**A Comparison of Time Series and Machine Learning Approaches for Forecasting Weekly Price of Garbled Black Pepper**

**Abstract**

The present study has made an attempt to identify the best forecasting model to predict weekly price of garbled black pepper from January, 2000 to December, 2020, in Kochi market of Kerala, India. The volatility in prices of black pepper throughout the year poses a significant challenge for both farmers and consumers, being a perennial crop. Understanding the predictability of these price fluctuations in the near future is crucial for devising relevant policy recommendations. Consequently, price forecasting of black pepper is of paramount importance. Both time series and machine learning models have been used to forecast weekly prices of garbled black pepper. Models like Seasonal Autoregressive Moving Average model (SARIMA), Time-delay Neural Network (TDNN) model, and Long Short-Term Memory model (LSTM) have been tried in the study to forecast weekly garbled black pepper price series. Based on the accuracy measures of the models fitted, TDNN(13:8s:1l) was the best model for forecasting the weekly price of garbled black pepper, for Kochi market of Kerala.

**Keywords**: Garbled black pepper, SARIMA, TDNN, LSTM, Forecasting

**INTRODUCTION**

Black pepper, known as the 'king of spices,' stands as one of the most popular and widely used seasonings, often found alongside salt on dinner tables worldwide. India, holding the position of the world's third-largest producer according to the International Pepper Community (2023), assumes a crucial role as both a significant consumer and exporter of black pepper. The states of Kerala and Karnataka contribute substantially to the nation's overall output, with Kerala ranking second in terms of black pepper acreage (76,160 ha) and production (33.29 MT), albeit seventh in productivity (0.44 MT/ha) (GOI, 2023). The historical market value of pepper played a pivotal role in shaping the city of Kochi of Kerala, India into a hub of international commerce. Kochi proudly hosts India's first exclusive pepper exchange, established by the Indian Pepper and Spice Traders Association (IPSTA), known for its well-regulated operations by traditional players, ensuring a lack of defaults on supply or delivery and minimizing volatility.

Forecasting agricultural commodity prices holds great significance for farmers, governments, and agribusiness industries. Such forecasts assist policymakers, producers, and consumers in making informed decisions, ultimately mitigating risks associated with price fluctuations (Gu et al., 2022). Recent records from the Spices Board indicate that the price of black pepper in the Kochi market has exceeded Rs.500/kg, marking the highest price since 2022. This surge underscores the highly volatile nature of prices of black pepper, posing a significant challenge for both farmers and consumers. Given the perennial nature of the crop, the substantial price variations within a year present a major concern that needs to be addressed. Understanding the extent to which these fluctuations can be forecasted for the foreseeable future is essential for future planning. Thus, analysis of the price series of black pepper is inevitable.

Time series forecast models are of various forms, broadly categorized into linear models, including Autoregressive Integrated Moving Average (ARIMA), and nonlinear models using machine learning techniques.

 Box- Jenkins had introduced ARIMA model(1970s) in times series forecasting and the model was popularized due to its statistical properties and methodology for model building. Price forecasting of agricultural commodities are being done for ages for various crops using traditional time series approaches. However, studies have found that in reality these series often have unknown nonlinear structure, which can be better understood using machine learning techniques like artificial neural network (ANN). A multivariate, non-linear, non-parametric statistical technique that is driven by data and is self-adaptive is called an artificial neural network (ANN). Its advantage being its flexible functional form and it is a universal functional approximator (Zhang *et. al*.,1998). Mahato *et. al*., in their study on sunflower and soyabean price data concluded that ANN is a better forecasting model for forecasting agriculture commodity price than the ARIMA model. Kumari *et al.* (2023) conducted a study to analyse and compare the efficiency of different traditional models like ARIMA and SARIMA to the deep artificial intelligence techniques like ANN and RNN in forecasting the prices of banana in Gujarat on the time series data and empirical result showed that RNN was the best fitted model among all other models of prediction due to less error accuracy measures. The M4 competition experimented by Gilliand (2022) showed that while pure machine learning methods performed poorly, hybrid approaches combining ML with statistical techniques outperformed benchmarks, demonstrating the potential of integrated forecasting models.

**METHODOLOGY**

 In this study, time series data on weekly average prices of garbled black pepper at Kochi market from January 2000 to December 2020, collected from Spices Board, Kochi, Kerala, India was used for evaluating and comparing forecast performance of different models. The data for subsequent period i.e., from January 2021 to December 2022 was used for model validation. All model building and forecasting was done using R software and ‘Keras’ and ‘Tensorflow’ packages of Python.

**Time Series Model**

**I. Seasonal Auto Regressive Integrated Moving Average (SARIMA)**

In the ARIMA model introduced by Box and Jenkins (1976), the estimated value of a variable is considered to be a linear combination of past values and past errors. The Seasonal Autoregressive Integrated Moving Average (SARIMA) model stands as a versatile and robust approach for time series forecasting, seamlessly incorporating both non-seasonal and seasonal components. Expressed as SARIMA (p, d, q) x (P, D, Q) S, this model encompasses parameters for non-seasonal autoregressive (AR), differencing (I), and moving average (MA) aspects, alongside their seasonal counterparts. The non-seasonal components (p, d, q) capture temporal patterns, while the seasonal elements (P, D, Q) with a seasonality 'S' account for recurring seasonal patterns. This multiplicative model offers a comprehensive framework for the analysis and prediction of time series data, enabling a nuanced understanding of both short-term and long-term trends.

The seasonal components of the SARIMA (Seasonal Autoregressive Integrated Moving Average) model consist of autoregressive (AR) and moving average (MA) expressions, represented as Φ(Bs) and Θ(Bs), respectively, with Bs indicating the seasonal lag. The non-seasonal components are similarly expressed through φ(B) and θ(B). Seasonal differencing, defined as the difference between a value and a lagged value that is a multiple of s (with s = 12 for monthly data), is given by $(1-B\_{12)})y\_{t}=y\_{t}-y\_{t-12}$. This process aims to create a stationary series, making the differences approximately uniform across each month and effectively removing both seasonal trends and non-stationary seasonal random walks. If a trend exists in the data, non-seasonal differencing is also necessary, often achieved through a first non-seasonal difference to detrend the data. The mathematical formulation of the

$SARIMA (p, d, q) (P, D, Q)\_{s}$ model in terms of lag polynomials is presented,

$ \left(1-\sum\_{i=1}^{p}φ\_{i }B^{i }\right)\left(1-\sum\_{i=1}^{p}φ\_{i }B^{is }\right)\left(1-B\right)^{d}(1-B^{s})^{D} Y\_{t}=(1-\sum\_{j=1}^{q}θ\_{j}B^{js} )(1-\sum\_{j=1}^{Q}Θ\_{j}B^{s} )ε\_{t}$

The main stages in setting up a forecasting ARIMA model are: (i) Model identification, (ii) Parameters estimation, (iii) Diagnostic checking and (iv) Forecasting. (Brockwell *et.al.,* 2002)

**II. Machine Learning Models:**

 **(i) Time-delayed Neural Network Model (TDNN)**

Neural networks excel in modeling the relationships between inputs and outputs, even in the presence of noisy data. One of the most significant benefits of neural networks lies in their capability to capture intricate non-linear relationships without requiring predefined assumptions about the nature of these relationships (Haykin, 1999).

The architecture of Time Delay Neural Networks (TDNN) includes an input layer for external data, one or more hidden layers introducing non-linearities to the model, and an output layer delivering the desired outcome. TDNN's capability to effectively model non-linear systems, coupled with its elevated forecasting accuracy, has substantially enhanced its attractiveness for time series forecasting applications. A single hidden layer neural network has the capacity to approximate any non-linear function, provided an adequate number of hidden nodes and a sufficient amount of training data points (Jha and Sinha, 2013).

An example of such an architectural design is a Time-Delay Neural Network (TDNN), which is employed in the present research. The incorporation of time delays in neural networks is inspired by neurobiology, as single delays are widely observed in the brain and hold significance in the neurobiological processing of information. In TDNN, the activation function for node *i* at time *t* can be expressed as follows:

$$y\_{t}=\left(t\right)=f\left(\sum\_{j-1}^{q}\sum\_{d-0}^{p}w\_{ij}(t-d)y\_{j}(t-d)\right)$$

where $y\_{i}(t)$ the output node *i* at time *t*, $w\_{ij}(t)$ is the connection weight between node *i* and *j* at time t, *p* is the number of tapped delays, *q* is the number of nodes connected to node *i* from preceding layer, d denotes the time delays and *f* is the activation function, typically the logistic sigmoid**.**

In this research study, attention is specifically directed towards situations where tapped delays exist exclusively in the input layer. The configurations of TDNN involving a solitary hidden layer can be represented as I:Hs:Ol, where I denotes the number of nodes in the input layer, H represents the number of nodes in the hidden layer, O signifies the number of nodes in the output layer, s indicates the logistic sigmoid transfer function, and l denotes the linear transfer function. Figure 1 provides a visual representation of a time delay neural network (TDNN).



 **Figure 1. Architecture of TDNN model**

Developing a TDNN model usually requires the utilization of both training and testing samples. Here, 80% of the weekly prices of black pepper (data) is used as training set and rest 20% is used as the testing set. An accurate TDNN model is found using ‘nnetar’ package in R software.

**(ii) Long Short-Term Memory Model (LSTM)**

 A Recurrent Neural Network (RNN) is a type of artificial neural network designed for processing sequential data by incorporating feedback loops that allow information to persist and be used in subsequent steps.

 LSTM, a type of deep learning RNN model , is widely used for time series forecasting due to its effectiveness in capturing long-term dependencies. Unlike other models such as feedforward control neural networks and Support Vector Machines, LSTM's feedback control network is beneficial as it maintains a memory state, allowing it to consider correlations in lagged values. The sequential data processing capability of RNNs, including LSTM, makes them especially useful for time series forecasting.

The primary limitation of traditional RNNs is the vanishing gradient problem, where the gradient diminishes and eventually approaches zero (Hochreiter *et*. *al*.). This issue is addressed by the LSTM network model, a specific type of RNN that incorporates memory and forget cells. LSTM is composed of read, write, and delete operations using different cells in the hidden layers enabled by the three gates: input gate, output gate and forget gate.

The LSTM model was implemented using the Python libraries Keras and TensorFlow. The dataset was split into 80% for training and 20% for testing. Prior to training, the data underwent normalization and differencing during preprocessing. The LSTM model consisted of three layers: input, hidden, and output, with a total of 126,452 parameters. These parameters were utilized for training the model, where predictions on the training data were made during the training process. A loss function calculated the error between these predictions and actual target values, with the goal of adjusting weights and biases to minimize this training loss.

The training process occurred over multiple epochs, each epoch involving an update of the model's weights based on the training data. The Mean Absolute Error (MAE) was employed as the measure to continue the epoch until it reached a minimum. Subsequently, the trained LSTM model was tested using the testing data. Forecast evaluation metrics, including Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE), were computed for model assessment.

Determining hyperparameter values involved a search, testing various combinations, and selecting values that yielded optimal results for the final model. The chosen hyperparameters included those for the neuron input layer, sequence length determining LSTM memory duration, epoch count to address overfitting, neuron count for hidden and output layers, as well as the activation and optimization functions, as summarized in Table 1.

**Table 1. Values/Functions used for the parameters in the LSTM model**

|  |  |
| --- | --- |
| **Parameters** | **Chosen value/function** |
| Neurons input layer | 100 |
| Sequence length | 10 |
| Epochs | 50 |
| Neurons hidden layer | 100 |
| Neurons output layer | 1 |
| Activation Function | tanh |
| Optimization Function | adam |

**Model Diagnostics**

**Portmanteau test: Ljung-Box test**

 It is crucial to carry out diagnostic checks to evaluate the suitability of the model and recommend any necessary revisions after the tentative model has been fitted to the data (Armstrong, 2009). Non-significance of autocorrelations of residuals can also be tested by using Portmanteau test. Ljung-Box $Q^{\*}$ statistic is the commonly used Portmanteau test and is given by

 $Q^{\*}= n(n+2)\sum\_{k=1}^{h}\frac{1}{n-k}r\_{k}^{2}$

where $‘h’$ is the maximum lag being considered and ‘$n$’ is the number of observations in the series. If the residuals are white noise, the statistic $Q^{\*}$has a chi-square ($χ^{2})$distribution with $(h-m)$ degrees of freedom where ‘m’ is the number of parameters fitted in the model which has been fitted. The series is not white noise when $Q^{\*}$is greater than table value of chi-square at $(h-m)$ degrees of freedom.

**Forecast Accuracy Measures**

 Best suitable model among SARIMA and TDNN models are evaluated using forecast accuracy measures like MAPE and RMSE values (Prathima, 2018). These measures are calculated as given below:

 $MAPE $= $\frac{1}{n} \sum\_{i=1}^{n}\frac{|Y\_{t }- \hat{Y\_{t}}|}{Y\_{t}} \*100$ where,

 $Y\_{t}$– Actual price at time t

 $\hat{Y\_{t}}$- Fitted price

$ n$- Number of observations

 Root Mean Square Error (RMSE) is given by,

$ RMSE =$ $\sqrt{\frac{1}{n}\sum\_{i=1}^{n}e\_{t}^{2}}$

where, $e\_{t}=Y\_{t }- \hat{Y\_{t}}$, is the difference between actual and fitted price

 $ n$ - Number of observations

**RESULTS AND DISCUSSION**

1. **SARIMA Model**

The SARIMA model is estimated after transforming weekly price data of garbled black pepper into a stationary series. The model selection is based on criteria such as minimum Mean Absolute Percentage Error (MAPE) and minimum Root Mean Square Error (RMSE). Based on model accuracy measures, SARIMA (2,1,2) (1,0,0)52 appropriately fitted the series for forecasting prices. The accuracy measures for this model are presented in Table 2, and the results indicate that SARIMA (2,1,2) (1,0,0)52 outperforms other models as it satisfies the criteria of minimum MAPE and RMSE.

 **Table 2. SARIMA(p,d,q)(P,D,Q)52 model**

|  |  |  |
| --- | --- | --- |
| **Model** | **MAPE** | **RMSE** |
| **SARIMA(2,1,2)(1,0,0)52** |  **3.23** | **9.96** |

The SARIMA (2,1,2)(1,0,0)52 model equation is as follows:

 (1−0.328*B*−0.639*B*2)(1-*B*)(1+0.025*B*52)yt=(1+0.238*B*+0.704*B2*)*ϵt*​

Actual and predicted weekly price of garbled black pepper using SARIMA(2,1,2)(1,0,0)52 model is plotted in Figure 2.



**Figure 2. Actual and fitted price series of of weekly price of garbled black pepper**

**using SARIMA(2,1,2) (3,0,2)12**

The adequacy of the model is also tested using the value of Box-Pierce Q statistics and was found to be insignificant. So, overall we can say SARIMA(2,1,2)(1,0,0)52model show satisfactory result, among different ARIMA models.

**Table 3. Ljung-Box ‘Q’ statistic for residuals of SARIMA(2,1,2)(1,0,0)52 model**

|  |  |
| --- | --- |
| **Statistic** | **p-value** |
| **93.023** | **0.6502NS** |

 **NS: Non-significant**

1. **TDNN Model**

Time delayed neural network (TDNN) model is fitted for weekly price of garbled black pepper. The best time lagged neural network with single hidden layer is found for each series by conducting experiments with the basic cross validation method. Out of a total of 112 neural network structures, a neural network model with thirteen lagged observations as input nodes and eight hidden layers (13:8s:1l) performed better than other competing models in respect of forecasting accuracy measures. This means that most accurate price forecast for the given series is obtained when the price of thirteen preceding weeks is used as inputs.

The selected TDNN model is described in Table 4 along with the forecasting accuracy measures for both training and testing set.

**Table 4. Model accuracy measures for TDNN model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **No. of parameters** | **MAPE** | **RMSE** |
| **Train** | **Test** | **Train** | **Test** |
| 13:8s:1l | 121 | 3.95 | 1.99 | 8.49 | 6.78 |

Actual and predicted weekly garbled black pepper price series using TDNN model is shown in Figure 3.****

**Figure 3. Actual and predicted values of weekly price of garbled black pepper using TDNN**

Model adequacy was evaluated using the value of Ljung-Box ‘Q’ statistic (Table 5)and it is found to be insignificant. Thus, concluding TDNN model to be a good fit.

**Table 5. Ljung-Box ‘Q’ statistic for residuals of TDNN model**

|  |  |
| --- | --- |
| **Statistic** | **p-value** |
| **159.57** | **0.3783NS** |

 **NS: Non-significant**

1. **LSTM Model**

The LSTM model is developed with three layers by performing epoch of 50 times. During the training process, the model makes predictions on the training data, and the loss function is used to calculate the error between these predictions and the actual target values. The training loss is the average of these errors across all training samples. The goal of training is to adjust the weights and biases of the model to minimize this training loss. After each epoch, the weights of the model get updated and the new epoch works on those updated values that process continues in every epoch. The parameters used for the LSTM model are provided in Table 1 and various accuracy measures for LSTM model are provided in Table 6. Mean Absolute Error (MAE) was used as a measure to continue the epoch until MAE reaches to a minimum which is shown in Figure 4.

**Figure 4. Training loss for LSTM model**

**Table 6. Model accuracy measures of LSTM model for weekly price of garbled**

 **black pepper**

|  |  |
| --- | --- |
| **Model Accuracy** | **Value** |
| RMSE | 16.36 |
| MAPE | 5.57 |

Actual and predicted weekly price series of garbled black pepper using LSTM model is plotted in Figure 5.



**Figure 5. Actual and predicted weekly price series of garbled black pepper using LSTM model**

Model adequacy tested using the value of Ljung-Box ‘Q’ statisticisfound to be insignificant (Table 7), indicating adequacy of LSTM model.

**Table 7. Ljung-Box ‘Q’ statistic for residuals of LSTM model**

|  |  |
| --- | --- |
| **Statistic** | **p-value** |
| **624.52** | **0.610NS** |

 **NS: Non-significant**

**Comparison of models**

As a next step, various models price forecasting models fitted in the study are compared in order to decide the best model to forecast weekly price of garbled black pepper at Kochi market. As observed in Table 8, TDNN model had the least value for MAPE and RMSE and is selected as the best among all the models fitted.

**Table 8. Comparison of price forecasting models for weekly price of garbled black pepper**

|  |  |  |
| --- | --- | --- |
| **Model** | **MAPE** | **RMSE** |
| SARIMA(2,1,2)(1,0,0)52 | 3.23 | 9.96 |
| **TDNN(13:8s:1l)** | **1.99** | **6.78** |
| LSTM | 5.57 | 16.36 |

Actual and forecasted weekly price of garbled black pepper using TDNN model (Figure 6) are in agreement.

The MAPE value are found to be 3.11 and 3.16 respectively based on the actual and forecasted weekly price of garbled black pepper using the TDNN model for the year 2021 and 2022.

**Figure 6. Actual, predicted and forecasted plot of TDNN model of weekly price of garbled black pepper**

The study revealed that the TDNN model, with a MAPE of 1.99, could accurately capture the pattern of weekly price of garbled black pepper. Hence, this model has emerged as a highly effective forecasting model, especially when compared to traditional statistical models such as ARIMA. These conventional approaches often assume linearity and fixed seasonality, limiting their effectiveness in capturing the complex, nonlinear, and time-dependent patterns common in agricultural price data. In contrast, TDNN uses time-lagged inputs to learn from historical data trends, enabling it to model temporal relationships more accurately. This makes TDNN particularly suitable for predicting prices of commodities like black pepper, which are influenced by a variety of dynamic factors including climate conditions, international trade policies, and market demand.

When compared to other machine learning models, TDNN holds a unique advantage in handling sequential data. While advanced models like LSTM networks also perform well in time series forecasting, TDNN offers a simpler structure and faster training time, making it more accessible for applications with limited computational resources. Its demonstrated performance—reflected in the lowest Mean Absolute Percentage Error (MAPE) among models tested for black pepper price forecasting—highlights its practical value in agriculture. By enabling more accurate predictions, TDNN supports better decision-making for farmers, traders, and policymakers, ultimately contributing to market stability, improved income planning, and agricultural policy formulation.

**CONCLUSION**

In this study, various time series models were employed to determine the most effective model for predicting prices of black pepper in the Kochi market. The forecasted prices were then compared with real-time prices, revealing that the TDNN model exhibited predictions that closely aligned with the actual weekly prices of processed black pepper. The forecasted values indicated an upward trend in prices in the coming months. Identifying the optimal forecasting model and achieving accurate predictions of market prices holds significant implications for farmers, consumers, wholesalers, and government decision-making. Short-term forecasting of future prices is particularly crucial for farmers in planning black pepper production throughout the seasons, considering resource availability and profitability compared to other crops. This enables strategic production planning to anticipate favourable prices in the near future, aiding informed decisions before planting and facilitating crop selection. With insight from forecasted prices, farmers can decide whether to sell their produce immediately or opt for storage post-harvest to secure a remunerative price. With global supply disruptions driven by climate change, unregulated trade practices, and export restrictions, black pepper prices have seen sharp fluctuations—posing risks to farmers, traders, and food industries alike. From a policy perspective, reliable price forecasting can support the formulation of targeted interventions such as minimum support prices, export-import regulations, and farmer subsidies. It also encourages digital integration in agriculture, aligning with global goals for smarter, climate-resilient farming. In sum, the TDNN model not only aids in navigating current market uncertainties but also strengthens the economic resilience of agriculture-dependent regions and supports evidence-based policymaking.

Disclaimer (Artificial intelligence)

Option 1:

Author(s) hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript.

**Yes, we declare that No generative AI technologies has used during the preparation and editing of this manuscript. And please note that is an original research article.**

Option 2:

Author(s) hereby declare that generative AI technologies such as Large Language Models, etc. have been used during the writing or editing of manuscripts. This explanation will include the name, version, model, and source of the generative AI technology and as well as all input prompts provided to the generative AI technology

Details of the AI usage are given below:

1.

2.

3.

**REFERENCES**

Armstrong, J.S. (2009). Selecting Forecasting Methods, *Principles of Forecasting : A Handbook for Researchers and Practitioners*, M.A: Kluwer Academic Publishers.

Brockwell, P.J. and Davis, R.A. (2002) Introduction to Time Series and Forecasting. 2nd Edition, Springer, New York.

 Gu, Y.H., Jin, D., Yin, H., Zheng, R., Piao, X., Yoo, & S.J. (2022). Forecasting Agricultural Commodity Prices Using Dual Input Attention LSTM. *Agriculture,* 12, 256.

Gilliland, M. (2020). The value added by machine learning approaches in forecasting. International Journal of Forecasting, 36(1), 161-166.

Indiastat. (2023). Indiastat. home page [on-line]. Available: <https://www.indiastat.com>.[ 28 Nov. 2023].

International Pepper Community. (2023). International Pepper Community. Home page [on-line]. Available: <https://www.ipcnet.org/> [28 Nov.2023].

 Jha, G.K., & Sinha, K. (2013). Agricultural Price Forecasting Using Neural Network Model: An Innovative Information Delivery System*. Agricultural Economics Research Review,* 26 (2), 229 – 239.

 Haykin, S. (1999). *Artificial neural networks* (2nd ed.). Pearson Education.

Hochreiter, S., Schmidhuber. J..(1997) Long short-term memory, *Neural computation*, 9

 (8)1735–1780.

Kumari, P., Goswami, V., N. H, and Pundir, R.S. (2023) Recurrent neural network architecture for forecasting banana prices in Gujarat, India. *Plos One.*18(6): 27-56.

 Makridakis, S., & Hibbon, M. (1979). Accuracy of forecasting : An empirical investigation. *Journal of the Royal Statistical Society*, 41(2), 97-145.

Mahto, A.K., Biswas, R., Ahmed, A. Short Term Forecasting of Agriculture Commodity Price by Using ARIMA: Based on Indian Market, pages 452{461. *Advances in Computing and Data Sciences*. ICACDS 2019. Springer, Singapore, 452-461.

Prathima, C.M. (2018). Statistical modelling of arrivals and prices of tender coconut in Maddur market. M.Sc. Thesis. University of Agricultural Sciences, Bengaluru, Karnataka.

Zhang, G., Patuwo, B. E., Hu, & M. Y. (1998). Forecasting with artificial neural networks: The state of the Art. *International Journal of Forecast*. 14: 35-62.