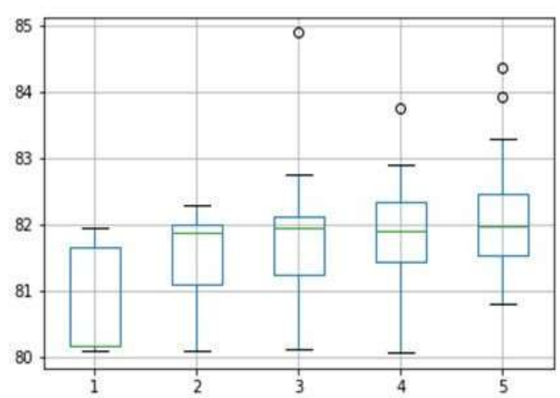


Fig. 6. RMSE Error Score Summary Plot with lag 3 considering epochs 1000

10.48047/jocaaa.2024.33.07.52

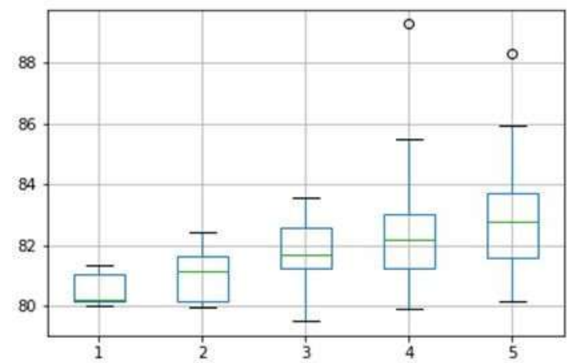


Fig. 7. Box Plot with lag 7 along with different HLNeurons

An Adam optimizer with a learning rate of 0.001 was used to train the model. To train the MLP, the size of the batch was set to 4. As MLP is differentiable, the parameters of the model can be learned through the backpropagation algorithm with the mean square error as a loss function, as shown in equation 7.

4 Result and Discussion

The evaluation of how well the experiment was carried out is the topic of discussion in this section. In the course of this investigation, there were three separate tests performed. The first experiment was done to determine the optimal time step for the model. The second step is to experiment to compare the performance of the test and train RMSE.

4.1 Evaluation Criteria and Measures

Calculating the root mean square (RMSE) is one of the methods used in this investigation for determining how well a model is doing. The root-mean-square error (RMSE) is a statistic that measures the gap between the actual value and the expected value.

During the process of making predictions about quantitative data, an absolute error is a useful metric for determining how accurately a certain model is being used.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_t - \hat{y}_t\right)^2} \quad (7)$$

Where yt is the actual value or observed value, ŷ t is the predicted value at a particular time t and n is the total no. of observations. RMSE is normally used

10.48047/jocaaa.2024.33.07.52

as a heuristic for training models and is also used to evaluate trained models for their accuracy or usefulness.

After each iteration of the experiment, the RMSE of the test set is printed out. A summary statistics table is generated after every run, including a row for every statistic and a configuration for every column.

A box and whisker plot is produced and kept for displaying the pattern of distribution of the test RMSE score for each particular type of instance. The plot demonstrates that each configuration exhibits a distinct spread in test RMSE box plot scores. The green line, which represents the median, tends to rise until it reaches three hidden layer neurons. After that point, it keeps growing with a lag of 7. However, for lag1, the median stays constant after 3 hidden layer neurons, as depicted in Figs. 6. and 7.

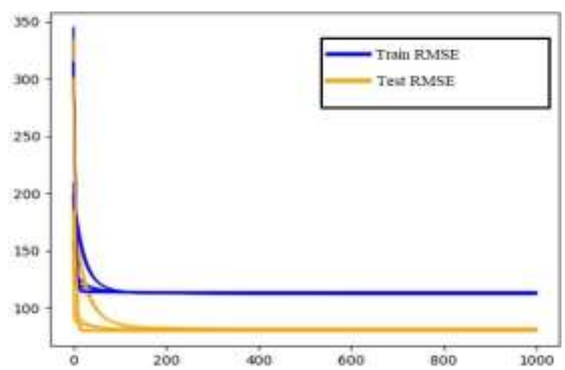


Fig. 8. Line Plot of Train and Test Performance of 1000 Epochs with lag1 on the commodity price Dataset with lag 1

The final train and test RMSE for each execution is displayed when the diagnostic is executed. The most fascinating aspect is the generated line plot's last line.

Following each training period, the train RMSE (blue) and test RMSE (orange) are shown on the line plot shown below in Fig. 8. Upon completion of about training epochs, the diagnostic plot reveals little variation in the train and test RMSE figures. Performance on both the test and the training sets approaches a flat line. This quick leveling out indicates that the model is at capacity and might need more data, such as lag observations or more neurons.

In this study, a framework is tested with a window dimension of three, where the total quantity of neurons in the layer that receives input is three, the number of neurotransmitters in the layer that is concealed is two, and the number of neurotransmitters in the layer that generates information is one and depicted in

10.48047/jocaaa.2024.33.07.52

fig.2. The epoch of ten is employed, and the values that were anticipated can be observed in Table 1.

Table 1. MLP Model configuring for 3I:2H:1O of 10 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	2	1	10	170.44
3	3	2	1	10	196.48
3	3	2	1	10	235.43
3	3	2	1	10	440.26
3	3	2	1	10	560.25

In this study, a framework is tested with a window dimension of three, where the total quantity of neurons in the layer that receives input is three, the number of neurotransmitters in the layer that is concealed is two, and the number of neurotransmitters in the layer that generates information is one. The epochs of twenty, fifty, hundred, five hundred & thousand were employed, and the values that were anticipated can be observed in Table II, Table III, Table IV, Table V & Table VI respectively.

Table 2. MLP Model configuring for 3I:2H:1O of 20 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	2	1	20	106.542
3	3	2	1	20	196.840
3	3	2	1	20	146.250
3	3	2	1	20	452.251
3	3	2	1	20	529.568

10.48047/jocaaa.2024.33.07.52

Table 3. MLP Model configuring for 3I:2H:1O of 50 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	2	1	50	133.959
3	3	2	1	50	393.420
3	3	2	1	50	1070.201
3	3	2	1	50	734.394
3	3	2	1	50	578.567

Table 4. MLP Model configuring for 3I:2H:1O of 100 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	2	1	100	488.51
3	3	2	1	100	187.89
3	3	2	1	100	360.98
3	3	2	1	100	480.56
3	3	2	1	100	520.25

Table 5. MLP Model configuring for 3I:2H:1O of 500 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	2	1	500	465.34
3	3	2	1	500	497.31
3	3	2	1	500	520.58
3	3	2	1	500	537.72
3	3	2	1	500	523.35

10.48047/jocaaa.2024.33.07.52

Table 6. MLP Model configuring for 3I:2H:1O of 1000 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	2	1	1000	514.52
3	3	2	1	1000	510.25
3	3	2	1	1000	298.03
3	3	2	1	1000	495.88
3	3	2	1	1000	555.35

In this empirical study, an organizational structure with a window factor of three is assembled for evaluation. In the aforementioned structure, the overall number of neurons in the segment that is capable of acquiring data is three, while the quantity of neurons in the layer by layer that is capable of concealing data is three, and the number of neurons in the layer that is capable of generating information is one.

Table 7. MLP Model configuring for 3I:3H:1O of 10 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	3	1	10	208.674
3	3	3	1	10	1047.023
3	3	3	1	10	513.508
3	3	3	1	10	307.493
3	3	3	1	10	229.305

10.48047/jocaaa.2024.33.07.52

Table 8. MLP Model configuring for 3I:3H:1O of 20 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	3	1	20	439.652
3	3	3	1	20	645.220
3	3	3	1	20	439.273
3	3	3	1	20	552.536
3	3	3	1	20	861.894

Table 9. MLP Model configuring for 3I:3H:1O of 50 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	3	1	50	261.853
3	3	3	1	50	759.635
3	3	3	1	50	490.048
3	3	3	1	50	569.435
3	3	3	1	50	493.258

Table 10. MLP Model configuring for 3I:3H:1O of 100 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	3	1	100	342.333
3	3	3	1	100	529.195
3	3	3	1	100	537.568
3	3	3	1	100	485.685
3	3	3	1	100	539.241

10.48047/jocaaa.2024.33.07.52

Table 11. MLP Model configuring for 3I:3H:1O of 100 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	3	1	100	342.333
3	3	3	1	100	529.195
3	3	3	1	100	537.568
3	3	3	1	100	485.685
3	3	3	1	100	539.241

Table 12. MLP Model configuring for 3I:3H:1O of 500 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	3	1	500	528.109
3	3	3	1	500	508.923
3	3	3	1	500	439.198
3	3	3	1	500	499.629
3	3	3	1	500	536.719

Table 13. MLP Model configuring for 3I:3H:1O of 1000 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	3	1	1000	491.789
3	3	3	1	1000	538.265
3	3	3	1	1000	515.682
3	3	3	1	1000	492.658
3	3	3	1	1000	534.250

10.48047/jocaaa.2024.33.07.52

Table 14. MLP Model configuring for 3I:5H:1O of 10 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	5	1	10	41.1994
3	3	5	1	10	514.495
3	3	5	1	10	729.046
3	3	5	1	10	463.1101
3	3	5	1	10	515.1522

Table 15. MLP Model configuring for 3I:5H:1O of 20 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	5	1	20	-159.969
3	3	5	1	20	726.998
3	3	5	1	20	463.101
3	3	5	1	20	726.998
3	3	5	1	20	510.585

Table 16. MLP Model configuring for 3I:5H:1O of 50 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	5	1	50	486.695
3	3	5	1	50	314.379
3	3	5	1	50	470.733
3	3	5	1	50	505.074
3	3	5	1	50	511.782

10.48047/jocaaa.2024.33.07.52

Table 17. MLP Model configuring for 3I:5H:1O of 100 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	5	1	100	311.665
3	3	5	1	100	483.535
3	3	5	1	100	470.733
3	3	5	1	100	508.226
3	3	5	1	100	514.187

Table 18. MLP Model configuring for 3I:5H:1O of 500 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	5	1	500	521.192
3	3	5	1	500	498.575
3	3	5	1	500	494.037
3	3	5	1	500	522.432
3	3	5	1	500	542.949

Table 19. MLP Model configuring for 3I:5H:1O of 1000 Epochs with predicted values

Ws=3	No. of Neurons in I/P layer	No. of Neurons with 1 Hidden layer	No. of Neurons in O/P layer	No. of Epochs	Predicted Value of Series
3	3	5	1	1000	510.529
3	3	5	1	1000	515.901
3	3	5	1	1000	525.217
3	3	5	1	1000	516.577
3	3	5	1	1000	530.007

Tables VII, VIII, IX, X, XI & XII show the projected readings based on the chosen time periods of the ten, twenty, fifty, five hundred, and thousand iterations.

10.48047/jocaaa.2024.33.07.52

In this empirical study, an organizational structure with a window factor of three is assembled for evaluation. In the aforementioned structure, the overall number of neurons in the segment that is capable of acquiring data is three, while the number of neurons in the layer by layer that is capable of concealing data is five and the number of neurons in the layer that is capable of generating information is one. Tables XIII,XIV,XV,XVI,XVII,XVIII show the projected readings based upon the chosen time periods of the ten, twenty, fifty, five hundred and thousand iterations. A detailed comparison of predicted prices across different MLP configurations and epochs is presented in Table XIX, showing the consistency and convergence of the model outputs.

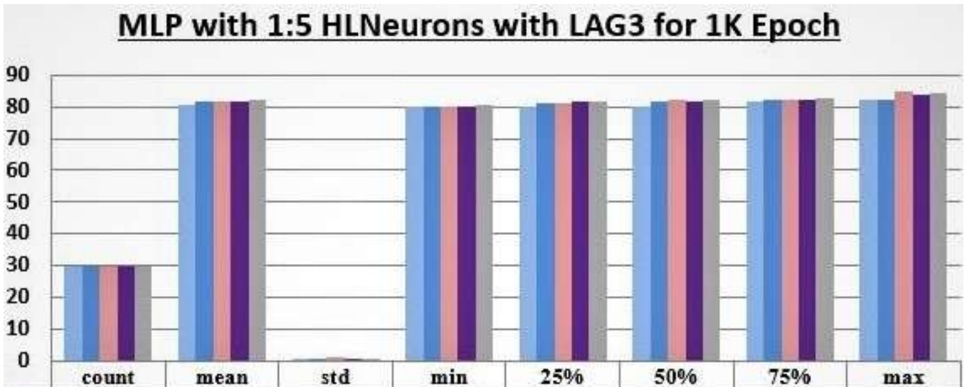
4.2 Optimal time step search

In time series prediction, the timestep is a hyperparameter that determines how many past data samples are used to predict future data, optimal time step may differ depending on the task to be solved. Several candidate's values were set and a a grid search was conducted to find the optimal timestep. Different timestep values were used for this study. In this study, an experiment was conducted to find a more suitable time step for the data of single agricultural commodity. In this experiment, the model was trained and tested, and its performance was measured while changing the timestep of the MLP-based model. Consequently, a lower MSE was recorded for agricultural commodities.

The experimental results show that the hidden layer with 1, 2, and 4 neurons, with lag5 for 1 k epochs, provides the most accurate mean value for predict the price of the commodity on the following day. Based on analysis of agricultural commodity price time series data, we find that when the number of hidden layer neurons is increased, the mean and standard deviation of root mean square error (RMSE), both rise at timestamps 3 and 5. It has been discovered that lag7 displays the lowest rmse value for 1k epochs and 30 counts, but lag3 displays the highest value for 50 percent. When looking at lag3, it has been shown that the standard deviation and minimum values both decrease for neurons in hidden layers 1 and 2, but then continue to increase after neuron 3. while at first, max grows and shrinks as a result of a rise in the number of neurons in the hidden layer, respectively. As the number of neurons in the hidden layer increases, the mean and standard deviation of lag5 continue to increaseincrease. It can be observed only when using lag7 and counting 30 for 1K epochs that the shortest possible RMSE is demonstrated and depicted in the form of a bar graph in Fig. 9., Fig. 10.

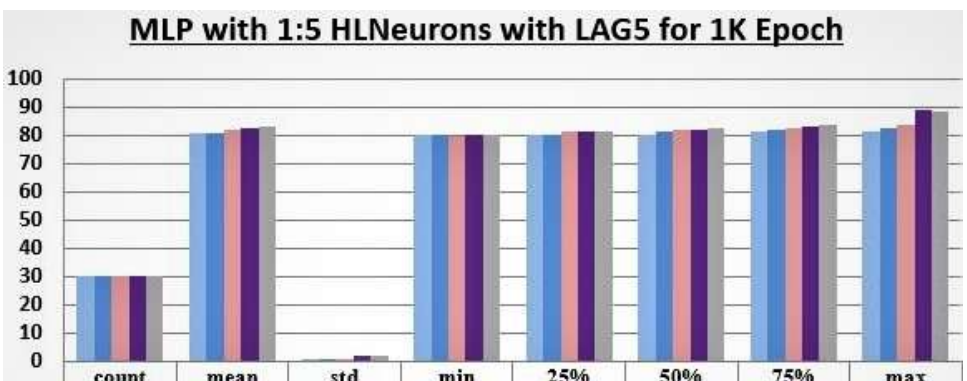
The research paper presents Tables I to XIX, which summarize predicted tomato prices using various MLP configurations, differing in input neurons, hidden layers, and number of epochs (10 to 1000) under a fixed window size. These tables illustrate how increasing training epochs and hidden neurons generally improves forecast stability and accuracy, with Table 13 and Table 19 capturing optimal results at 1000 epochs. The series of tables collectively demonstrate the model's evolving prediction capability, emphasizing how model depth and training time influence output precision in agricultural price forecasting.

10.48047/jocaaa.2024.33.07.52



	count	mean	std	min	25%	50%	75%	max
1	30	80.728861	0.778927	80.106556	80.165871	80.165899	81.668561	81.945158
2	30	81.475566	0.740062	80.106377	81.111731	81.885544	81.992012	82.288201
3	30	81.720348	1.013907	80.118111	81.252098	81.963978	82.124447	84.891281
4	30	81.883903	0.73838	80.074749	81.434881	81.908702	82.347127	83.756365
5	30	82.0688562	0.858036	80.802294	81.545653	81.981329	82.472252	84.359273

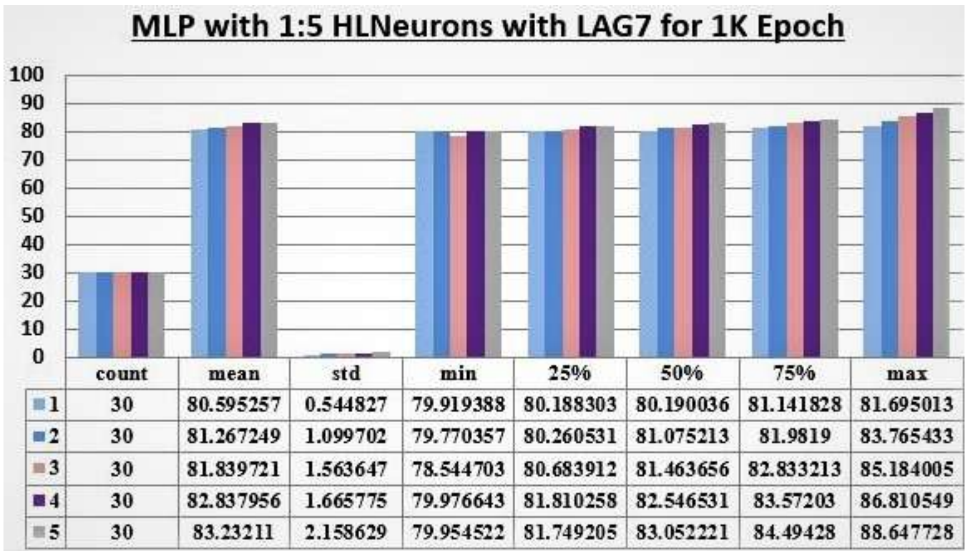
Fig. 9. RMSE Error Score Box Plot with lag 3 considering epochs 1000



	count	mean	std	min	25%	50%	75%	max
1	30	80.547415	0.492355	80.042407	80.17833	80.198977	81.04377	81.33887
2	30	81.019182	0.805004	79.961757	80.188228	81.133229	81.636113	82.449612
3	30	81.789237	1.049816	79.525644	81.260368	81.685008	82.565002	83.573028
4	30	82.567209	1.894939	79.917541	81.269002	82.184571	83.024591	89.26344
5	30	82.975939	1.739662	80.171279	81.620009	82.783047	83.720686	88.303367

Fig. 10. RMSE Error Score Box Plot with lag 5 considering epochs 1000

10.48047/jocaaa.2024.33.07.52



	count	mean	std	min	25%	50%	75%	max
1	30	80.595257	0.544827	79.919388	80.188303	80.190036	81.141828	81.695013
2	30	81.267249	1.099702	79.770357	80.260531	81.075213	81.9819	83.765433
3	30	81.839721	1.563647	78.544703	80.683912	81.463656	82.833213	85.184005
4	30	82.837956	1.665775	79.976643	81.810258	82.546531	83.57203	86.810549
5	30	83.23211	2.158629	79.954522	81.749205	83.052221	84.49428	88.647728

Fig. 11. RMSE Error Score Box Plot with lag 7 considering epochs 1k

The final evaluation of model predictions under the optimized configuration is depicted in Fig. 11., highlighting the model's generalization ability on unseen data.

5 Conclusion

The multilayer perceptron model is used in order to carry out the process of price forecasting for agricultural commodities. The MLPTSM model is used for the study of nonlinear pricing time series data. In the beginning stages of this experiment, data are gathered, and then the gathered data are subjected to preprocessing. Differentiation and normalization are operations that are carried out on the data. In order to provide an accurate price prediction, a model must first be trained using the training data and then evaluated using the test data after the data has been divided in a 70:30 ratio. In time series prediction, several timesteps such as 3,5 and 7 are explored for experimentation; nevertheless, it has been shown that the ideal time step will be 5, since it exhibits the smallest mean RMSE. According to our findings, which are derived from an examination of time series data pertaining to the prices of agricultural commodities, as the number of neurons in the hidden layer increases, both the mean and the standard deviation of the root mean square error (RMSE) grow at timestamps 3 and 5. In order to discover the timestep that worked best, many candidates for the value were established, and a grid search was performed. The values of the time-steps

10.48047/jocaaa.2024.33.07.52

varied throughout the investigation. For the purpose of this research, we focused on one agricultural commodity and performed experiments to determine which timestep would work best for its data. The model was trained and tested, and performance was monitored while adjusting the MLP-based model's timestep.

Declarations

Availability of data and material

The data sets analyzed during the current study are publicly available from the following sources:

1. Machine Learning India: http://ml-india.org/datasets/
2. National Horticulture Board (NHB): https://www.nhb.gov.in/OnlineClient/MISDailyReport.aspx
3. Data Portal India: https://data.gov.in/resource/annual-wholesale-price-index-agriculture-produce

Competing interests

The authors declare that they have no competing interests.

Funding

The authors did not receive any specific grant or funding from public, comercial, or not-for-profit sectors for conducting this research.

Authors' contributions

- **RK**: Conceptualization, data collection, model development, experiment design, manuscript writing, data analysis.
- **JP**: Supervision, Technical guidance, result interpretation, manuscript review.
- **SS**: Supervision, validation of methodology, refinement of experimental results, manuscript proofreading.

All authors read and approved the final manuscript.

Acknowledgements

The authors express special thanks to the **Government of India** and the **Horticulture Department** for providing the necessary data on agricultural commodity prices. Special appreciation is also extended to **Dr. Ramkrishna Rathore** for guidance and valuable suggestions throughout the development of this research paper. 7

10.48047/jocaaa.2024.33.07.52

References

1. Agriculture & Food Management.chapter-7," Information is provided by Ministry of Finance., Government of India, Economic Survey 2021-22 234-264.
2. Nowrouz Kohzadi, Milton S. Boyd, Bahman Kermanshahi, and Iebling Kaastra, "A comparison of artificial neural network and time-series model for forecasting commodity prices," NeuroComputing.
3. Rafik A. Aliev, Bijan Fazlollahi, and Rustam M. Vahidov, "Genetic algorithm-based learning of fuzzy neural networks.Part 1:feed-forward fuzzy neural network," Fuzzy Sets and Systems.
4. M. L. Shih, B. w. Huang, Nan-Hsing Chiu, C. Chiu, and W. Y. Hu, "Farm price prediction using case based reasoning approach-A case of broiler industry in Taiwan," Computer and Electronics in Agriculture 2009.
5. H. F. Zou, G. P. Xia, H. Y. Wang, and F. T. Yang, "An investingation and comparision of artificial neural network and time series model for Chinese food grain price forecasting.," NEURO COMPUTING: Elsevier January 2007.
6. Gan-qiong Li, Shi-wei Xu, and Zhe-min Li, "Short-Term price forecasting for Agro-products using Artificial Neural Networks," in Elsevier, 2010, pp. 278-287.
7. Celma Q. Ribeiro and Sydnei M. Oliveria, "A hybrid commodity price forecasting model applie d to the sugar-alcohol sector," Autralian Journal of Agricultural and Resource Economics, vol. 55, no. 2, pp. 180-198, April 2011.
8. Girish K. Jha and Kanchan Sinha, "Agricultural Price Forecasting Using Neural Network Model: an Innovative Information Delivery System," Agricultural Economics Research Review, vol. 26, no. 2, pp. 229-239, July-December 2013.
9. Lean Yu, Shouyang Wang, and Kin Lai Keung, "A neural network based non-linear metamodeling approach to financial time series forecasting.," Applied Soft Computing, vol. 9, pp. 563-574, August 2009.
10. Tao Xiong, Chongguang Li, Zhongyi Hu, and Lu Zhang, "A combination method for interval forecasting of agricultural commodity future prices," Knowledge-Based System, pp. 92-102, January 2015.
11. M. Subhasree and Mrs. C. Arun Priya, "Forecasting Vegetable price using Time Series Data," International Journal of Advanced Research, vol. 3, pp. 535-541, February 2016.
12. H. Ahumada and M. Cornejo, "Forecasting Food Prices: The case of Corn, Soyabean, and Wheat," International Journal of Forecasting, pp. 838-848, 2016.
13. Deina, Carolina, et al. "A Methodology for Coffee Price Forecasting Based on Extreme Learning Machines." Information Processing in Agriculture, vol. 9, no. 4, Dec. 2022, pp. 556–65. DOI.org (Crossref), https://doi.org/10.1016/j.inpa.2021.07.003.
14. Castro Campos, B. 2020. Are there asymmetric relations between real interest rates and agricultural commodity prices? Testing for threshold effects of US real interest rates and adjusted wheat, corn, and soybean prices. Empirical Economics 59 (1):371–94.
15. Costa, F. S. D., and P. Dolog. 2019. Collective embedding for neural context-aware recommender systems. Proceedings of the 13th ACM conference on recommender systems. Copenhagen, Denmark.
16. Mingsheng, F., Q. Hong, Y. Zhang, L. Lu, and Y. Liu. 2018. A novel deep learning-based collaborative filtering model for recommendation system. IEEE Transactions on Cybernetics 49 (3):1084–96.

10.48047/jocaaa.2024.33.07.52

17. Wenquan, X., and H. Hui. 2023. A novel FRBF-Type model for nonlinear time series prediction. Mathematical Problems in Engineering 2023:1–14.
18. Yeong Hyeon, G., D. Jin, H. Yin, R. Zheng, X. Piao, and S. Joon Yoo. 2022. Forecasting agricultural commodity prices using dual input attention LSTM. Agriculture 12 (2):256.
19. Wang, Yijia. "Agricultural Products Price Prediction Based on Improved RBF Neural Network Model." Applied Artificial Intelligence, vol. 37, no. 1, Dec. 2023.
20. Wang, Pengwen, et al. "An Improved Multilayer Perceptron Approach for Detecting Sugarcane Yield Production in IoT Based Smart Agriculture." Microprocessors and Microsystems, vol. 82, Apr. 2021, p. 103822
21. Anand, C. "Comparison of Stock Price Prediction Models Using Pre-Trained Neural Networks." Journal of Ubiquitous Computing and Communication Technologies, vol. 3, no. 2, July 2021, pp. 122–34.
22. Chi, Yeong Nain, and Orson Chi. "Modeling and Forecasting of Monthly Global Price of Bananas Using Seasonal ARIMA and Multilayer Perceptron Neural Network." Econometrics, vol. 25, no. 3, 2021, pp. 21–41.
23. Noh, Jooman, and Hong Chong Cho. "Forecasting System Marginal Price Using Multilayer Perceptron and Nonlinear Autoregressive Exogenous Model." Journal of the Korean Society of Mineral and Energy Resources Engineers, vol. 57, no. 6, Dec. 2020, pp. 585–92.
24. The Faculty of Science and Technology, Suan Sunandha Rajabhat University, Bangkok 10300, Thailand, et al. "Improving Dropout Forecasting during the COVID-19 Pandemic through Feature Selection and Multilayer Perceptron Neural Network." International Journal of Information and Education Technology, vol. 12, no. 9, 2022, pp. 851–57
25. Madhiarasan, Manogaran, and Mohamed Louzazni. "Different Forecasting Horizons Based Performance Analysis of Electricity Load Forecasting Using Multilayer Perceptron Neural Network." Forecasting, vol. 3, no. 4, Nov. 2021, pp. 804–38
26. Xie, Yuhong, et al. "A Two-Stage Short-Term Load Forecasting Method Using Long Short-Term Memory and Multilayer Perceptron." Energies, vol. 14, no. 18, Sept. 2021, p. 5873
27. Sheela, K. Gnana, and S. N. Deepa. "Review on Methods to Fix Number of Hidden Neurons in Neural Networks." Mathematical Problems in Engineering, vol. 2013, 2013, pp. 1–11.
28. Keynia, Farshid. "A New Feature Selection Algorithm and Composite Neural Network for Electricity Price Forecasting." Engineering Applications of Artificial Intelligence, vol. 25, no. 8, Dec. 2012, pp. 1687–97.
29. Wang, Xiao, et al. "A Machine Learning Approach to Forecasting Carry Trade Returns." Applied Economics Letters, vol. 29, no. 13, July 2022, pp. 1199–204.
30. Menhaj, Mohammad Hossein, and Mohammad Kavoosi-Kalashami. "Developing a Hybrid Forecasting System for Agricultural Commodity Prices (Case Study: Thailand Rice Free on Board Price)," Ciˆencia Rural, vol. 52, no. 8, 2022
31. Makridis, Georgios, et al. "A Deep Learning Approach Using Natural Language Processing and Time-Series Forecasting towards Enhanced Food Safety." Machine Learning, vol. 112, no. 4, Apr. 2023.
32. Liu, Chang, et al. "Food Price Dynamics and Regional Clusters: Machine Learning Analysis of Egg Prices in China." China Agricultural Economic Review, vol. 15, no. 2, May 2023.
33. An, Wuyue, et al. "Text-based Soybean Futures Price Forecasting: A Two-stage Deep Learning Approach." Journal of Forecasting, vol. 42, no. 2, Mar. 2023

10.48047/jocaaa.2024.33.07.52

34. Cheung, Liege, et al. "Using a Novel Clustered 3D-CNN Model for Improving Crop Future Price Prediction." Knowledge-Based Systems, vol. 260, Jan. 2023
35. Atalan, Abdulkadir. "Forecasting Drinking Milk Price Based on Economic, Social, and Environmental Factors Using Machine Learning Algorithms." Agribusiness, vol. 39, no. 1, Jan. 2023
36. Sridhar, Adithya, et al. "Global Impact of COVID-19 on Agriculture: Role of Sustainable Agriculture and Digital Farming." Environmental Science and Pollution Research, vol. 30, no. 15, Mar. 2022.
37. Ulussever, Talat, et al. "Estimation of Impacts of Global Factors on World Food Prices: A Comparison of Machine Learning Algorithms and Time Series Econometric Models." Foods, vol. 12, no. 4, Feb. 2023
38. Gülerce, Mustafa, and Gazanfer Ünal. "FORECASTING OF OIL AND AGRICULTURAL COMMODITY PRICES: VARMA VERSUS ARMA." Annals of Financial Economics, vol. 12, no. 03, Sept. 2017, p. 1750012.