A COMPARATIVE ANALYSIS OF MACHINE LEARNING METHOD FOR COMMODITY DEMAND FORECASTING

Atcharawan Kamsai[*], Nuwee Wiwatwattana[2]

Abstract

Forecasting demand for agricultural commodities is essential for optimizing production, managing inventory, staying competitive, mitigating risks, and informing policy decisions. This research study focuses on evaluating and comparing the performance of various forecasting models in this context. Five models were analyzed: Prophet, Multilayer Perceptron (MLP), Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing, and Time Series Mixer (TSMixer). Evaluation metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) were utilized to assess the performance of each model. The findings revealed that the MLP model exhibited the highest forecasting performance, with the lowest RMSE of 5.01, MAE of 3.12, and a MAPE of only 0.0083, indicating high accuracy and low error. This study underscores the potential of neural network techniques in forecasting complex data and suggests avenues for further research and model development. By enhancing the understanding and utilization of forecasting models, this research contributes to the efficiency and sustainability of the agricultural industry and other related sectors.

Keywords: Time Series Forecasting, Machine Learning, Deep Learning, Prophet, ARIMA, Exponential Smoothing, TSMixer (Time Series Mixer)

---

[1] Data Science, Faculty of Science, Srinakharinwirot University, Bangkok, 10110, Thailand

[2] Faculty of Science, Srinakharinwirot University, Bangkok, 10110, Thailand

* Corresponding author:Tel.: 062-4469415 E-mail address: atcharawan.kamsai@g.swu.ac.th

1. Introduction

Agricultural soft commodities such as coffee, sugar, cocoa, and cotton are seasonal and climate-dependent crops. These commodities significantly impact the global economy and the livelihoods of millions of farmers and traders. Forecasting the demand for soft commodities is a challenging task that requires considering various factors, including weather patterns, consumer preferences, and geopolitical events. Taking sugar as an example, the current weakening of the U.S. dollar has supported the prices of various commodities, including sugar. Additionally, increased sugar production in Brazil has put pressure on sugar prices, while India's sugar supply is expected to decrease after the Indian Food Minister requested sugar exporters to limit exports to 6 million tons by May 2023. Consequently, global demand for raw sugar is projected to increase in the latter half of 2023, as countries like China and Indonesia cannot further delay sugar imports.

Thus, changes in sugar prices reflect fluctuations in global market conditions and domestic conditions in sugar-exporting countries, including exchange rate variations, production changes, and demand shifts. In 2023, global sugar demand is forecasted to reach 180.05 million tons, while India's sugar production is estimated at around 36 million tons, and Brazil's sugar output is projected to be approximately 42.6 million tons. Therefore, the global sugar market in 2023 will depend on the demand and supply dynamics in major sugar-exporting countries.

Traditional forecasting methods rely on historical data and statistical assumptions to capture demand patterns. However, these methods have several limitations, such as the need for manual parameter tuning, difficulties in handling non-stationary data [1], and the inability to incorporate external variables. Furthermore, these methods may not capture the complex relationships present in agricultural commodity markets. Conventional techniques for forecasting agricultural commodity demand, such as Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing, and Prophet, have limitations in handling complex and uncertain data and often require prior knowledge about the data and relationships between relevant variables. In contrast, neural network, and deep learning techniques like Multilayer Perceptron (MLP) and Time Series Mixer (TSMixer) can learn from large datasets and automatically extract relevant features without prior knowledge or human intervention, potentially better capturing the complexity and uncertainty of the data.

The main research question is whether neural network and deep learning techniques like MLP and TSMixer outperform traditional methods such as Prophet, ARIMA, and Exponential Smoothing in forecasting the demand for agricultural soft commodity time series data. To address this question, the researcher will compare the performance of these methods on agricultural commodity datasets and evaluate them using appropriate metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Additionally, trade-offs between these methods, such as computational time and forecasting accuracy, will be considered.

2. Materials and Methods

This section describes the datasets, forecasting techniques, and evaluation metrics employed to compare the performance of deep learning methods against traditional methods for forecasting the demand for agricultural soft commodities.

2.1 Datasets

The study utilizes time series data for various agricultural soft commodities, such as coffee, sugar, cocoa, and cotton. The datasets contain historical information on commodity prices, trading volumes, and relevant market indicators.

2.2 Traditional Forecasting Methods

Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average (ARIMA) is a statistical model commonly used for time series forecasting. It combines autoregression (AR), differencing (I), and moving average (MA) components to capture different aspects of the time series data. The autoregressive component models the relationship between an observation and a number of lagged observations. The differencing component removes trends and seasonality from the data by taking differences between consecutive observations. The moving average component models the relationship between an observation and a residual error from a moving average model applied to lagged observations. ARIMA models are typically identified by three parameters: p, d, and q, which represent the number of autoregressive terms, the degree of differencing, and the number of moving average terms, respectively. ARIMA models are widely used in various fields, including economics, finance, and meteorology, for forecasting time series data.

$$\boldsymbol{Y(t) = a0 + \beta 1Y(t-1) + \beta 2Y(t-2) + \ldots + \beta pY(t-p) + b1e(t-1) + b2e(t-2) + \ldots + bme(t-m)}$$

Where:

Y(t) is the data in the time series at time t.

a0 is a constant value.

$\beta_1, \beta_2, \ldots, \beta_p$ are the AR parameters.

b1, b2, ..., bm are the MA parameters.

e(t) is the error of the model at time t.

Exponential Smoothing

Exponential Smoothing is a simple and widely used method for time series forecasting. It works by calculating a weighted average of past observations, with more recent observations being given higher weights. The weights decay exponentially as observations become older, resulting in a smoother forecast that places greater emphasis on recent data. Exponential Smoothing is particularly useful for forecasting time series data with no clear trends or seasonal patterns. It is easy to implement and requires minimal computational resources, making it suitable for applications where computational efficiency is important. Exponential Smoothing models are commonly used in industries such as inventory management, sales forecasting, and capacity planning.

$$F(t) = \alpha Y(t-1) + (1-\alpha)\beta F(t-1)$$

Where:

F(t) is the forecast value at time t.

α is the parameter of the model.

β is the leveling parameter.

Y(t-1) is the data in the time series at time t-1.

F(t-1) is the forecast value at time t-1.

Prophet

Prophet is a time series forecasting model developed by Facebook that is specifically designed to handle datasets with seasonal patterns, holidays, and other recurring events. It is a flexible and customizable model that can automatically detect and incorporate such patterns into its forecasts. Prophet employs a decomposable time series model consisting of trend, seasonality, and holiday components. The trend component captures non-periodic changes in the data, while the seasonality component captures periodic fluctuations. The holiday component allows the model to account for the effects of holidays and other special events on the time series data. Prophet is particularly useful for forecasting time series data with complex patterns and is widely used in various domains such as retail sales forecasting, financial forecasting, and demand forecasting.

$$Y(t) = g(t) + \beta s(t) + e(t)$$

Where:

Y(t) is the data in the time series at time t.

g(t) is the trend component.

βs(t) is the seasonal component.

e(t) is the error of the model at time t

2.3 Neural Networks

Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) is a type of artificial neural network (ANN) that consists of multiple layers of interconnected neurons. It is a feedforward neural network, meaning that information flows in one direction, from the input layer through the hidden layers to the output layer. MLP is trained using the backpropagation algorithm, which adjusts the weights of connections between neurons to minimize prediction errors. MLP is known for its ability to capture complex patterns and nonlinear relationships in data, making it suitable for a wide range of tasks, including time series forecasting. In time series forecasting, MLP can learn from historical data to make predictions about future values. It is particularly effective when the relationship between input and output variables is nonlinear or when the data has complex patterns that cannot be captured by traditional statistical models.

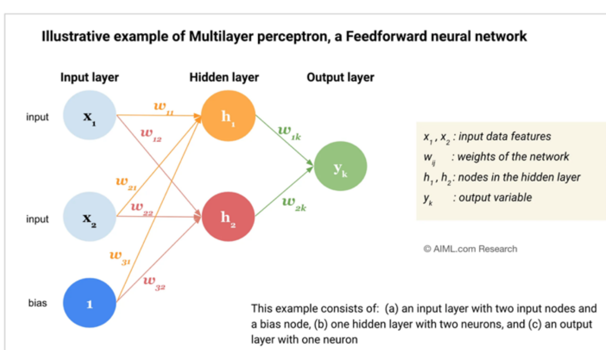

Figure 1. Multilayer Perceptron (MLP

Source: https://aiml.com/what-is-a-multilayer-perceptron-mlp/

2.4 Deep Learning Methods

Time Series Mixer (TSMixer)

Time Series Mixer (TSMixer) is a relatively new approach to time series forecasting that combines temporal and static convolutional networks. TSMixer aims to leverage the strengths of both types of networks to improve forecasting accuracy. The temporal convolutional network processes the temporal aspect of the data, capturing sequential patterns and dependencies over time. The static convolutional network processes static features that do not change over time, such as categorical variables or external factors. By combining these two networks, TSMixer can effectively model both temporal and static aspects of the data, leading to more accurate forecasts. TSMixer is still an area of active research, and its performance may vary depending on the specific characteristics of the time series data and the chosen network architecture.

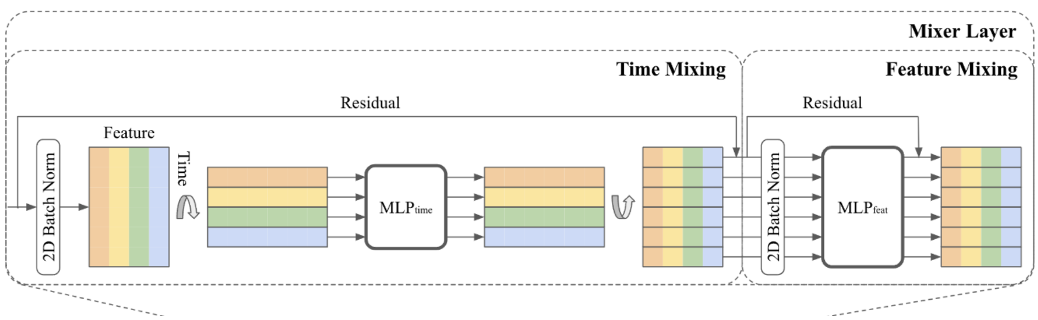

Figure 2. Time Series Mixer (TSMixer)

Source: https://blog.research.google/2023/09/tsmixer-all-mlp-architecture-for-time.html

2.4 Data Preprocessing

Before applying the forecasting methods, the datasets will undergo necessary preprocessing steps, such as handling missing values, data normalization, and feature engineering. Techniques like sliding window or sequence generation will be used to prepare the time series data for deep learning models.

2.4.1 Data Exploration and Analysis (Exploratory Data Analysis)

The EDA section delves into the exploration and analysis of the dataset. This involves techniques such as data visualization, summary statistics, and identifying patterns or trends within the data.

a. Initial Data Check

As the data used in this research is time series data, the researcher performed an initial check of the data types for each column and converted the date column to an index. This step facilitates the analysis of time series data in the subsequent stages. By efficiently handling the time series data, the researcher can effectively utilize the available time for data analysis.

b. Exploratory Data Analysis (EDA)

Through the Exploratory Data Analysis (EDA), the researcher uncovered several noteworthy findings. First, a histogram was created for the 'volume' column, an effective tool for visualizing data distribution. This allowed the researcher to observe the distribution pattern, frequency, and volatility of the trading volume data, revealing an irregular distribution. Next, a bar plot was generated for the 'commodity' column, enabling the visualization of the categorical distribution of data and facilitating comparisons of observation counts across different commodity groups. Furthermore, the researcher plotted the distributions of the 'open', 'high', 'low', and 'close' variables for each commodity type. These distribution plots provided insights into the market price distributions for individual commodities and allowed for comparisons of price distributions across different commodities.

Finally, time series plots were created for the 'volume' variable, segregated by commodity type. These plots helped identify patterns in the daily changes of trading volumes and enabled comparisons of volume fluctuations among different commodities. This exploratory data analysis helped the researcher gain a deeper understanding of the data and its underlying relationships, which will be beneficial for preparing the data for further analysis. The next step will be to conduct the Augmented Dickey-Fuller (ADF) test on the selected commodity time series data. The ADF test will assist the researcher in determining whether the time series data is stationary or not, a crucial step in preparing the data for forecasting using machine learning techniques. Knowing the stationarity of the data will enable the researcher to select appropriate models for efficient forecasting.

c. Augmented Dickey-Fuller (ADF Test) for Commodity Data

The Augmented Dickey-Fuller (ADF) Test is a statistical test used to check if a given time series data is stationary or non-stationary. It tests the null hypothesis of a unit root in the time series. A p-value less than 0.05 indicates that the data is stationary. For the commodity dataset used in this research, the ADF test gave a p-value of 0.15, which is greater than 0.05. This means the null hypothesis is failed to the 5% significance level. In other words, the data does not exhibit strong evidence of stationarity and is likely non-stationary.

Additionally, the ADF test statistic value is around -2.35. This statistic is used to determine the stationarity of a time series. The more negative this value is, the stronger the evidence against the null hypothesis of non-stationarity. Typically, a more negative ADF statistic indicates stationarity.

Given the relatively high p-value and not very negative test statistic value obtained here, the researcher may need to consider applying differencing techniques or other pre-processing steps to make this commodity data stationary before using it for forecasting purposes.

Random Length Lumber dataset, which has been processed on Google Colab for this research, the following plots are generated to illustrate the analysis and findings.

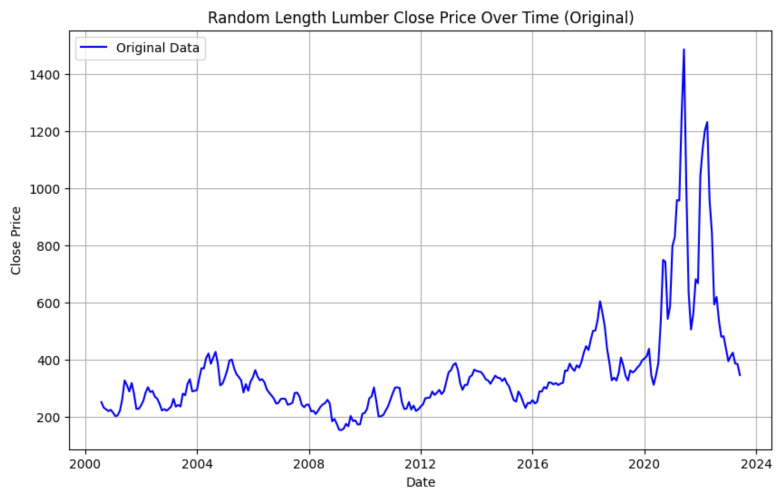

Figure 3. Time Series Plot of Close Prices Over Time Before Differencing

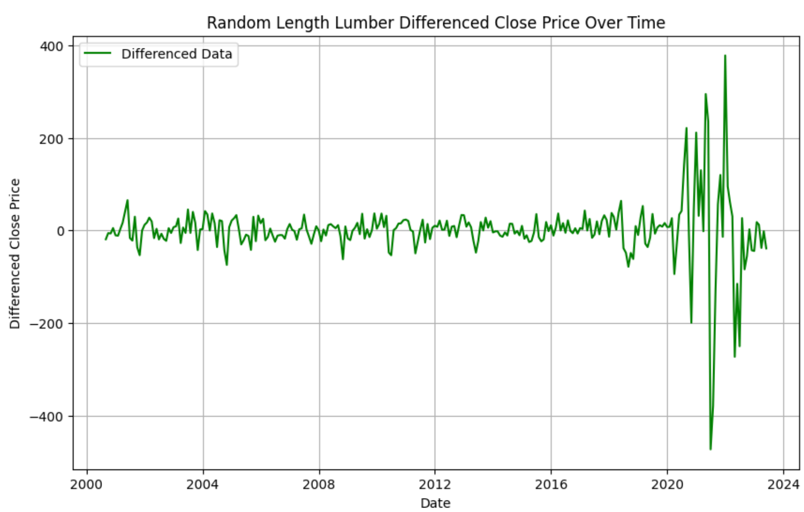

Figure 4. Time Series Plot of Close Prices Over Time After Differencing

The Augmented Dickey-Fuller (ADF) test results suggest the raw commodity data likely violates the stationarity assumption required for many traditional time series forecasting methods like Autoregressive Integrated Moving Average (ARIMA). Appropriate data transformations may be required before building forecasting models.

2.5 Model Training and Evaluation

The traditional and deep learning forecasting models will be trained on the preprocessed datasets using appropriate techniques, such as backpropagation and gradient descent optimization for the deep learning models, and maximum likelihood estimation or least squares estimation for the traditional methods. The models will be evaluated using the following metrics:

a) Root Mean Squared Error (RMSE)

The Root Mean Squared Error (RMSE) is a metric used to assess the accuracy of a predictive model. It is calculated by taking the square root of the mean squared error, which is the average of the squared differences between predicted and actual values. RMSE measures the model's accuracy by considering the magnitude of all errors that occur, regardless of their direction (positive or negative). A lower RMSE value indicates higher model accuracy.

$$RMSE = \frac{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}}{n}$$

Where:

y = true value

$\hat{y}$ = predicted value

n = number of data points

b) Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) measures the average absolute difference between the predicted and actual values. It is calculated as the sum of the absolute errors divided by the number of data points. In other words, MAE quantifies the size of the errors without considering their direction. A high MAE indicates low model efficiency.

$$MAE = \sum_{i=1}^{n}(\frac{\hat{y}_i - y_i)}{n}$$

Where:

n = number of samples

$\hat{y}$ = predicted value

y = true value

c) Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) is another metric used to evaluate the accuracy of a predictive model. It is calculated by taking the average of the absolute percentage errors, where each error is expressed as a percentage of the actual value. MAPE measures the model's accuracy by considering the relative size of all errors that occur, regardless of their direction. Like RMSE, a lower MAPE value indicates higher model accuracy.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|A_{t-}F_t|}{A_t}$$

Where:

n: number of iterations for the summation calculation

At: actual value

Ft: forecast value

Evaluating forecasting models using multiple metrics like MAE, RMSE, and MAPE provides a comprehensive understanding of the model's performance. These metrics capture different aspects of the errors, such as the average magnitude (MAE), the sensitivity to large errors (RMSE), and the relative size of errors (MAPE). By considering these metrics together, researchers can make informed decisions about the suitability and effectiveness of different forecasting models for their specific applications.

2.6 Comparative Analysis

The performance of the neural network methods (MLP and TSMixer) was compared against the traditional forecasting methods (Prophet, ARIMA, and Exponential Smoothing) using the evaluation metrics mentioned above. Statistical tests, such as hypothesis testing or analysis of variance (ANOVA), were employed to determine the significance of the performance differences. Additionally, trade-offs between the methods, such as computational time and forecasting accuracy, were analyzed.

Through this systematic approach, the research aims to provide insights into the effectiveness of deep learning techniques like MLP and TSMixer for forecasting the demand for agricultural soft commodities compared to traditional methods such as Prophet, ARIMA, and Exponential Smoothing.

Bottom of Form

3. Results and Discussion

The research aimed to compare the forecasting performance of various models, including Facebook Prophet, Multilayer Perceptron (MLP), Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing, and Time Series Mixer (TSMixer), in predicting commodity demand. Here are the key findings and discussions:

Multilayer Perceptron (MLP): The MLP model demonstrated exceptional forecasting performance, achieving the lowest error values across all evaluation metrics (RMSE, MAE, and MAPE). Its ability to capture complex patterns in commodity demand data effectively contributed to its superior accuracy. The neural network architecture of MLP allowed it to model intricate nonlinear relationships, outperforming traditional methods.

ARIMA and Exponential Smoothing: These traditional time series forecasting methods showed competitive performance, although with slightly higher error values compared to MLP. They effectively captured linear patterns and trends in the data, making them viable options in certain scenarios.

Prophet: The Prophet model, designed for time series forecasting, displayed the poorest performance among the evaluated techniques. Its high error values across all metrics suggest limitations in effectively capturing the underlying patterns and seasonality present in commodity demand data.

Time Series Mixer (TSMixer): TSMixer, which combines temporal and static convolutional networks, showed promising potential with low RMSE and MAE values. However, its infinite MAPE value indicates issues in accurately forecasting certain values, potentially due to its inability to handle outliers or special events in the data effectively. Further investigation and refinement may be required to enhance its forecasting capabilities.

The evaluation of forecasting models using the commodity demand dataset revealed significant performance differences across various techniques. The Multilayer Perceptron (MLP) model emerged as the clear frontrunner, demonstrating superior accuracy and minimal forecasting errors compared to other models.

Specifically, the MLP achieved the lowest error values across all evaluation metrics, with an RMSE of 5.01, MAE of 3.12, and an exceptionally low MAPE of 0.0083. These results underscore the MLP's exceptional ability to capture complex, nonlinear patterns in commodity demand data, outperforming traditional time series forecasting methods.

Importantly, these findings underscore the importance of selecting appropriate forecasting techniques for accurate demand forecasting, as inaccurate estimations can lead to inventory management and production planning issues, directly impacting organizational operations and decision-making. Overall, the results demonstrate the superiority of deep learning methods, particularly the MLP model, in capturing complex patterns and delivering accurate forecasts for commodity demand. While traditional methods like ARIMA and Exponential Smoothing can still be useful in certain scenarios, they may struggle to match the performance of advanced deep learning techniques.

It is essential to consider the trade-offs between model complexity, training time, and data requirements when selecting a forecasting method. Deep learning models, while highly accurate, often require significant computational resources and larger datasets for training. In contrast, traditional methods may be more computationally efficient but less accurate in capturing nonlinear patterns.

Future research could explore ensemble methods that combine the strengths of traditional and deep learning techniques, as well as investigate the potential of transfer learning and domain adaptation for commodity demand forecasting. Additionally, incorporating exogenous variables, such as market conditions and economic indicators, may further enhance the forecasting accuracy of these models.

Comparison of Model Results:

Model	RMSE	MAE	MAPE
Prophet	280.91	219.03	38.11
MLP (Multilayer Perceptron)	5.01	3.12	0.0083
ARIMA (Autoregressive Integrated Moving Average)	116.41	66.9	136.04

Exponential Smoothing	114.23	65.61	182.83
TSMixer	0.8398	0.7055	inf

The TSMixer model also exhibited promising performance, attaining the lowest RMSE of 0.8398 and MAE of 0.7055 among all models evaluated. However, its MAPE value was infinite, indicating potential issues in accurately forecasting certain values. This limitation could be attributed to the model's inability to effectively handle outliers or special events present in the data, warranting further investigation and refinement of the TSMixer approach.

The ARIMA (Autoregressive Integrated Moving Average) and Exponential Smoothing models demonstrated competitive forecasting capabilities, with comparable RMSE and MAE values. Nonetheless, their MAPE values were substantially higher than the MLP model, suggesting higher percentage errors in their forecasts. While these traditional time series techniques effectively captured linear patterns and trends, they may struggle to match the performance of advanced deep learning methods in modeling intricate nonlinear relationships.

In contrast, the Prophet model, an additive regression model designed for time series forecasting, exhibited the poorest performance among the evaluated techniques. Its high error values across all metrics (RMSE: 280.91, MAE: 219.03, MAPE: 38.11) indicate significant limitations in effectively capturing the underlying patterns and seasonality present in the commodity demand data.

It is crucial to note that these findings are specific to the dataset and evaluation criteria employed in this study. The selection of an appropriate forecasting model may depend on various factors, including the nature of the data, computational resources available, and the specific requirements of the forecasting task at hand.

4. Research Summary

The research compared the forecasting performance of Prophet, Multilayer Perceptron (MLP), Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing, and Time Series Mixer (TSMixer) models in predicting commodity demand. Evaluation metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) were used. The results indicated that the Multilayer Perceptron (MLP) model outperformed others, achieving an RMSE of 5.01, MAE of 3.12, and remarkably low MAPE of 0.0083, demonstrating high accuracy and minimal errors. This comprehensive assessment highlighted MLP's superiority in forecasting commodity demand over Prophet, ARIMA, Exponential

Smoothing, and TSMixer models. The study's implications for businesses and decision-makers are significant, emphasizing the potential for enhanced accuracy and reliability in demand prediction through MLP. Its ability to minimize forecasting errors enables better decision-making in inventory management, production planning, and resource allocation. Adopting advanced forecasting models like MLP can lead to improved operational efficiency, cost reduction, and increased competitiveness in volatile markets. Moreover, the research underscores the importance of innovative methodologies in demand forecasting, facilitating continuous improvement and optimization of business processes. Embracing these insights empowers organizations to navigate market dynamics effectively, capitalize on emerging opportunities, and achieve sustainable growth in today's dynamic business landscape.

Top of Form

5. Acknowledgments

The researchers would like to express their gratitude to the Graduate School of Srinakharinwirot University for supporting the presentation of this research work. The authors extend their sincere appreciation for the opportunity to disseminate their findings.

Reference

[1] S. Wang, C. Li, and A. Lim, "Why Are the ARIMA and SARIMA not Sufficient," Apr. 2019, Accessed: Aug. 20, 2023. [Online]. Available: https://arxiv.org/abs/1904.07632v3

[2] E. M. de Oliveira and F. L. Cyrino Oliveira, "Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods," Energy, vol. 144, pp. 776–788, Feb. 2018, doi: 10.1016/J.ENERGY.2017.12.049.

[3] M. Navratil and A. Kolkova, "Decomposition and forecasting time series in business economy using prophet forecasting model," Central European Business Review, vol. 8, no. 4, pp. 26–39, 2019, doi: 10.18267/J.CEBR.221.

[4] A. M. C. H. Attanayake, S. S. N. Perera, and U. P. Liyanage, "COMBINING FORECASTS OF ARIMA AND EXPONENTIAL SMOOTHING MODELS," Adv Appl Stat, vol. 59, no. 2, pp. 199–208, Dec. 2019, doi: 10.17654/AS059020199.

[5] B. Kumar Jha and S. Pande, "Time Series Forecasting Model for Supermarket Sales using FB-Prophet," Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021, pp. 547–554, Apr. 2021, doi: 10.1109/ICCMC51019.2021.9418033.

[6] S. Arslan, "A hybrid forecasting model using LSTM and Prophet for energy consumption with decomposition of time series data," PeerJ Comput Sci, vol. 8, p. e1001, Oct. 2022, doi: 10.7717/peerj-cs.1001.

[7] V. Kramar and V. Alchakov, “Time-Series Forecasting of Seasonal Data Using Machine Learning Methods,” Algorithms, vol. 16, no. 5, 2023, doi: 10.3390/a16050248.

[8] S. J. G., C. P., and U. M. Rajasekaran, “Time-series analysis and Flood Prediction using a Deep Learning Approach,” in 2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), 2022, pp. 139–142. doi: 10.1109/WiSPNET54241.2022.9767102.

[9] Md. M. Rahman et al., “A comprehensive study and performance analysis of deep neural network-based approaches in wind time-series forecasting,” J Reliab Intell Environ, vol. 9, no. 2, pp. 183–200, 2023, doi: 10.1007/s40860-021-00166-x.

[10] J. Guo, P. Lin, L. Zhang, Y. Pan, and Z. Xiao, “Dynamic adaptive encoder-decoder deep learning networks for multivariate time series forecasting of building energy consumption,” Appl Energy, vol. 350, p. 121803, 2023, doi: https://doi.org/10.1016/j.apenergy.2023.121803.

[11] A. Zeng, M. Chen, L. Zhang, and Q. Xu, “Are Transformers Effective for Time Series Forecasting?” Oct. 2022. doi: 10.48550/arXiv.2205.13504.

[12] S.-A. Chen et al., “TSMixer: An All-MLP Architecture for Time Series Forecasting,” Mar. 2023, Accessed: Mar. 31, 2024. [Online]. Available: https://arxiv.org/abs/2303.06053v5