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

Kanchana Weerasinghe
Dalarna University,Sweden
Email: h2kalwe@du.se

Navodya Ranasinghe
Dalarna University,Sweden
Email: v23navra@du.se

***Abstract*— Accurate time series forecasting is crucial for retail and e-commerce businesses, serving as a foundation for effective demand prediction and inventory management. 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) referring to number of past days. 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 5 achieved exceptional results, with an MSE of 1.10809527323373, MAE of 0.789265452, and an R^2 of 0.559834248 using single split technique, 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; Single Split, Hyperparameter Optimization

I. INTRODUCTION

Accurate time series forecasting is crucial in the stock market and the broader financial sector. As the marketplace grows increasingly competitive, businesses and investors depend on precise stock data predictions to make informed decisions, optimize their strategies, 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 window size 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 window sizes 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).

In this context, window size refers to the number of prior days in the data that are used to form the input sequence for the LSTM model. This project aims to systematically explore the impact of varying window sizes on the accuracy and generalization of LSTM models in open price forecasting. By addressing the key research questions of how the choice of window size 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 and optimize their strategies.

Through an experimental approach involving historical stock 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 window size 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 stock price 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 window size 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 window size 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 window sizes, 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 window size 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 window sizes and trainingparameters in LSTM models.

III. METHOD DESCRIPTION

A. The Dataset

The dataset utilized in this study comprises historical stock price data sourced from Yahoo Finance from https://finance.yahoo.com/quote/GE/history/. 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

For the time series analysis Date was used for time series and Open variables was used to predict the future open price.

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.

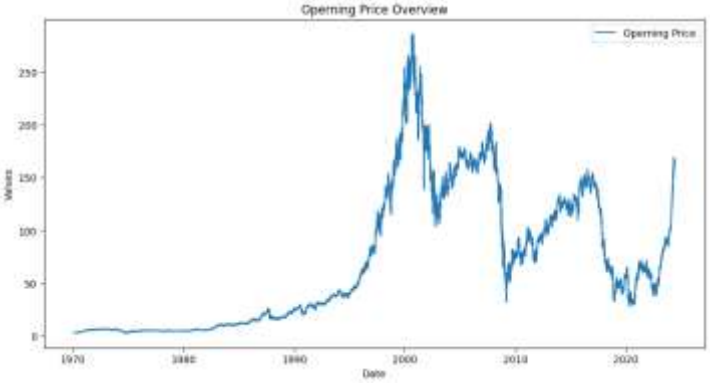

Figure 1.0 Opening Price over Time

.

Figure 1.0 displays the variation of open price from year 1970 to 2024.

To standardize the Open price and improve model performance,the StandardScaler from scikit-learn was utilized to normalizethe training data. It ensures that all Open Price have a similar scale, typically with a mean of 0 and a standard deviation of 1

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. We have fed previous actual dates with price to the model to predict immediate next dates with price.

To optimize the LSTM model's performance, a systematic exploration of window size and the early stop method is used. 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. Early stopping was used to train the model to identify the other best hyperparameters.

During the training phase two cross validation processes were employed. Namely single split and rolling window size. 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 June 12, 2024, immediately following the training period, which covered February 2, 1970, to June 1, 2024, using the make_predictions function.

For each window size, 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 2.0 displays the step-by-step process we have followed in this study.

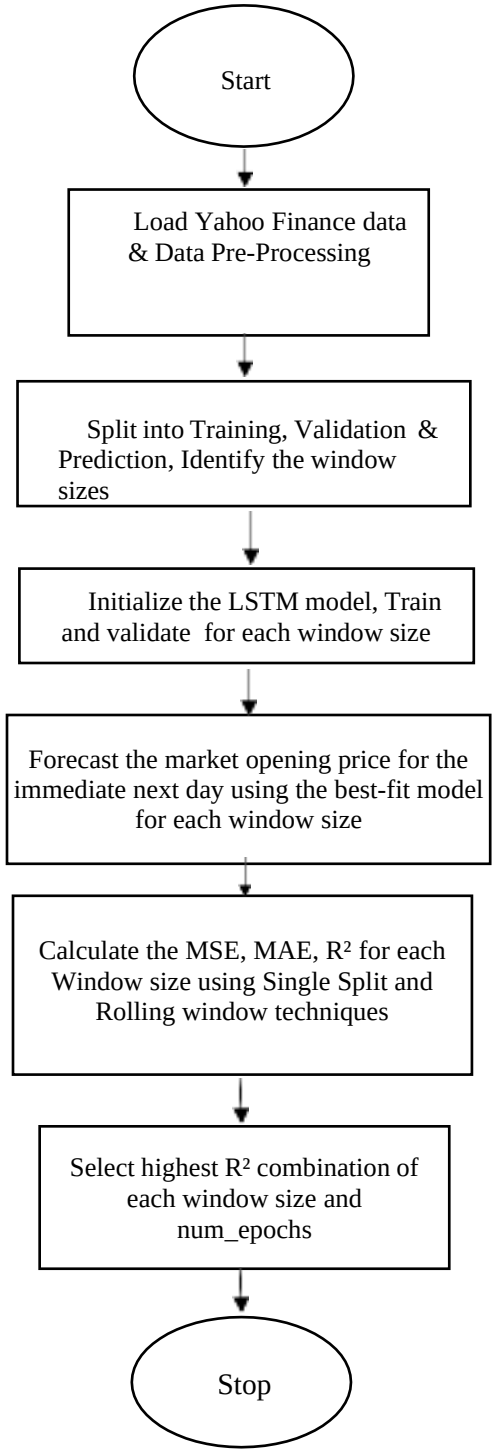

Figure 2.0 Flow diagram of Methodology

IV. RESULTS AND ANALYSIS

The model successfully identified key temporal patterns in the stock data, highlighting its effectiveness in open price forecasting. In the context of LSTM models, there is no need to explicitly address the stationarity or seasonality of the time series, as the model inherently captures temporal dependencies and patterns directly from the data.

Figure 3.0 Actual vs Predicted (Window size-5) – Single Split

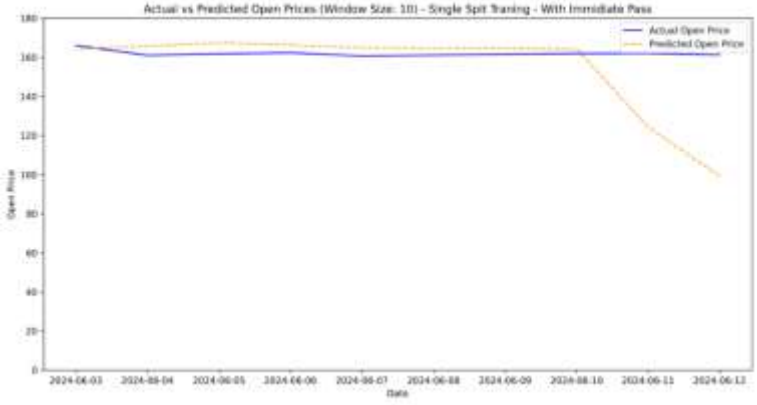

Figure 4.0 Actual vs Predicted (Window size-10) – Single Split

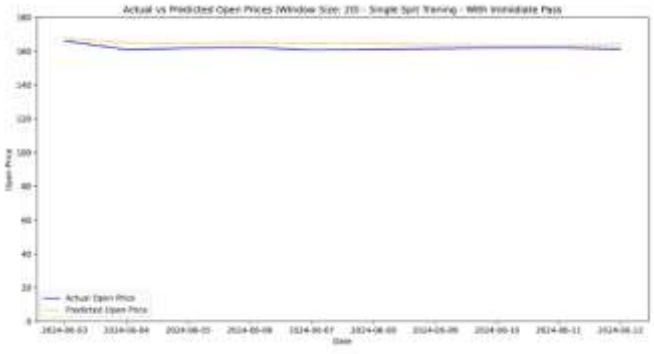

Figure 5.0 Actual vs Predicted (Window size-20) – Single Split

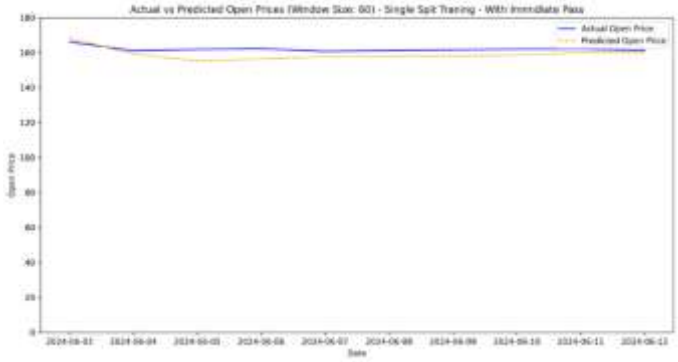

Figure 6.0 Actual vs Predicted (Window size-60) – Single Split

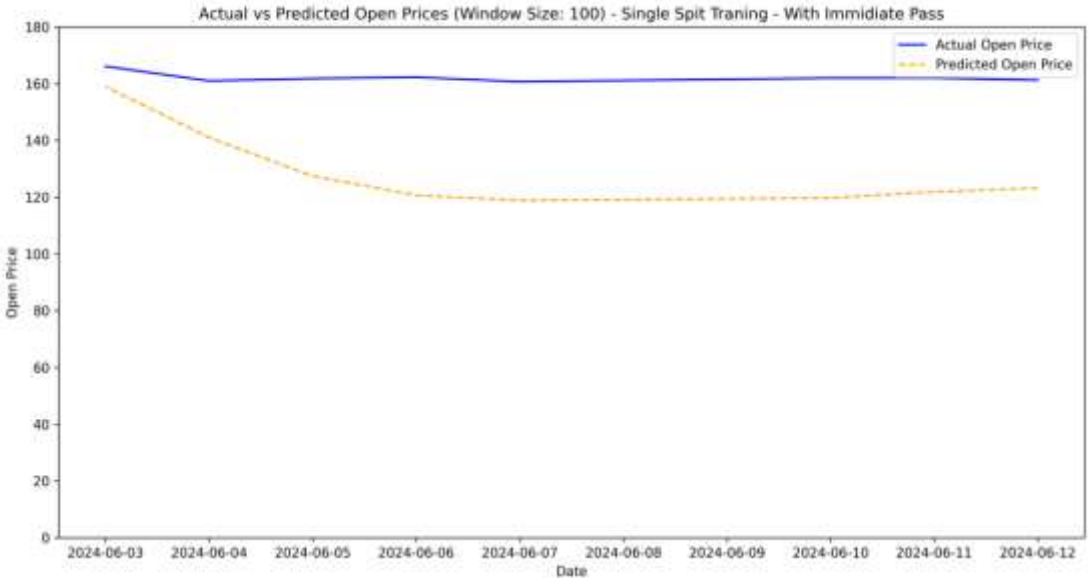

Figure 7.0 Actual vs Predicted (Window size-100) – Single Split

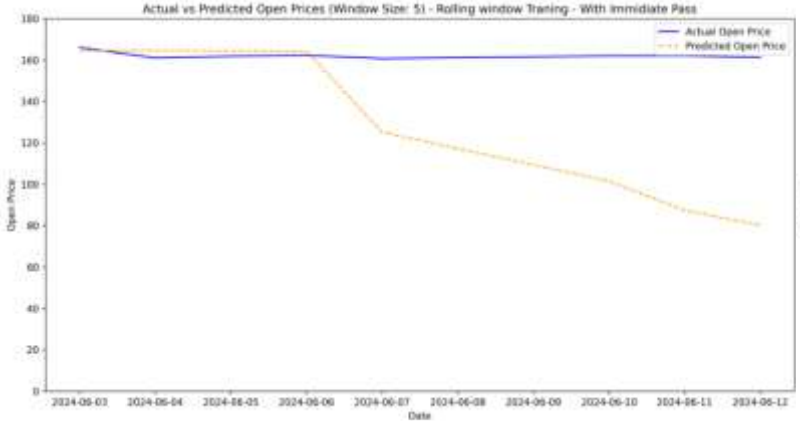

Figure 8.0 Actual vs Predicted (Window size-5) – Rolling Window

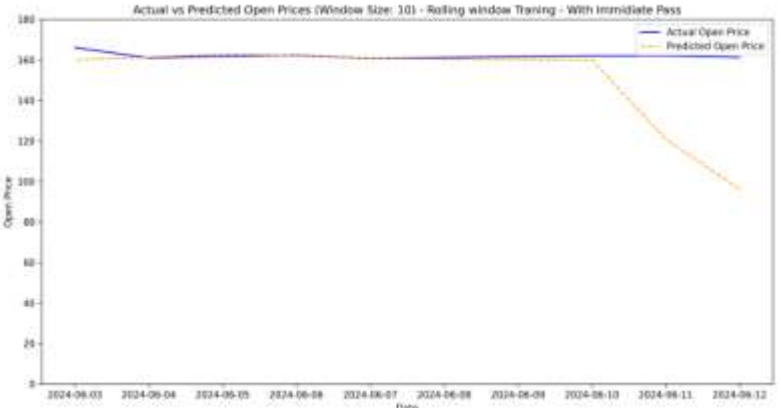

Figure 9.0 Actual vs Predicted (Window size-10) – Rolling Window

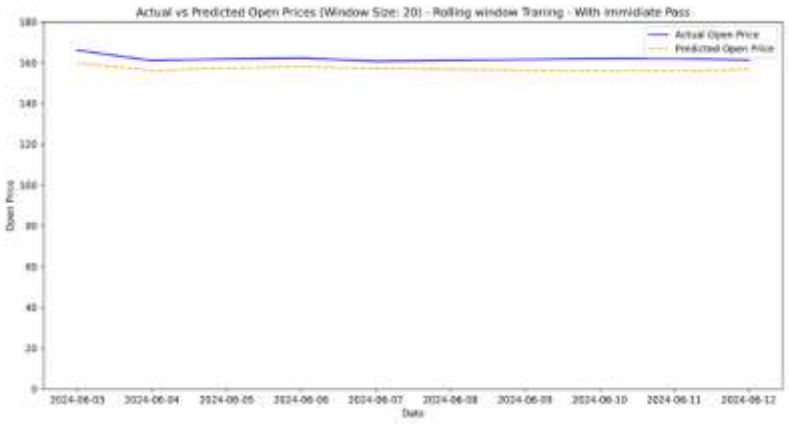

Figure 10.0 Actual vs Predicted (Window size-20) – Rolling Window

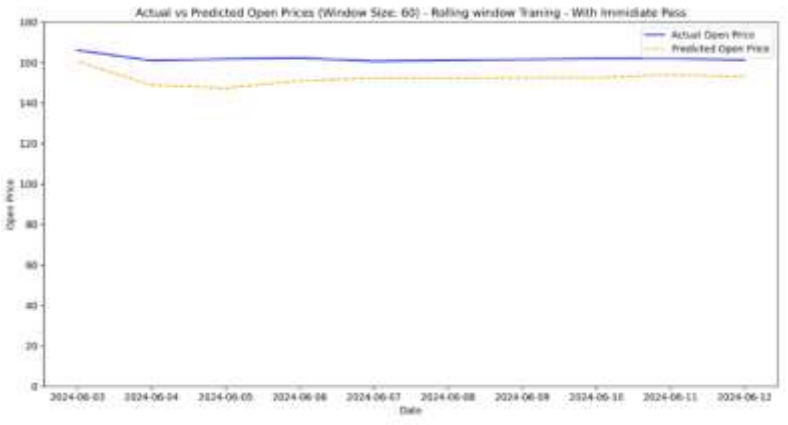

Figure 11.0 Actual vs Predicted (Window size-60) – Rolling Window

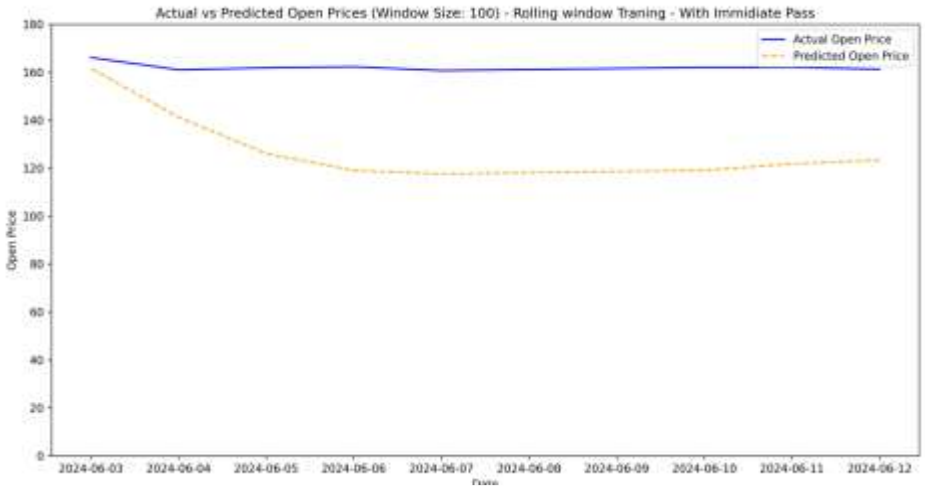

Figure 12.0 Actual vs Predicted (Window size-100) – Rolling Window

Table 2.0 Performance metrices using Single split for different window size

Window Size	Mean Squared Error	Mean Absolute Error	R-Squared
5	2118.9938	31.9132	-840.7223
10	684.1771	15.9087	-270.7738
20	1.1080	0.7892	0.5598
60	72.1273	8.1437	-27.6509
100	1013.0342	29.1886	-401.4049

Table 3.0 Performance metrices using Rolling Window for different window size

Window Size	Mean Squared Error	Mean Absolute Error	R-Squared
5	2138.1015	32.7240	-848.3124
10	745.7326	14.49124	-295.2254
20	25.87911	5.00674	-9.27989
60	99.97660	9.6517	-38.7134
100	1288.72106	33.4141	-510.9152

Figure 3.0 to 12.0 illustrates the actual vs prediction variation over time for selected window sizes. Table 2.0 displays the performance metrics using single split and Table 3.0 displays the performance metrics using rolling window size.

V. DISCUSSION

According to the above results, keeping all other parameters constant except window size, 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. The R^2 found through single split technique were having both positive and negative values In rolling window size technique, the R^2 were all negative values.

A negative R-squared value indicates that, for certain window sizes in this dataset, the model performs worse than simply predicting the mean value across most data points.

For smaller window sizes (5 and 10) , both MSE and MAE are significantly higher, and R^2 values are large negative numbers, indicating poor performance. This is likely because shorter sequences fail to capture the underlying patterns and longer-term dependencies in the data Small windows may lead to overfitting local variations, resulting in reduced generalization to unseen data.

The results for a medium window size (20) are notably better, with a significantly lower MSE and MAE. The R^2 values for single split show a positive value (0.5598), indicating that the model is able to explain a substantial portion of the variance in the data. This suggests that a medium window size effectively balances capturing relevant patterns without overfitting or underfitting.

While larger window sizes (60, 100) theoretically allow for capturing more extensive temporal dependencies, the performance metrics degrade, especially for the largest window size (100). High MSE and MAE, along with negative R^2 values, indicate that overly large windows introduce noise and unnecessary complexity, leading to diminished model performance.

The metrics in single split testing generally show lower error values compared to rolling window, reflecting differences in how the data is partitioned. Single split testing may inadvertently lead to favorable conditions where the train-test distribution aligns more closely. On the other hand, rolling window offer a more robust evaluation by testing across diverse subsets, often revealing limitations in the model's ability to generalize.

VI. CONCLUSION

This project explored the critical role of sequence length (or window size) in determining the accuracy and generalization of LSTM models in time series forecasting using sales data. The findings addressed the research questions and provided actionable insights into designing more effective models for practical applications.

The analysis demonstrated that window size has a profound impact on both model learning and generalization. Small window sizes failed to capture long-term dependencies, resulting in poor generalization and high error metrics. In contrast, overly large windows introduced noise and unnecessary complexity, negatively affecting performance.

Among the tested window sizes, a window size of 20 consistently yielded the best performance, striking a balance between capturing meaningful patterns and avoiding overfitting. This optimal size minimized error metrics, including Mean Squared Error (MSE) and Mean Absolute Error (MAE), while providing the highest R-Squared (R^2) value, indicating better model generalization using single split technique.

The results revealed a clear trade-off: shorter window size resulted in higher MSE and MAE, as the model struggled to learn relevant temporal patterns. Conversely, larger window sizes increased noise and complexity, also leading to higher errors. The intermediate window size of 20 offered the lowest MSE and RMSE, showcasing its ability to capture the temporal dependencies necessary for accurate predictions without overfitting.

The study highlights the importance of systematically evaluating sequence length when designing LSTM-based forecasting models. A well-chosen window size, such as the optimal size identified here, can significantly enhance forecasting accuracy and generalization.

These findings provide a foundation for further exploration, including dynamic or adaptive window size strategies and their potential to improve model performance across varying datasets and domains.

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.