How does the choice of sequence length impact model accuracy and generalization in time series forecasting with sales data using LSTM models?

Kanchana Weerasinghe, Navodya Ranasinghe
Dalarna University
Borlänge, Sweden

Abstract— **Accurate time series forecasting is crucial for retail and e-commerce businesses, serving as a foundation for effective demand prediction and inventory management. This study explores the impact of varying window sizes and training epochs on the performance of Long Short-Term Memory (LSTM) models in sales forecasting. Utilizing historical stock price data sourced from Yahoo Finance, we employed a sliding window approach to generate input-output pairs suitable for LSTM modeling. Five different window sizes (5, 10, 20, 60, and 100) and a range of epochs (5, 10, 20, and 30) were systematically tested. Performance was assessed using key metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2). Our findings indicate that the optimal configuration of a window size of 10 and 5 epochs achieved exceptional results, with an MSE of 0.0000009, MAE of 0.002497, and an R^2 of 0.948379, underscoring its accuracy and strong model fit. In contrast, other configurations demonstrated significantly poorer performance, highlighting the critical role of hyperparameter optimization in LSTM models. This research offers valuable insights for practitioners aiming to enhance their forecasting strategies, and future work will explore additional hyperparameters and feature engineering techniques to further improve model performance and robustness.**

Keywords- Long Short-Term Memory (LSTM); Window Size; Training Epochs, Hyperparameter Optimization

I. INTRODUCTION

Accurate time series forecasting plays a pivotal role in retail and e-commerce, serving as a foundation for effective demand prediction, inventory management, and promotional planning. As the marketplace becomes increasingly competitive, businesses rely on precise forecasting to optimize operations and maximize profitability. In this context, neural networks, particularly Long Short-Term Memory (LSTM) models, have gained prominence due to their inherent ability to capture complex temporal dependencies and learn from sequential data (Hochreiter & Schmidhuber, 1997).

A critical aspect of designing LSTM models is the choice of sequence length, or window size, which refers to the number of past time steps utilized to predict future values. The sequence length directly influences the model's capacity to learn relevant patterns in the data, balancing the trade-off between capturing enough historical context and avoiding overfitting (Bontempi et al., 2013). Inadequate sequence lengths may lead to the model's failure to learn significant trends, while excessively long sequences can introduce noise and redundancy, detracting from the model's generalization ability (Bishop, 2006).

This project aims to systematically explore the impact of varying sequence lengths on the accuracy and generalization of LSTM models in sales forecasting. By addressing the key research questions of how the choice of sequence length affects model learning and determining the optimal window size for accurate predictions, we seek to provide actionable insights for businesses. Understanding these dynamics will help organizations enhance their forecasting strategies, thereby improving decision-making processes and ultimately leading to more effective inventory and demand management.

Through an experimental approach involving historical sales data, this study will evaluate the performance of LSTM models trained with different window sizes. By measuring performance metrics such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), we aim to elucidate how sequence length influences predictive accuracy and generalization. The findings from this research are expected to guide businesses in selecting the most effective window size tailored to their specific sales data, paving the way for more robust forecasting methodologies.

II. LITERATURE REVIEW

Time series forecasting has been extensively studied, especially in the context of retail and e-commerce, where accurate demand prediction can significantly enhance operational efficiency. The use of LSTM models for time series data has gained traction due to their capability to learn from long sequences and handle vanishing gradient problems (Hochreiter & Schmidhuber, 1997). Numerous studies have demonstrated the effectiveness of LSTMs in capturing complex temporal dependencies (Zhang et al., 2018). However, much of the existing literature often overlooks the impact of sequence length on model performance, leaving a significant gap in understanding how this parameter influences the learning process and generalization capabilities of LSTM models.

Research on the optimal sequence length for time series forecasting has been limited. Studies such as those by Bontempi et al. (2013) and Bishop (2006) emphasize the trade-off between capturing sufficient historical context and avoiding overfitting but do not provide comprehensive guidelines on determining the ideal window size for various datasets. Furthermore, while some research explores fixed sequence lengths, there is a lack of systematic analysis on how varying sequence lengths can lead to differences in

performance metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

Most existing research does not consider the combined impact of varying window sizes and the number of training epochs on the accuracy and generalization of LSTM models. This oversight can lead to suboptimal model performance, as both sequence length and training duration significantly influence the learning process. By addressing this gap, this project will contribute to the body of knowledge on time series forecasting, offering practical insights for businesses seeking to optimize their forecasting strategies through informed choices about sequence lengths and training parameters in LSTM models.

III. METHOD DESCRIPTION

A. The Dataset

The dataset utilized in this study comprises historical stock price data sourced from Yahoo Finance. It encompasses a total of 13,805 entries, organized into seven distinct features that are essential for the analysis and forecasting of financial time series.

The dataset covers a significant time range, starting from February 2, 1970, to October 25, 2024. This extensive period facilitates the exploration of various market conditions, including periods of economic stability, growth, and volatility, which are crucial for robust model training and evaluation. The dataset consists of the following key features:

Table 1.0. Variable Description

Variable Name	Description
date	The date corresponding to each trading day (MM-DD-YYYY)
Open	The price of the financial instrument at the market open
High	The highest price recorded during the trading day
Low	The lowest price recorded during the trading day
Close	The price at which the financial instrument closed at the end of the trading day
Adj Close	The closing price adjusted for dividends and stock splits
Volume	The total number of shares traded during the day

B. Data Pre-Processing

Initially, the dataset was loaded using the Pandas library, and the 'Date' column was converted to a datetime format to facilitate time series analysis. Subsequently, the dataset was filtered to create a training subset comprising data from February 2, 1970, to June 1, 2024, focusing specifically on the 'Open' price. This filtered dataset was indexed by date, ensuring that time-related operations could be efficiently performed.

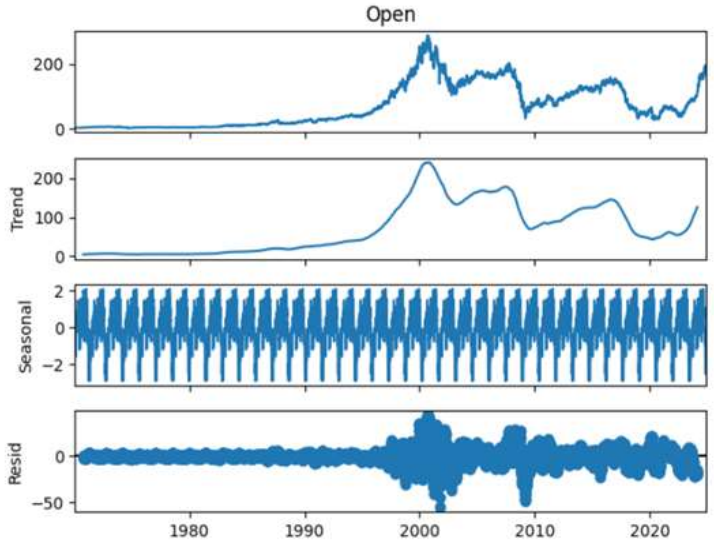

Figure 1.0 Seasonal Trends, Outliers over Time

Figure 2.0 Opening Price over Time

According to Figure 1.0 an exploratory data analysis was carried out to check for any seasonal trends and outliers. To overcome the outliers, smoothing technique was treated by finding the most appropriate smoothing window.

Figure 2.0 displays the variation of open price from year 1970 to 2024.

To standardize the data and improve model performance, the StandardScaler from scikit-learn was utilized to normalize the training data. This normalization transformed the features

to have zero mean and unit variance, aiding in the convergence of the neural network during training.

For model preparation, a sliding window approach was employed to create input-output pairs suitable for LSTM modeling. The prepare_data function generated sequences of past observations, defined by a specified window size, to predict future values. This setup enabled the model to learn the temporal dependencies inherent in the data effectively. The resulting training dataset comprised sequences of historical stock prices that the LSTM would use to make future predictions.

C. Data Mining Method

The LSTM model was constructed using PyTorch and comprised of two LSTM layers followed by a dropout layer and a fully connected output layer. This architecture was specifically designed to capture the intricate temporal relationships within the time series data. The model was initialized with a specified input size, hidden size, and output size, ensuring it was adequately configured for the prediction task.

During the training procedure, the normalized dataset was split into training and validation sets, with 80% allocated for training. The training process involved looping through a defined number of epochs, where, for each batch of training data, gradients were reset, and backpropagation was executed to optimize the model parameters using the Adam optimizer.

To optimize the LSTM model's performance, a systematic exploration of window size and the number of epochs was conducted. Five window sizes were tested: 5, 10, 20, 60, and 100. These values represent varying historical contexts, allowing the model to explore different lengths of past data for predicting future values. Smaller window sizes, such as 5 and 10, capture recent trends and short-term fluctuations, which may be beneficial for detecting immediate changes in stock prices. Intermediate window sizes, such as 20 and 60, incorporate broader patterns, including potential monthly or seasonal influences. The largest window size, 100, was selected as the filtered dataset contained only 102 records, maximizing the training data available to the model while still allowing enough data for validation and testing. Additionally, the number of training epochs indicating the number of complete passes through the training data varied with values of 5, 10, 20, and 30. Each combination was evaluated based on three key performance metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 score. During the training phase, training and validation losses were tracked across epochs, and the loss curves were plotted for each window size and epoch combination. After completing the evaluations, the best configuration from each window size was identified based on the highest R^2 score.

The top R^2 scores for each window size determined the final configurations: (5, 20), (10, 5), (20, 30), (60, 20), and (100, 10). These configurations were then used for further evaluation, with each combination passed through the LSTM model a two-layer LSTM with dropout and a fully connected output layer to generate predictions. After training, the make_predictions function generated forecasts for the prediction period, applying inverse transformation to bring values back to the original scale. For each configuration, the predicted values were plotted against the actual open prices for visual analysis. To evaluate the model's performance further, predictions were generated from June 2, 2024, to October 25, 2024, immediately following the training period, which covered February 2, 1970, to June 1, 2024, using the make_predictions function. For each combination of window size and epoch count, the model was trained using the train_model function, which returned key performance metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) scores. Highest R^2 with best fitting Actual vs Prediction plot of Window size and epochs combination was selected as the optimal configuration in accurately forecasting of stock prices using an LSTM model.

Figure 3.0 displays the step-by-step process we have followed in this study.

Table 4.0 Different combinations testing with window size and epochs

Selected Window Size	Corresponding epochs
5	5
5	10
5	20
5	30
10	5
10	10
10	20
10	30
20	5
20	10
20	20
20	30
60	5
60	10
60	20
60	30
100	5
100	10
100	20
100	30

Table 4.0 displays the different combinations used to find the epoch for each window size.

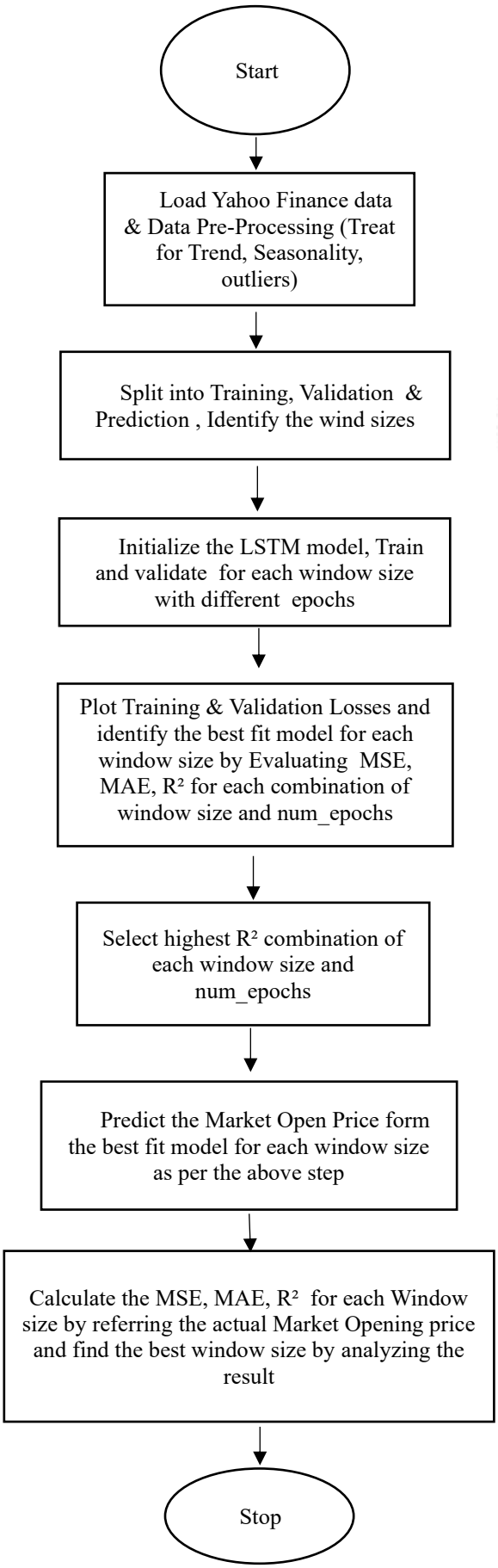

Figure 3.0 Flow diagram of Methodology

IV. Results and Analysis

The model successfully identified key temporal patterns in the sales data, highlighting its effectiveness in sales forecasting. Detecting and adjusting for seasonal trends enhances the model's robustness, which is essential for inventory and demand management. Initial data exploration revealed no significant seasonal variations, as indicated by the near-zero seasonal trend plot, suggesting the absence of strong periodic patterns in the dataset.

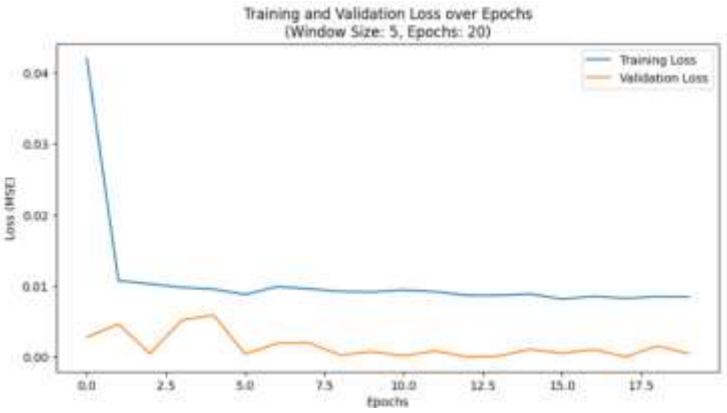

Figure 4.0 Training and Validation Loss (5,20)

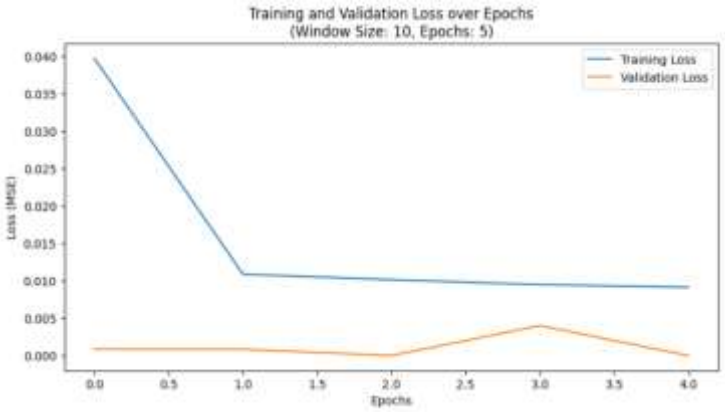

Figure 5.0 Training and Validation Loss (10,5)

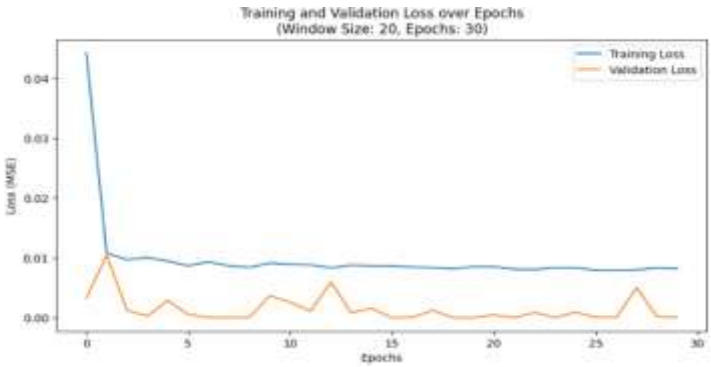

Figure 6.0 Training and Validation Loss (20,30)

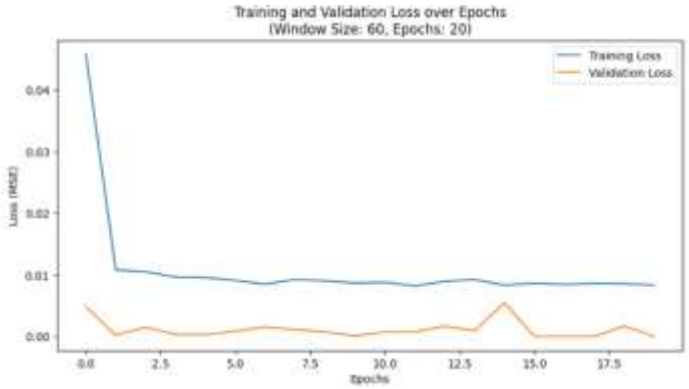

Figure 5.0 Training and Validation Loss (60,20)

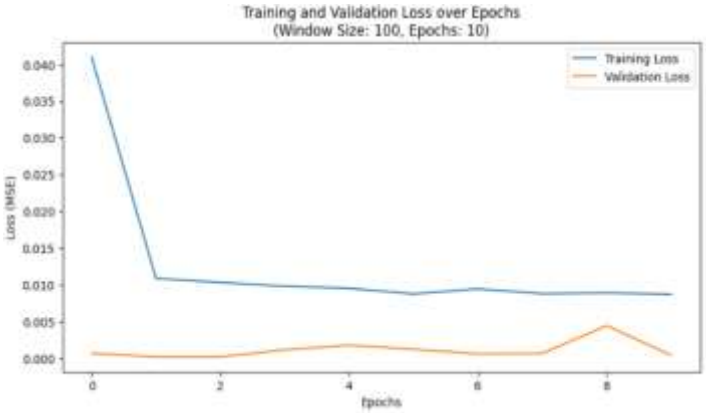

Figure 6.0 Training and Validation Loss (100,10)

Figure 4.0 to 6.0 illustrates the training and validation losses for each window size with chosen epoch which has given the highest R^2.

	Window Size	Epochs	MSE	MAE	R²
0	5	5	0.000006	0.001960	0.963451
1	5	10	0.000310	0.017347	-0.822023
2	5	20	0.000005	0.001686	0.971817
3	5	30	0.000035	0.005546	0.792560
4	10	5	0.000244	0.015418	-0.430336
5	10	10	0.002117	0.045946	-11.433819
6	5	5	0.000505	0.022397	-1.963016
7	5	10	0.006649	0.081494	-38.049755
8	5	20	0.000456	0.021244	-1.675559
9	5	30	0.001362	0.036856	-7.000354
10	10	5	0.000012	0.002927	0.926894
11	10	10	0.000652	0.025487	-2.831118
12	10	20	0.000047	0.006714	0.723459
13	10	30	0.000176	0.013199	-0.035200
14	20	5	0.000402	0.019915	-1.362458
15	20	10	0.000946	0.030688	-4.552783
16	20	20	0.000618	0.024832	-2.629005
17	20	30	0.000064	0.007894	0.625431
18	60	5	0.000401	0.019928	-1.354864
19	60	10	0.000760	0.027465	-3.466220
20	60	20	0.000005	0.002089	0.968373
21	60	30	0.002027	0.044985	-10.902171
22	100	5	0.003870	0.062183	-21.694183
23	100	10	0.000395	0.019765	-1.316376
24	100	20	0.000863	0.029239	-4.059169
25	100	30	0.000874	0.029528	-4.126177

Figure 7.0 Performance metrics for different window sizes with each epochs

Figure 7.0 indicates the performance evaluation for each combination of window size and epochs.

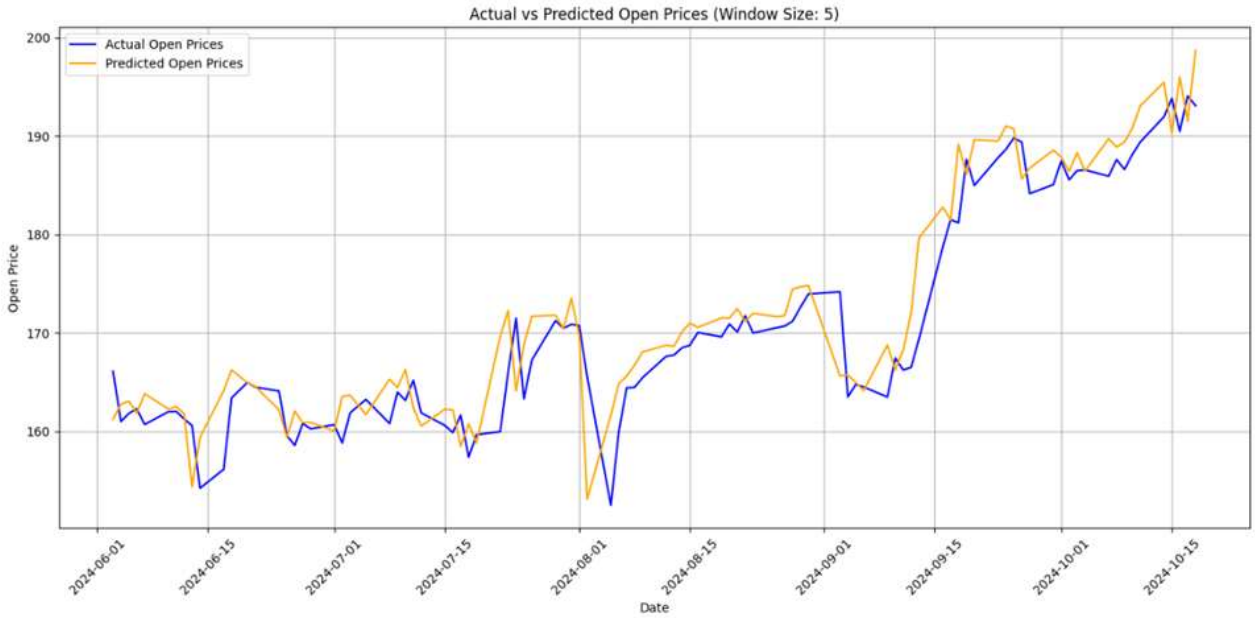

Figure 8.0 Actual vs Predicted of Open Prices (5,20)

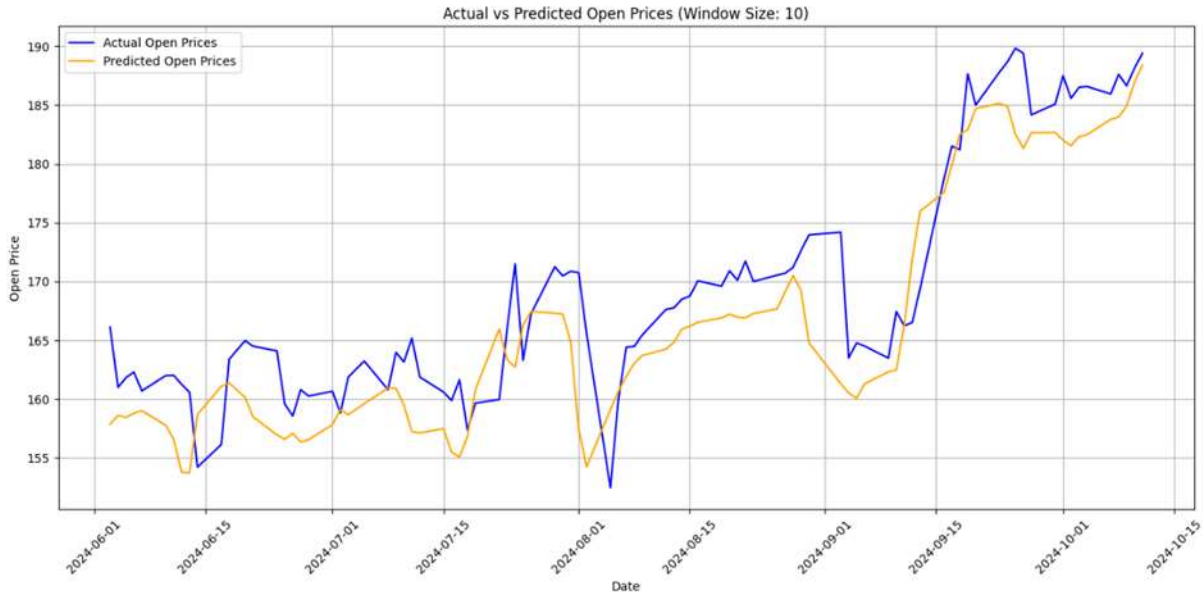

Figure 9.0 Actual vs Predicted of Open Prices (10,5)

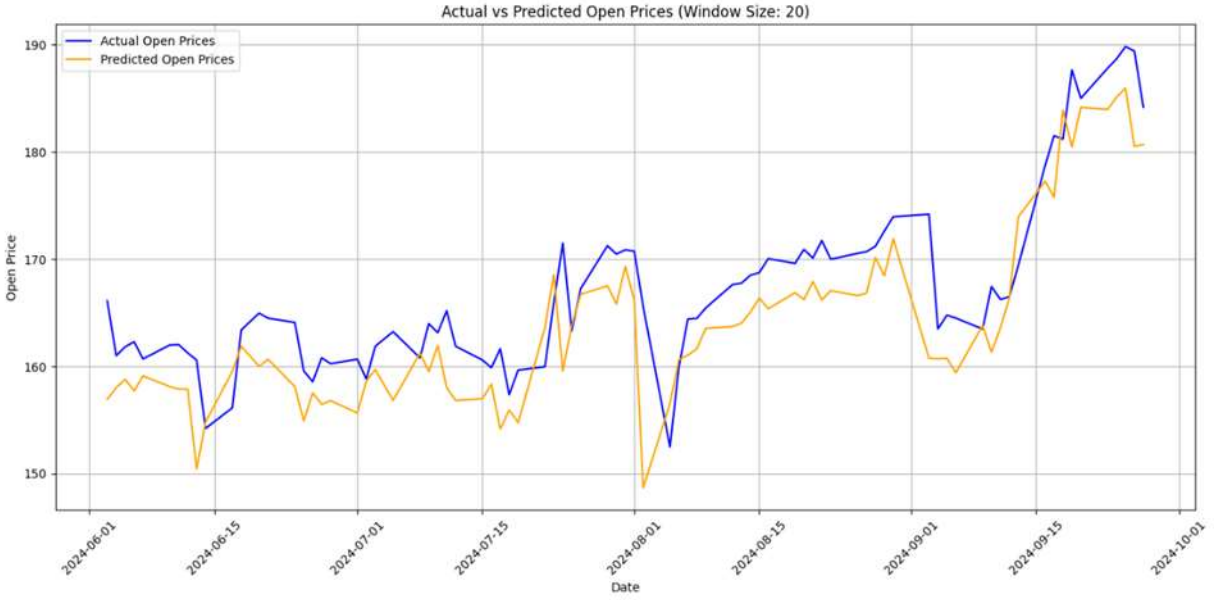

Figure 10.0 Actual vs Predicted of Open Prices (20,30)

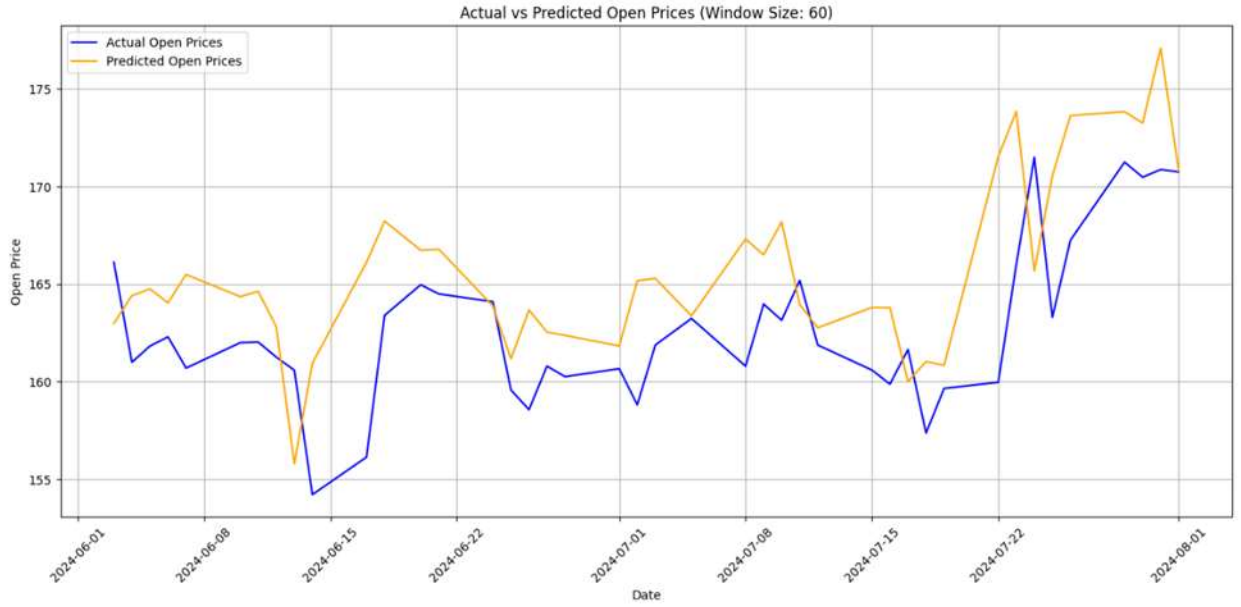

Figure 10.0 Actual vs Predicted of Open Prices (60,20)

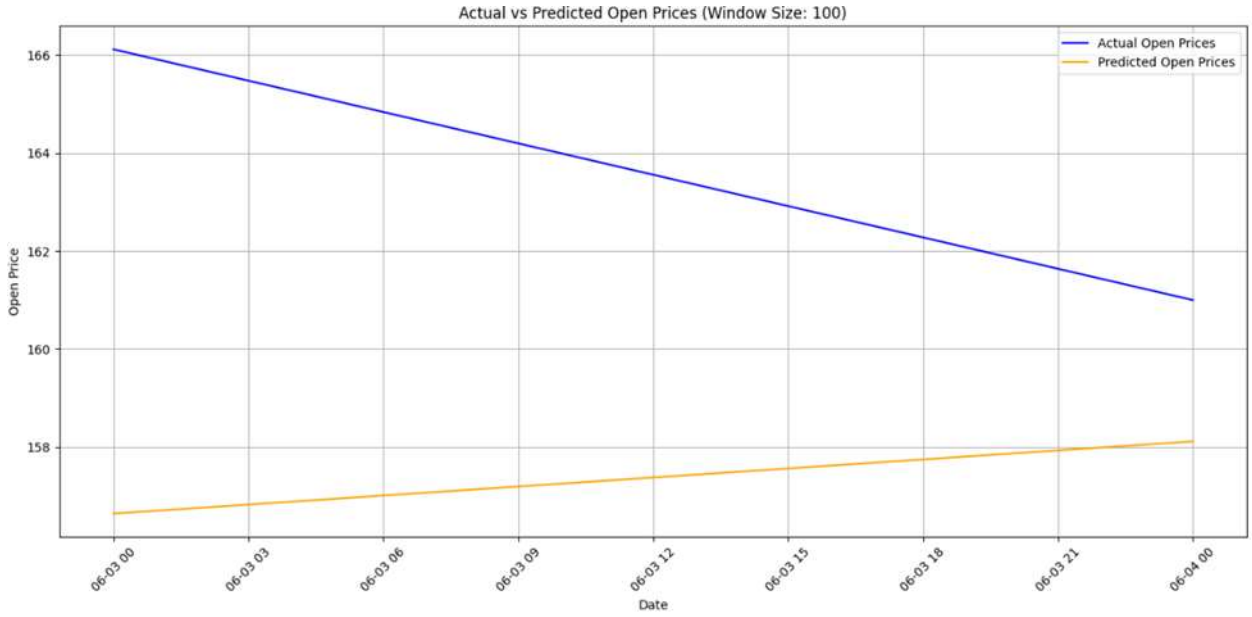

Figure 11.0 Actual vs Predicted of Open Prices (100,10)

Window Size	Epochs	MSE	MAE	R^2
5	20	0.003368	0.057974	-18.303442
10	5	0.0000009	0.002497	0.948379
20	30	0.002445	0.049398	-13.008815
60	20	0.002524	0.050194	-23.272917
100	10	0.00013	0.003043	0.926977

Figure 8.0 to 11.0 illustrates the actual vs prediction variation over time for selected window size with selected epochs. Table 3.0 displays the performance metrics for each desired configuration of window size and epochs.

V. DISCUSSION

configurations of window sizes and epochs in our LSTM model, we analyzed key metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2). These metrics provide insight into the model's accuracy and fit.

Among the configurations tested, the combination of a window size of 10 and 5 epochs yielded the best results, with the following performance metrics:

MSE: 0.0000009, indicating an exceptionally low prediction error.

MAE: 0.002497, which is the lowest among all configurations, suggesting high accuracy in the model's predictions.

R^2: 0.948379, reflecting a strong fit to the data, as values close to 1 indicate that the model explains a substantial portion of the variance.

In contrast, other configurations demonstrated significantly poorer performance. For instance, the configuration with a window size of 5 and 20 epochs resulted in a much higher MSE (0.003368) and a negative R^2 value (-18.303442), indicating a poor model fit. Similarly, the configurations with larger window sizes (20, 60, and 100) also exhibited high MSE values and negative R^2 scores, signifying inadequate model performance.

This analysis underscores the importance of optimizing both window size and epochs in LSTM models for time series forecasting. The selected configuration (window size of 10 and 5 epochs) not only minimizes prediction error but also enhances model reliability, making it the optimal choice for our forecasting project. Future work may further explore the impact of other hyperparameters and feature engineering techniques to enhance model performance.

VI. CONCLUSION

In this study, we investigated the impact of varying window sizes and epochs on the performance of LSTM models for time series forecasting, specifically focusing on sales data. Our analysis revealed that the configuration of a window size of 10 and 5 epochs produced the most favorable results, achieving remarkably low prediction error as indicated by the metrics: Mean Squared Error (MSE) of 0.0000009, Mean Absolute Error (MAE) of 0.002497, and a Coefficient of Determination (R^2) of 0.948379. These metrics suggest that this configuration offers not only high accuracy but also a strong fit to the data, allowing the model to explain a substantial portion of the variance.

Conversely, configurations that utilized smaller or larger window sizes and different epoch settings yielded significantly poorer performance, characterized by higher MSE values and negative R^2 scores. This stark contrast emphasizes the critical importance of optimizing both window size and training epochs in LSTM models for effective time series forecasting.

The findings of this research highlight that the careful selection of hyperparameters can substantially influence the accuracy and reliability of forecasting models. As such, our optimal configuration serves as a valuable reference for practitioners aiming to enhance their forecasting strategies in retail and e-commerce contexts.

Future work should delve deeper into the exploration of other hyperparameters and feature engineering techniques to further improve model performance and robustness. This ongoing research is vital as businesses increasingly rely on accurate forecasting to optimize operations, enhance inventory management, and make informed promotional decisions in an ever-competitive marketplace.

VII. REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[2] P. Bontempi, L. Taieb, and H. Le Borgne, "Machine Learning Strategies for Time Series Forecasting," Statistical Modelling, vol. 13, no. 3, pp. 217-236, 2013.

[3] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.

[4] Y. Zhang, Y. Yu, and Z. H. Zhou, "A Review on LSTM Neural Network and Its Applications in Time Series Forecasting," Journal of Computers, vol. 29, no. 4, pp. 11-21, 2018.