“Machine intelligence is the last invention that humanity will ever need to make."

– Nick Bostrom, philosopher

Author	Student ID	Date	Git Hub Link
Jency Francis	**23082639**	**27th March 2025**	**GitHub Repository**


Comparing Classical and Modern Time Series Forecasting
Methods On the Airline Passenger Dataset

INDEX

1. INTRODUCTION TO TIME SERIES FORECASTING

Time series forecasting is a technique for predicting future data points based on historical patterns. This method is applicable in various fields, including finance, weather forecasting, and inventory management. Among the numerous methods available, ARIMA (AutoRegressive Integrated Moving Average) and LSTMs (Long Short-Term Memory networks) are particularly effective tools for managing time-dependent data.

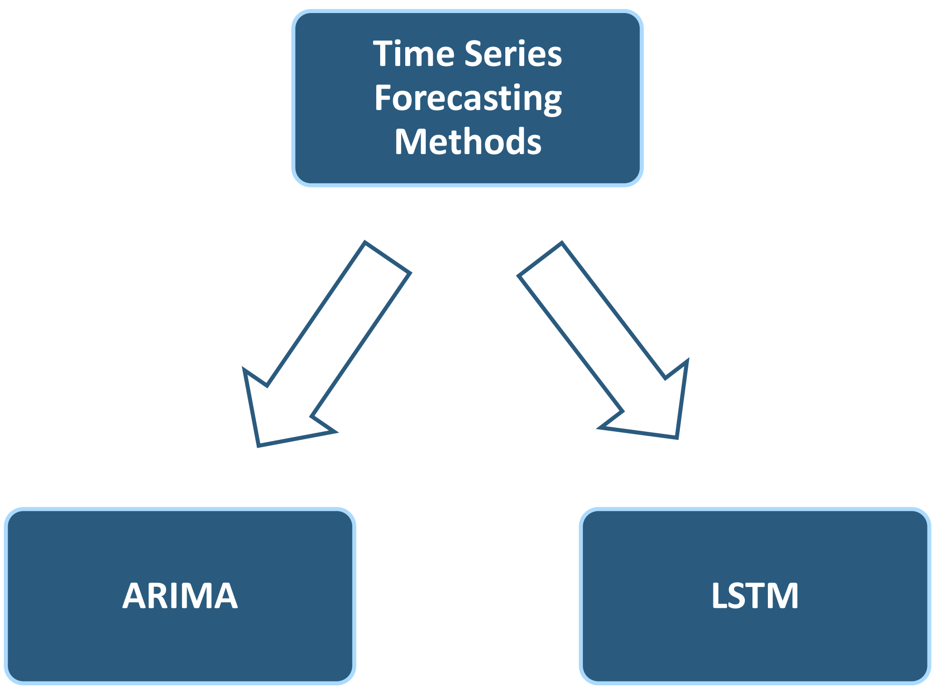


Figure 1. “A diagram depicting Time Series Forecasting Methods: ARIMA and LSTM”

2. WHAT IS ARIMA?

ARIMA (AutoRegressive Integrated Moving Average) stands as a formidable statistical model, celebrated for its expertise in time-series forecasting. This model excels at unearthing trends, seasonality, and autocorrelation within stationary datasets, which is vital for producing precise forecasts. Stationarity refers to a condition in which the mean, variance, and autocorrelation remain constant over time, ensuring the integrity of the predictions of the ARIMA model. When faced with non-stationary data, ARIMA skillfully tackles this challenge through its integrated (I) component, employing differencing techniques to effectively eliminate trends. The legacy of ARIMA was brought to prominence by the brilliant minds of George Box and Gwilym Jenkins, who crafted the renowned Box-Jenkins methodology—a structured approach that laid the groundwork for the widespread acclaim and application of ARIMA.

2.1 ARIMA : Three core components

AutoRegressive (AR)	This element captures the intricate dependencies between an observation and its lagged predecessors, quantified by the parameter **p**
Integrated (I)	Designed to achieve stationarity, this component employs differencing, quantified by **d**
Moving Average (MA)	This aspect addresses the relationships between an observation and the residual errors from previous observations, governed by the parameter **q**

2.2 The Role of ARIMA

ARIMA is known for its remarkable flexibility in modelling linear patterns and handling stationary time-series data. It shines particularly brightly in the domain of short-term forecasting, where trends and seasonal patterns reign supreme. Analysts utilize tools like ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots to expertly uncover the optimal lag order for autoregression and moving averages. When challenged with datasets that unveil seasonal effects, extensions such as SARIMA (Seasonal ARIMA) come into play, enhancing predictive capabilities even further. However, ARIMA encounters challenges when dealing with multivariate time-series data or nonlinear relationships, rendering it less effective for datasets characterized by abrupt changes or complex dependencies. In these scenarios, deep learning models like LSTMs (Long Short-Term Memory networks) often rise to prominence. Nonetheless, ARIMA retains its significance in hybrid approaches, where it cleverly preprocesses linear components while LSTMs capture the nuances of nonlinear patterns. By harmonizing simplicity, interpretability, and predictive power, ARIMA continues to be a foundation of time-series forecasting, leaving a permanent mark on diverse industries such as finance, retail, and energy management. Its enduring appeal lies in its ability to provide insight and clarity in an ever-evolving data landscape.

2.3. Flowchart of ARIMA

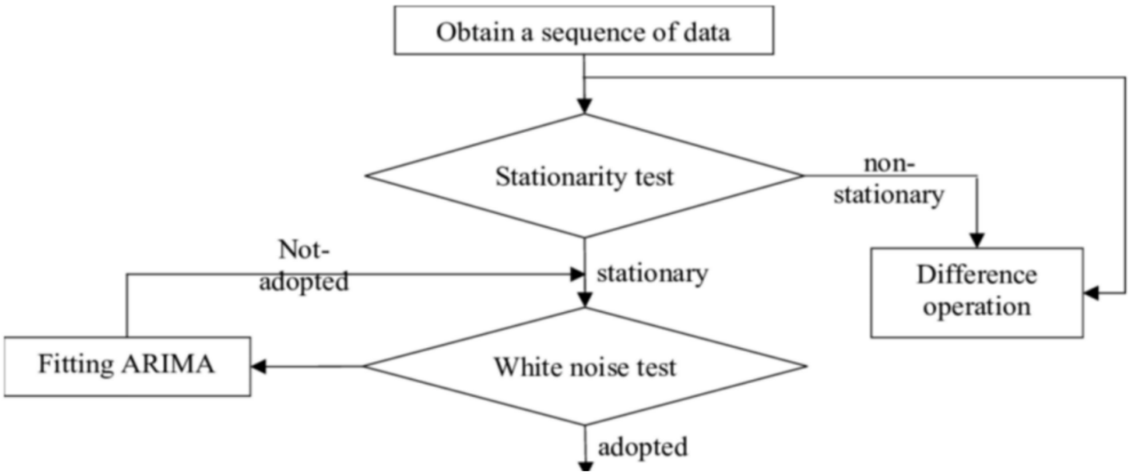


Figure 2. "Obtaining a sequence of data.Perform stationarity test - if data is non-stationary (not adopted), proceed to difference operation to make it stationary.After achieving stationarity (adopted), fit ARIMA model.Conduct white noise test - if passed (adopted), the model is ready for use." Image Reference

3.WHAT IS LSTM?

Long Short-Term Memory (LSTM) networks are a specialized form of Recurrent Neural Networks (RNNs) designed to address the challenges of tailoring long-term dependencies in sequential data. Introduced by Hochreiter and Schmidhuber in 1997, LSTMs are capable of storing and accessing long-term memory, making them invaluable for tasks like time-series forecasting, natural language processing, and speech recognition.

In sequential data, current observations often depend on past values that may be far removed in time. Traditional RNNs struggle to exhibit these dependencies effectively due to the vanishing gradient problem, where gradients diminish during backpropagation, preventing meaningful updates to weights. LSTMs overcome this limitation by incorporating mechanisms that retain relevant information for extended periods.

3.1. Structure of LSTM

Like all RNNs, LSTMs consist of a chain-like structure where data flows through repeating modules. However, unlike standard RNNs, where each module is a simple layer (e.g., tanh), LSTMs use a more sophisticated architecture made up of four interacting components designed to regulate information flow.

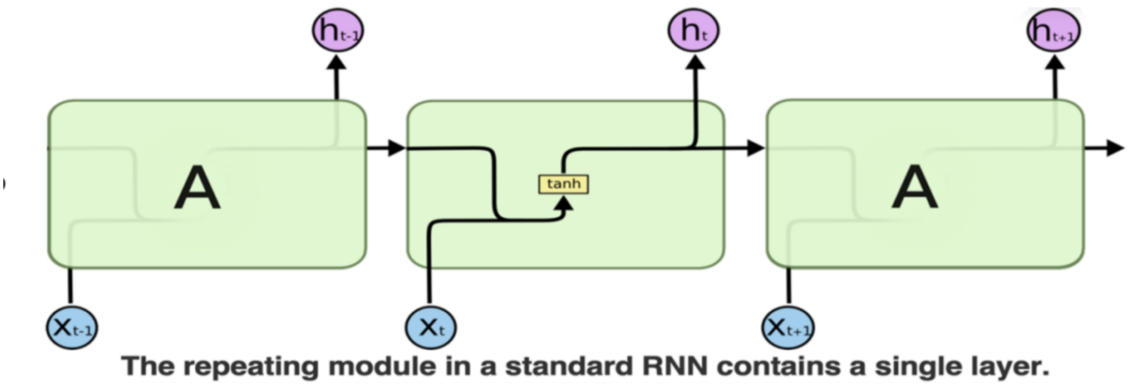


Figure3. "Diagram of a standard RNN (Recurrent Neural Network) module showing: Input (A) flows into the network at time step t (labeled X_t). A tanh activation function processes the input. Output is passed to the next time step t+1 (labeled X_{t+1})". Image Reference

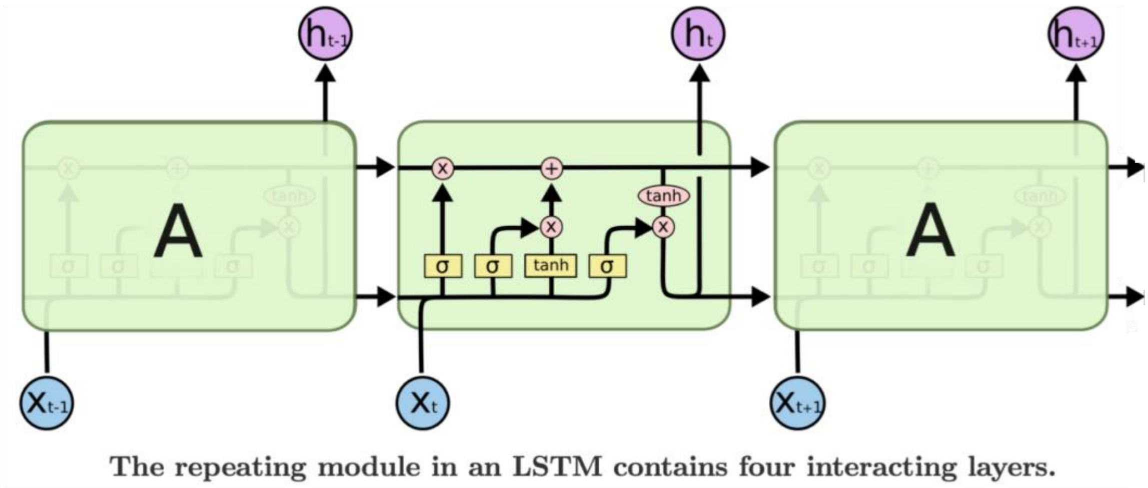
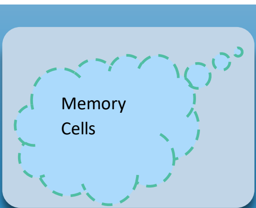


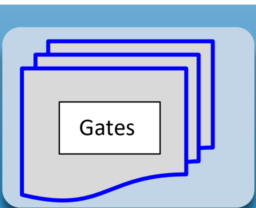
Figure 4. "Diagram of an LSTM (Long Short-Term Memory) module showing:Four vertical time steps labelled t-3 (bottom), t-2, t-1, and current input A (top).Each time step connects to a tanh activation layer (represented by four stacked tanh blocks).Arrows indicate interactions between layers". Image Reference

3.2. Key Features of LSTM



Memory Cells :

Central to LSTMs, these cells maintain information over time, enabling the network to "remember" critical details across long sequences.



Gates :

Gates are mechanisms that control the flow of information. Each gate acts like a "switch" to determind what information is retained, discarded, or passed forward.

3.3. Types of Gates in LSTM

Forget Gate	Input Gate	Output Gate
Determines which information from the previous cell state should be discarded. Outputs values between 0 and 1, where: 0: "Completely forget this information.", 1:"Completely retain this information."	Decides which new information should be added to the memory cell. Combines a sigmoid layer (selecting values to update) and a tanh layer (creating new candidate values for the memory).	Regulates the output from the memory cell and decides what part of the information contributes to the next layer. Uses the cell state and filtered data to determine the final output. These gates interact to ensure that the network retains only the most relevant information at each timestep.

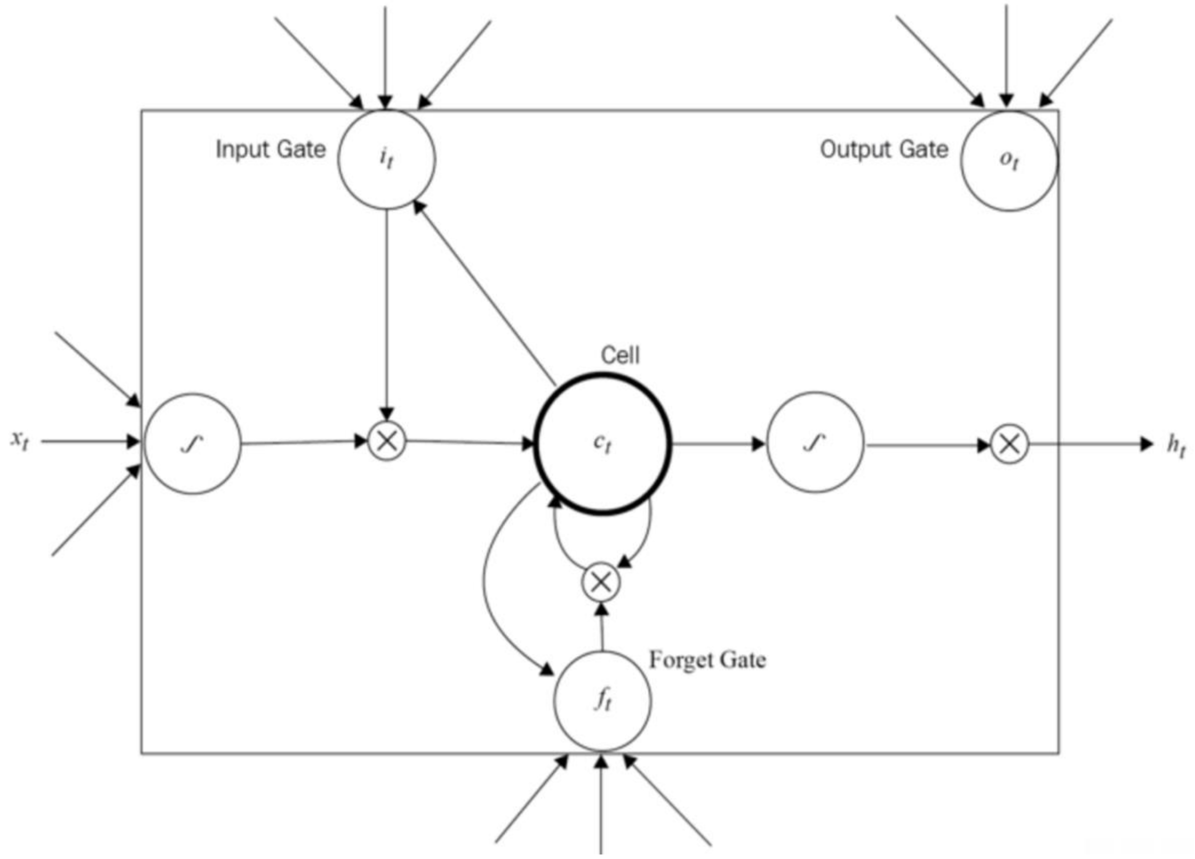


Figure 5. "Diagram of LSTM gates showing: Cell State (center, labeled c_t) as the horizontal memory line.Three gating mechanisms:Input Gate (top): Controls new information flow into the cell.Forget Gate (bottom left, labeled f_t): Decides what to remove from memory. Output Gate right):Determines what to send to the next time step. Hidden state (labelled h_t) as the final output.Arrows illustrate interactions between gates and the cell state." Image Reference

4. ARIMA VS LSTM

The following table illustrates the key features of ARIMA and LSTM.

Feature	ARIMA	LSTM
Data Type	Univariate	Univariate/Multivariate
Model Type	Statistical	Neural Network
Data Pattern	Linear	Nonlinear
Data Size	Small/Medium	Large
Stationarity Requirement	Required	Not Required
Training Time	Fast	Slow
Computational Cost	Low	High
Interpretability	High	Medium

5. APPLICATION OF ARIMA AND LSTM ON THE AIRLINE PASSENGERS DATASET

We selected the Airline Passenger Forecasting Dataset from Kaggle, which contains monthly international airline passenger numbers from 1949 to 1960. This dataset was ideal for our analysis due to its clear trends and seasonality, making it suitable for time-series forecasting. Using this data, we applied both ARIMA and LSTM models to predict future passenger numbers. ARIMA was employed to model linear trends and seasonal patterns, providing interpretable short-term predictions. In contrast, LSTM, a deep learning model, was used to capture nonlinear patterns and long-term dependencies within the data. After implementing both models, we evaluated their performance using metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). This evaluation allowed us to gain insights into the strengths and limitations of each model for forecasting tasks.

5.1. Airline Passengers Over Time

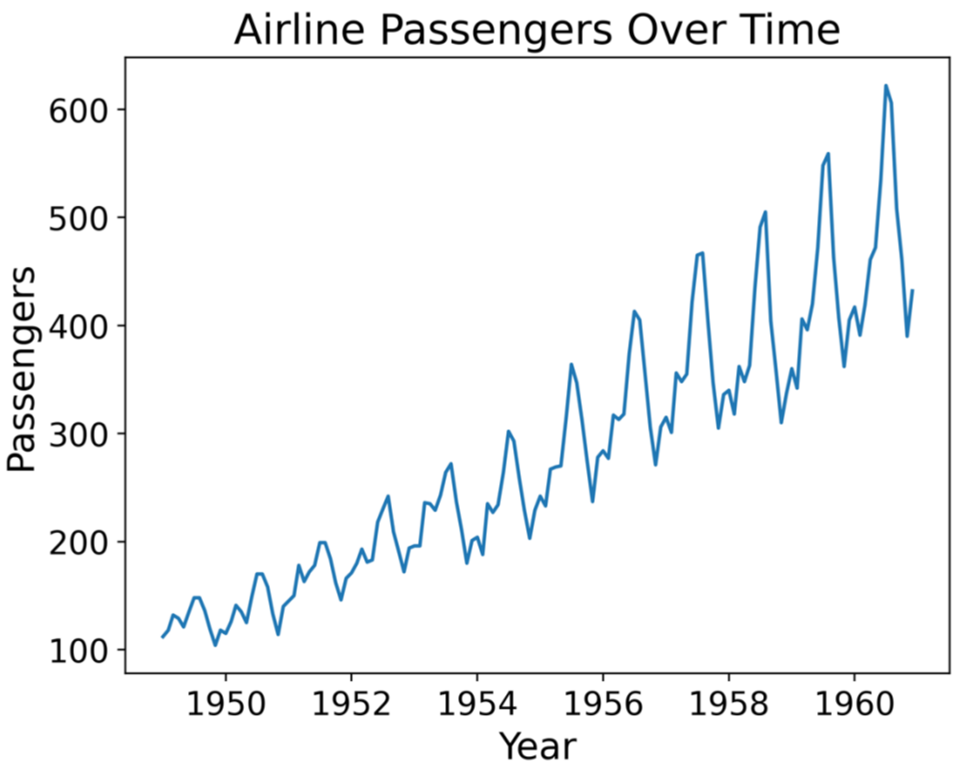


Figure 6. “The Airline Passengers Over Time plot illustrates a clear upward trend in passenger numbers, reflecting sustained growth likely driven by increasing global travel demand. Seasonal patterns are prominent, with recurring peaks during high travel seasons and dips in off-peak times, showcasing predictable cycles in travel behavior. This visualization effectively highlights both long-term trends and short-term fluctuations, offering valuable insights for strategic planning and operational optimization.”

5.2. ARIMA forecast of Airline Passengers Over Time

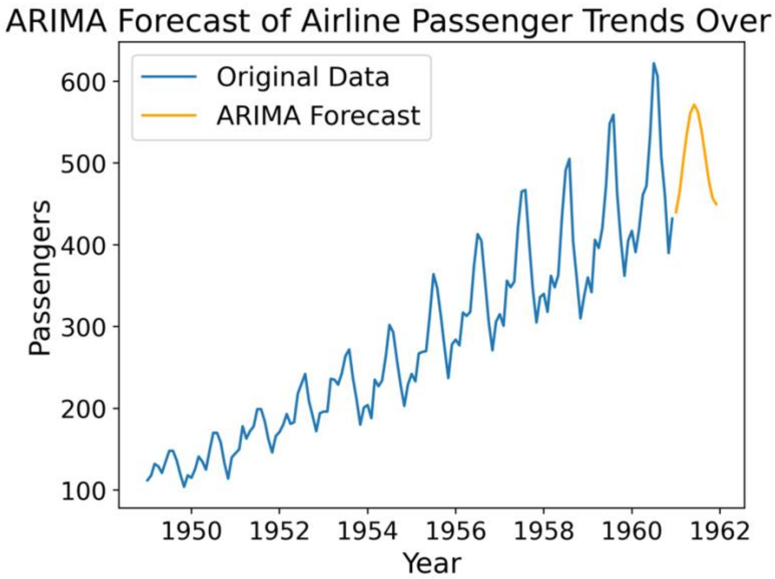


Figure 7. “ Data on airline passengers over time shows a consistent upward trend, indicating an increase in travel demand and global connectivity. Seasonal fluctuations are observed regularly, with peaks during high travel seasons and troughs during off-peak periods, reflecting predictable demand cycles. This pattern emphasizes the importance of recognizing both long-term growth and seasonal variations, providing valuable insights for optimizing airline operations, capacity planning, and informed decision-making.”

5.3. ARIMA Parameterization

ARIMA parameterization is essential because it dictates how effectively the model represents the underlying patterns in time series data, which in turn impacts the accuracy of forecasts. The three key parameters—p (autoregressive order), d (differencing), and q (moving average order)—have distinct roles and must be carefully adjusted to align with the characteristics of the dataset.

5.3.1. Samples to support the importance of ARIMA Parameterization

Simpler Models (e.g., ARIMA(1, 0, 0))

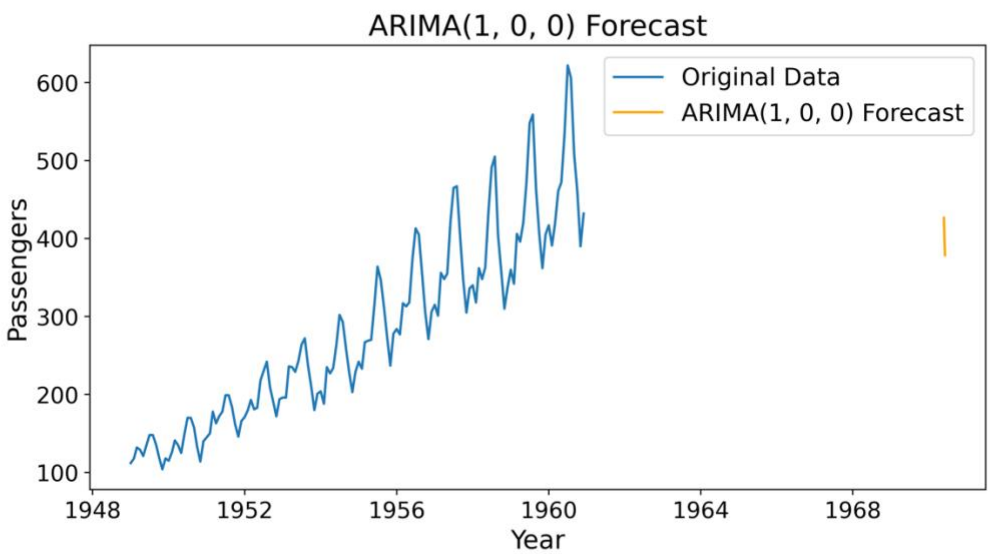


Figure 8. “This text emphasizes how simple configurations are effective for basic datasets.”

Moderately Complex Models (e.g., ARIMA(0, 1, 1))

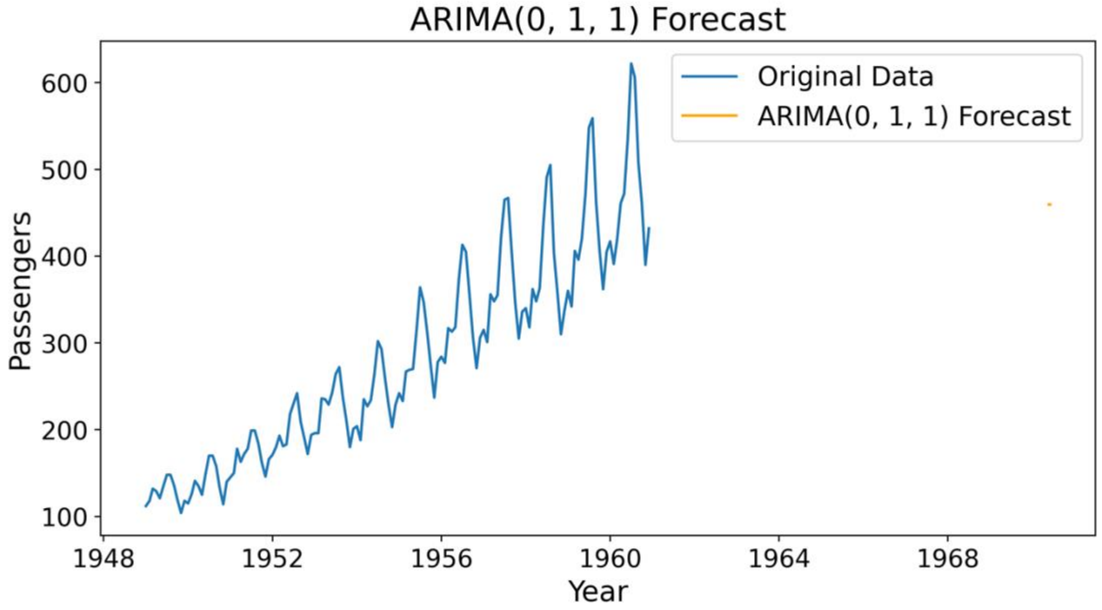


Figure 9. "Demonstrates how differencing and smoothing techniques address non-stationarity and manage moderate complexity."

Advanced Models (e.g., ARIMA(2, 1, 2))

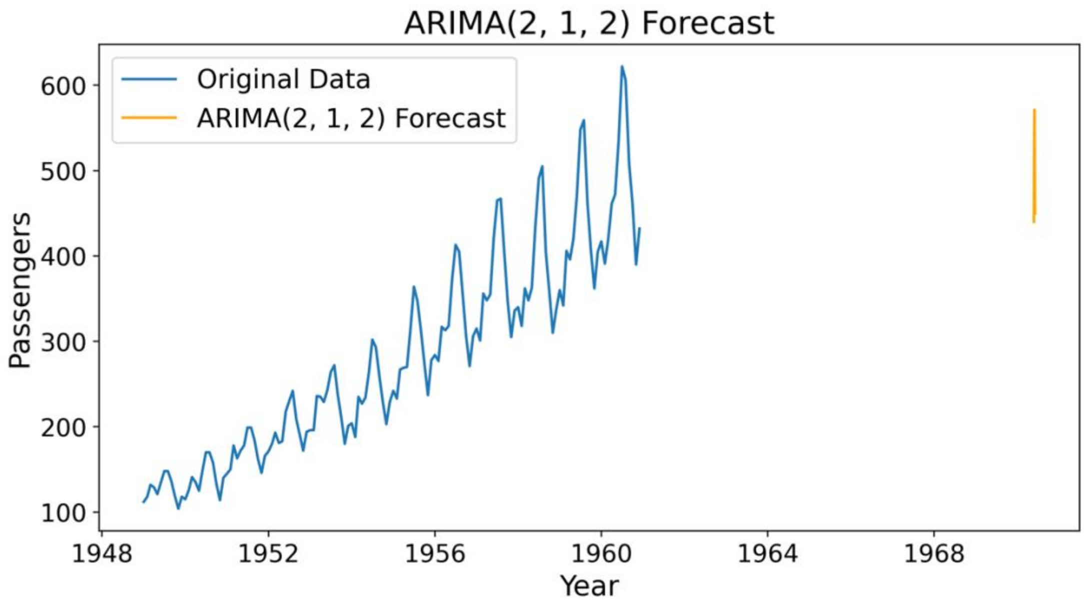


Figure 10. "Adding more parameters enhances the ability to capture complex patterns and dependencies."

Over-parameterized Models (optional, e.g., ARIMA(2, 2, 2))

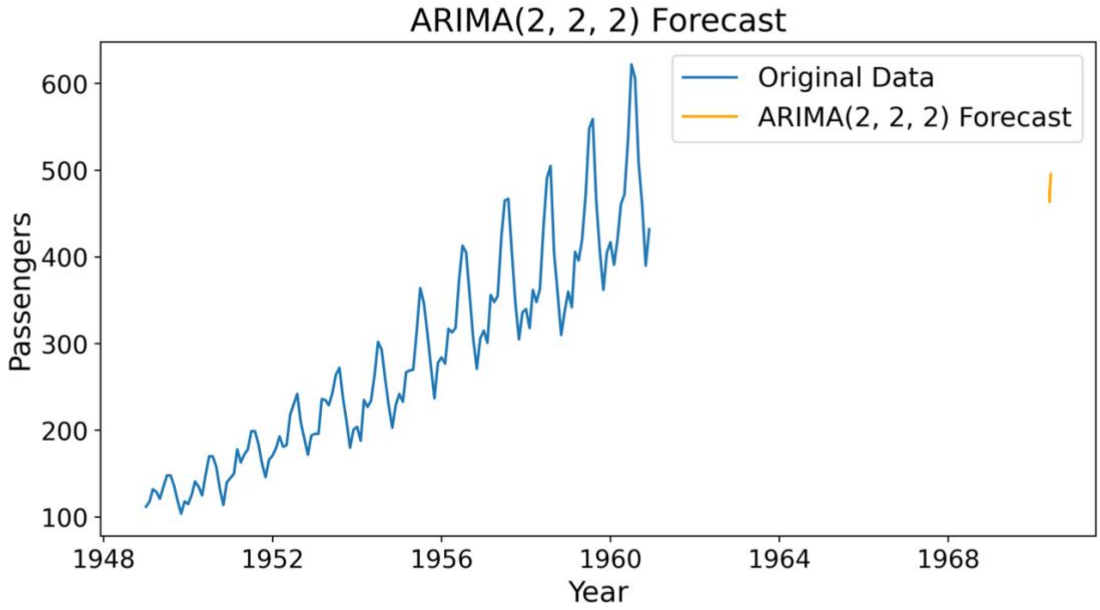


Figure 11. “The ARIMA(2, 2, 2) model exhibits complexity, capturing intricate dependencies and trends while addressing non-stationarity, although it may risk overfitting. ”

Edge Cases (optional, e.g., ARIMA(0, 0, 0))

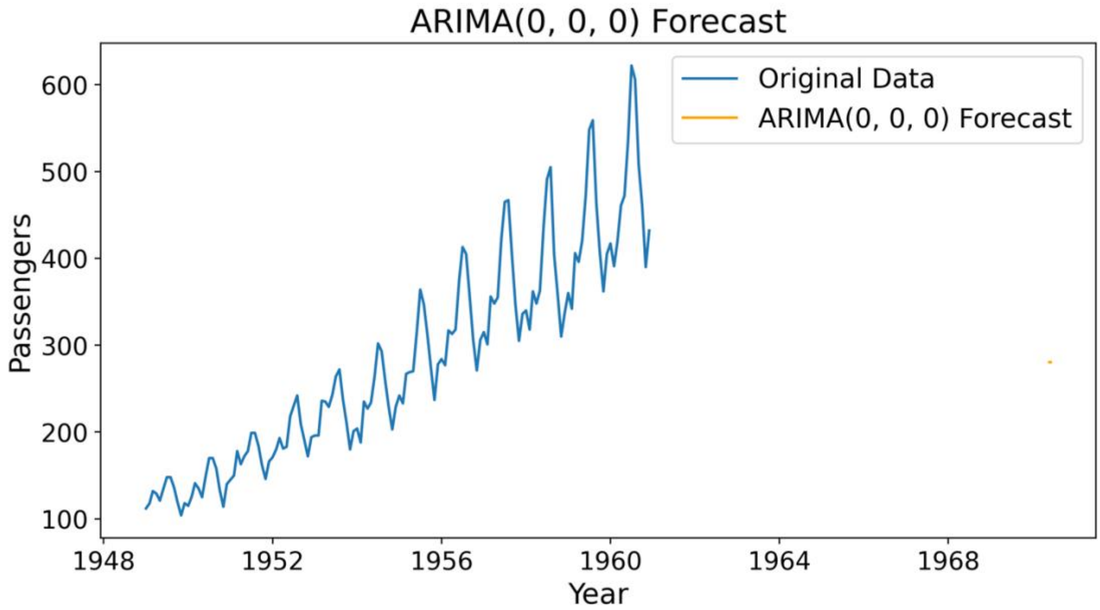


Figure 12. “This serves as a baseline, illustrating how limited parameters can fail to capture trends or seasonal variations.”

5.4. LSTM forecast of Airline Passengers Over Time

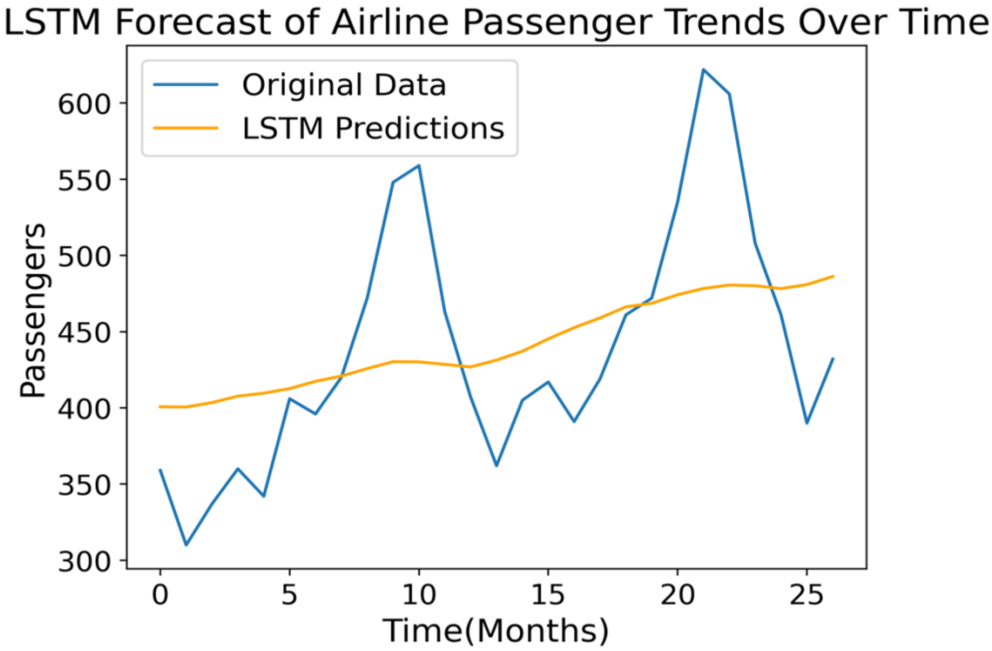


Figure 13. “The LSTM forecast, developed with varying layers and neurons, effectively captures the steady upward trend in airline passenger numbers, which is indicative of sustained growth driven by increasing travel demand. Seasonal patterns are apparent, featuring regular peaks and troughs that correspond to high and low travel seasons. These insights highlight how LSTM models can effectively reveal recurring trends, assisting airlines in optimizing their scheduling and resource allocation to meet predictable seasonal demand.”

5.4.1. Analyzing LSTM Configurations: The Impact of Layers and Neurons (Width) on Forecast Accuracy.

The number of layers and neurons in a neural network plays a significant role in determining forecast accuracy. Adding more layers allows the model to better capture complex patterns and relationships in the data. However, if the network becomes too deep, it may overfit the training data, which can harm its ability to generalize to new, unseen data. Similarly, increasing the number of neurons in each layer enhances the capacity of the model to learn complex features. Besides, having too many neurons can lead to redundancy and computational inefficiency. Hence, finding the right balance between depth and width is essential for achieving optimal performance based on the complexity of the dataset.

5.4.2. Samples to support the Impact of Layers and Neurons(Width)

LSTM (1 Layer, 20 Neurons)

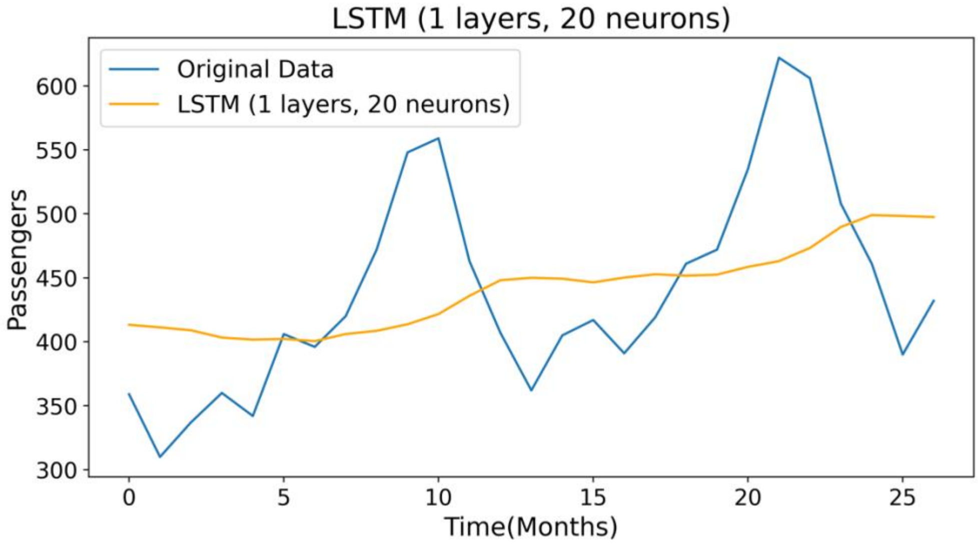


Figure 14. “Shows how low complexity can lead to underfitting, making it difficult for the model to identify patterns in the data. This is helpful in illustrating the limitations of single-layer models with minimal width”

LSTM (1 Layer, 50 Neurons)

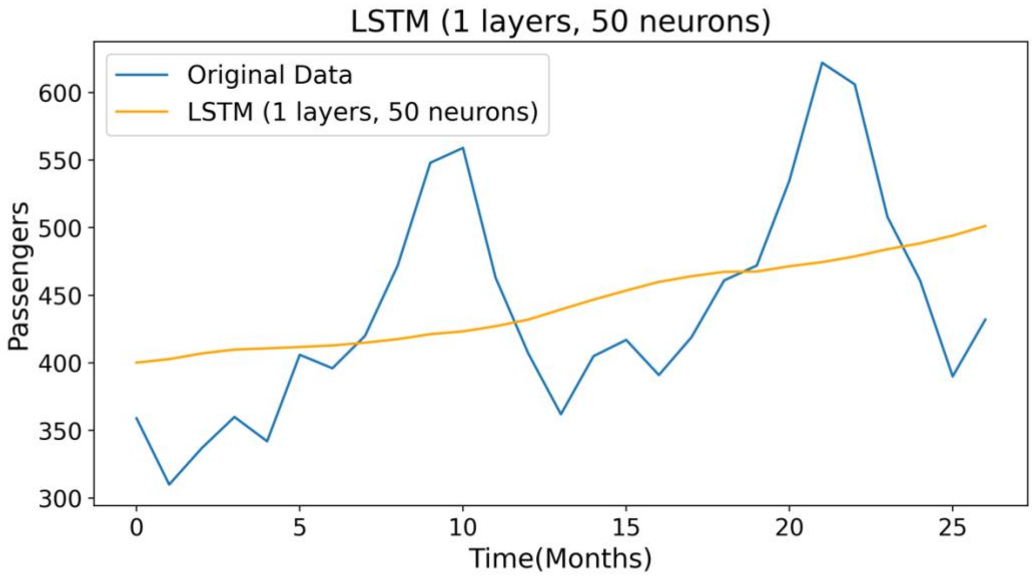


Figure 15. “Emphasizes how a moderate number of neurons balances simplicity and performance for basic patterns in the data. It provides an example of suitable configurations for datasets with low to moderate complexity.”

LSTM (1 Layer, 100 Neurons)

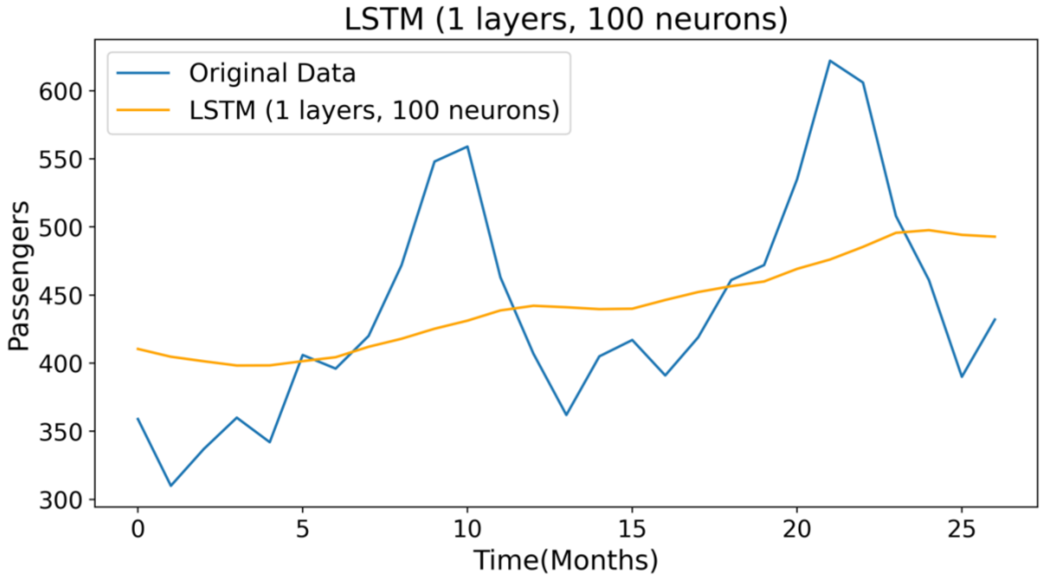


Figure 16: “Demonstrates the potential risks of overfitting from having too many neurons in a single-layer model. It effectively illustrates how increased width affects highly complex datasets.”

LSTM (2 Layers, 50 Neurons)

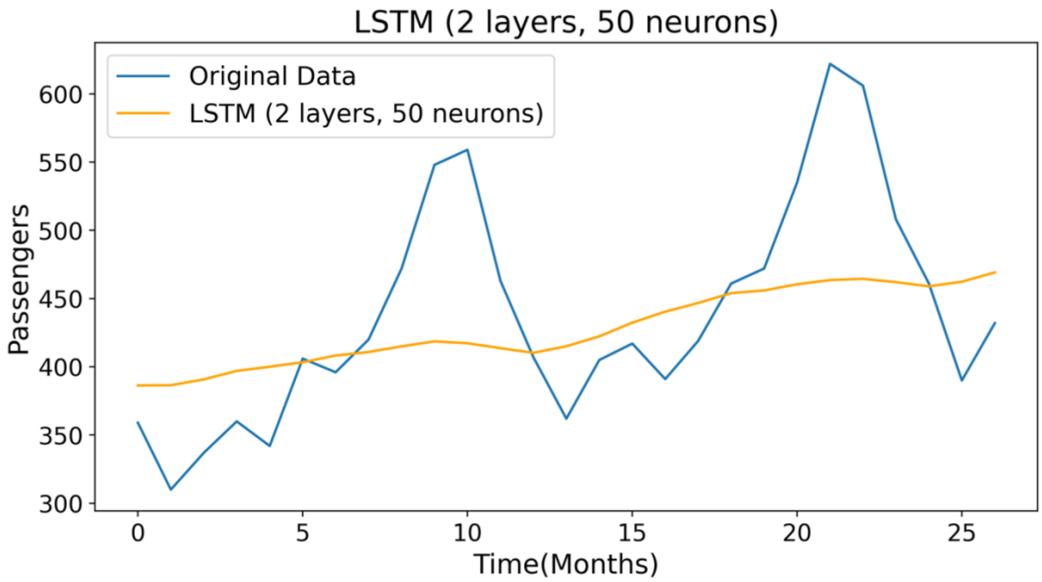


Figure 17. “Displays balanced configurations with enhanced depth, improving the handling of moderately complex data. It is a strong contender for optimal forecasting performance across diverse datasets.”

LSTM (3 Layers, 20 Neurons)

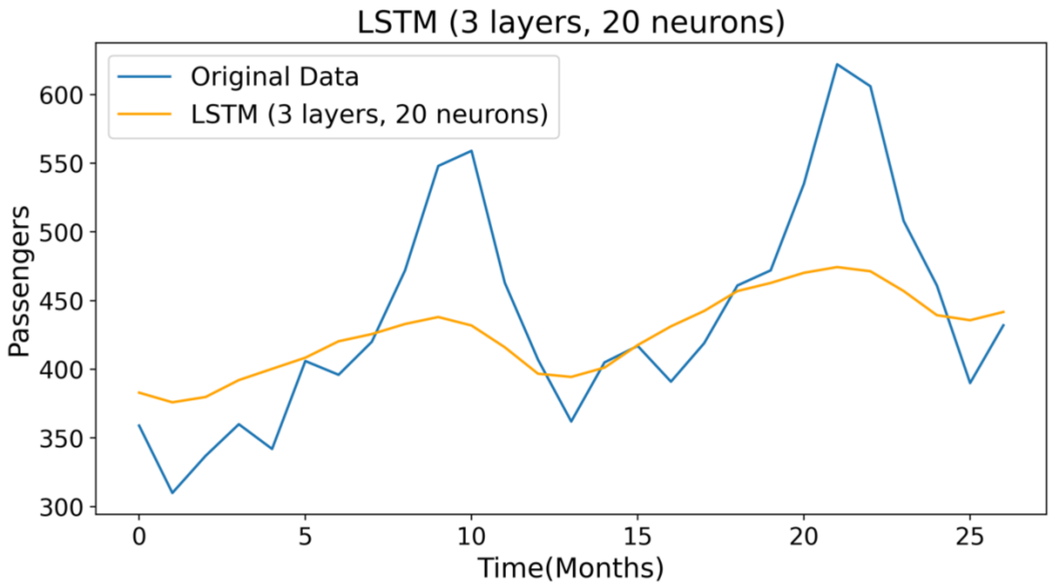


Figure 18. “Increasing layers without adequate width can lead to unnecessary complexity and the risk of overfitting. It is crucial to align the depth and width of a model with the characteristics of the dataset.”

LSTM (3 Layers, 100 Neurons)

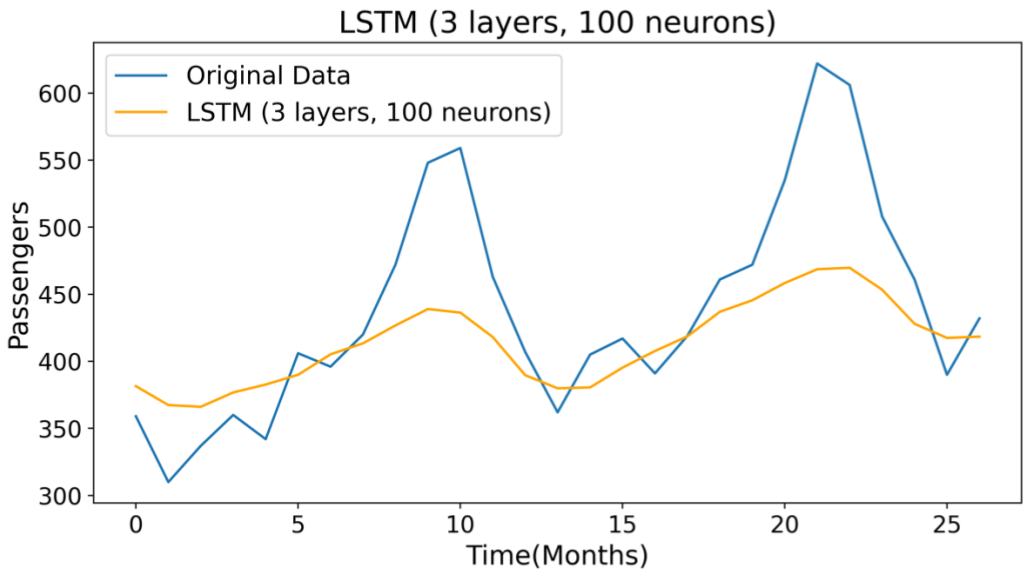


Figure 19. “Addresses the risk of overfitting that can arise when both depth and width are excessively large. It examines cases where advanced configurations are advantageous only for large and highly complex datasets.”

5.5. Result Comparison

Result Comparison

```
from sklearn.metrics import mean_absolute_error, mean_squared_error

# ARIMA evaluation
mae_arima = mean_absolute_error(data.values[-12:], forecast_arima)
rmse_arima = np.sqrt(mean_squared_error(data.values[-12:], forecast_arima))

# LSTM evaluation
mae_lstm = mean_absolute_error(data.values[-len(y_test):], predictions)
rmse_lstm = np.sqrt(mean_squared_error(data.values[-len(y_test):], predictions))

print(f"ARIMA - MAE: {mae_arima}, RMSE: {rmse_arima}")
print(f"LSTM - MAE: {mae_lstm}, RMSE: {rmse_lstm}")
```

```
ARIMA - MAE: 50.60082608413314, RMSE: 58.19079756589749
LSTM - MAE: 53.71385588469329, RMSE: 66.74946298359886
```

The comparison between ARIMA and LSTM models shows that ARIMA has slightly better accuracy, with a Mean Absolute Error (MAE) of 50.60 and a Root Mean Squared Error (RMSE) of 58.19. This indicates that ARIMA aligns consistently with the dataset. In contrast, while LSTM exhibits slightly higher error metrics—MAE of 53.71 and RMSE of 66.75—it has the potential to capture complex patterns, suggesting that it could improve with further optimization. Overall, this analysis highlights ARIMA's reliability for identifying structured trends, while LSTM offers greater flexibility for capturing complex dependencies.

6.CONCLUSION

The parameterization of ARIMA is crucial for effectively modelling time series data, as its key parameters (p,d,q) must be carefully adjusted to fit the characteristics of the dataset. In LSTM neural networks, the number of layers and neurons(width) play a pivotal role in determining forecast accuracy. While more layers help capture intricate patterns, too much depth can lead to overfitting, and an excessive number of neurons can create redundancy and inefficiencies. ARIMA is ideal for capturing simpler, linear relationships, whereas LSTM is well-suited for recognizing complex, nonlinear dependencies. A hybrid approach that combines ARIMA for linear trends and LSTM for residuals can provide a more balanced and accurate forecasting model.

7.REFERENCES

1. Box, G.E.P., Jenkins, G.M., & Reinsel, G.C. (2015), Time Series Analysis: Forecasting and Control, Wiley.

2. R. J. Hyndman, G. Athanasopoulos, 2014, Forecasting: Principles and Practice, OTexts.

3. Saadeddin,Zaina, Last updated on Jan 7,2025, Last visited on Mar 24,2025, datacamp, ARIMA for Time Series Forecasting: A Complete Guide, https://www.datacamp.com/tutorial/arima

4. Samad,Mirza,Dec 5, 2024, Last visited on Mar 24,2025,Published in Major Digest,Medium, When to Use ARIMA, LSTM, and Gated Neural Networks in Time Series Analysis, https://medium.com/majordigest/when-to-use-arima-lstm-and-gated-neural-networks-in-time-series-analysis-8bdd9db2d60c

5.Albeladi, Khulood&Zafar,Bassam&Mueen,Ahmed, Vol. 14, No. 1, 2023, DOI: 10.14569 /IJACSA.2023.0140133,IJACSA, The Science And Information (SAI) Organization, Time Series Forecasting using LSTM and ARIMA, https://thesai.org/Downloads/Volume14No1/Paper_33-Time_Series_Forecasting_using_LSTM_and_ARIMA.pdf

6. Taslim ,Deddy Gunawan &Murwantara, I Made , Vol. 13, No. 3, June 2024, DOI: 10.11591/eei.v13i3.6034, Bulletin of Electrical Engineering and Informatics, ResearchGate, Comparative analysis of ARIMA and LSTM for predicting fluctuating time series data, https://www.researchgate.net/publication/381072094_Comparative_analysis_of_ARIMA_and_LSTM_for_predicting_fluctuating_time_series_data

7. Siami-Namini,Sima& Tavakoli, Neda& Namin, Akbar Siami,2018, 17th IEEE International Conference on Machine Learning and Applications, National Science Foundation(.gov), A Comparison of ARIMA and LSTM in Forecasting Time Series, https://par.nsf.gov/servlets/purl/10186768

8. Ma,Qihang,2020,E3S Web of Conferences 218, 01026(2020), DOI: 10.1051/e3sconf/202021801026 ResearchGate,Comparison of ARIMA,ANN, and LSTM for Stock Price Prediction, https://www.researchgate.net/publication/347611169_Comparison_of_ARIMA_ANN_and_LSTM_for_Stock_Price_Prediction/link/609579fea6fdccaebd15c946/download/

9. Zhang,Zizhe, Available online 10 October 2023, DOI: 10.2991/978-94-6463-268-2_6, Proceedings of the 2023 2nd International Conference on Economics, Smart Finance and Contemporary Trade(ESFCT 2023), Atlantis Press, Comparison of LSTM and ARIMA in Price Forecasting: Evidence from Five Indexes, https://www.atlantis-press.com/proceedings/esfct-23/125992891

10. Dabakoglu,Caner, Jun23,2019, Last visited on Mar 24,2025,Medium, Time Series Forecasting — ARIMA, LSTM, Prophet with Python, https://medium.com/@cdabakoglu/time-series-forecasting-arima-lstm-prophet-with-python-e73a750a9887

11. Olah,Christopher,August 27,2015, Last visited on Mar 24,2025,colah's blog, Understanding LSTM Networks, https://colah.github.io/posts/2015-08-Understanding-LSTMs/

12. Upadhyay, Vivek Kumar, Feb 19,2024, Last visited on Mar 24,2025, Medium, ARIMA vs LSTM: A Comparative Study of Time Series Prediction Models, https://vivekupadhyay1.medium.com/arima-vs-lstm-a-comparative-study-of-time-series-prediction-models-91fa4219d9d9

13. Uslu,Ozlem, 2021(Updated 11 days ago), Last visited on Mar 26,2025,kaggle, Airline Passenger Forecasting, https://www.kaggle.com/datasets/oezlem/airline-passenger-forecasting

14. Brownlee,Jason, 2021, Introduction to Time Series Forecasting with Python, Machine Learning Mastery.