Demand/Sales Forecasting in Supply Chain using LSTM Machine Learning model

Aastha Chawla, Anupama Mishra, Sowmya Sri Samudrala

The University of Texas at Austin

December 8, 2023

1 Introduction and Motivation

A supply chain is a network of organizations, individuals, technologies, activities, and resources involved in the creation and delivery of a product or service to end customers. It encompasses the entire process from the initial production of raw materials to the final delivery of the product to the consumer. The goal of a supply chain is to efficiently and cost-effectively move goods or services from the point of origin to the point of consumption while meeting customer demands.

Demand forecasting is a key player in achieving this goal. It acts like a crystal ball, giving businesses insights into the future, and helping them predict the quantity of products, customers are likely to buy, which in turn helps in minimizing costs. This prediction goes beyond just numbers and has a profound impact on various aspects of business operations:

- **Optimizing Inventory Levels:** In an ideal world, products are readily available without dealing with excess inventory. Accurate demand forecasting achieves this delicate balance, helping businesses avoid the challenges of holding too much stock (costly storage, outdated products) or too little (lost sales, frustrated customers).
- **Streamlining Production Planning:** Precise demand forecasting guides factories in production scheduling. By knowing exactly what to produce and when - manufacturers can prevent underproduction (leading to lost sales) and overproduction (tying up resources and capital).
- **Facilitating Supply Chain Collaboration:** A well-functioning supply chain relies on smooth communication and coordination between all involved parties. Demand forecasting fosters collaboration by giving everyone a clear picture of future demand, preventing issues like product shortages or surpluses.
- **Boosting Customer Satisfaction:** Happy customers are the backbone of any business. Accurate demand forecasting ensures that products are available when and where customers expect them. This results in a smoother buying experience, building customer satisfaction and loyalty, ultimately contributing to business success.

Machine learning (ML) plays a significant role in improving demand forecasting by leveraging data-driven approaches to analyze historical patterns and predict future demand more accurately. Below is the basic architecture of using machine learning in demand forecasting (Refer Figure 1).

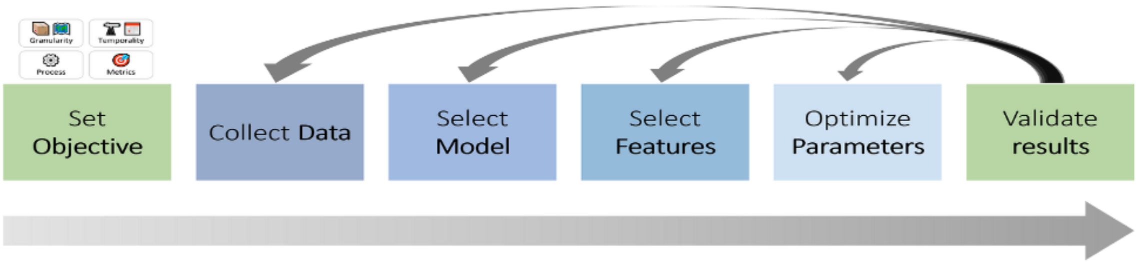

Figure 1: Machine Learning Basic Architecture

Some of the benefits of using ML in demand forecasting include:

- **Improved Accuracy:** Machine learning helps make demand forecasts more precise by analyzing historical data and identifying patterns that may not be apparent through traditional methods. This leads to more accurate predictions of future customer needs.

- **Adaptability to Changes:** ML models can adjust to shifts in market conditions and customer behaviour in real time. This adaptability allows businesses to respond quickly to changes, ensuring that forecasts remain relevant and reliable.

- **Timely Decision-Making:** ML algorithms process large volumes of data quickly, enabling timely decision-making. This is crucial in fast-paced business environments where quick responses to changing market conditions are essential for success.

- **Continuous Improvement:** Machine learning models learn from new data over time, continuously improving their forecasting accuracy. This adaptability allows businesses to stay competitive in dynamic markets by evolving their forecasting methods.

Some of the popular ML models used for demand forecasting include Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA), Exponential Smoothing, Long Short-Term Memory (LSTM) and XGBoost. Figure 2 shows an example of the output of how ML models predict future demand and how it is compared with the true demands.

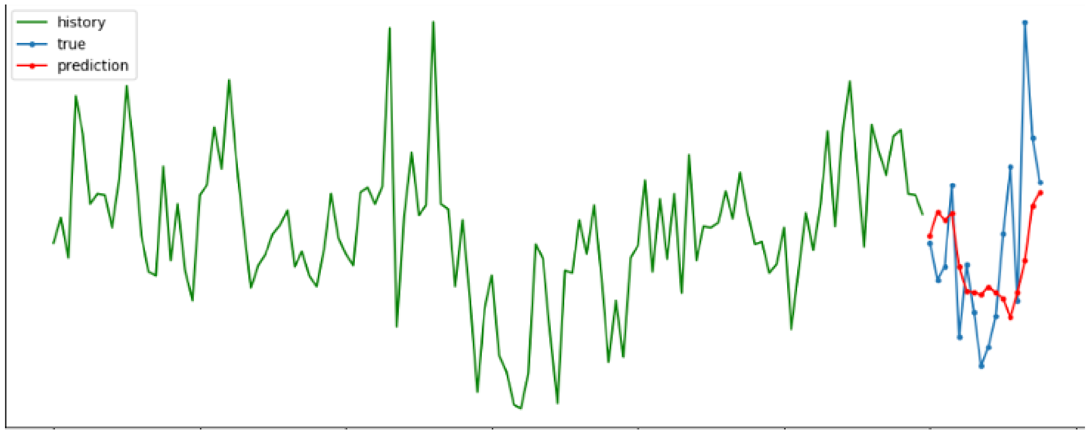

Figure 2: Example of ML model prediction of future demand

This report aims to delve into the challenges encountered in demand forecasting and to provide a comprehensive examination of the solution utilizing the Long Short-Term Memory (LSTM) machine learning model.

2 Literature Review

The literature review performed in this research has identified relevant papers related to Machine Learning applied to the demand forecasting stage of supply chain management and its evolution over time in various industries.

Historically, traditional statistical methods such as ARIMA and exponential smoothing have been the go-to approaches for demand forecasting. However, recent years have witnessed a significant shift in perspective and methodology, with ML methods gaining prominence. Aamer et. al [1] concluded in their research that most of the advancement in the machine learning side of supply chain management started only in 2018 with a major focus on the application of neural network algorithms in demand forecasting. We can observe the change in the perspective of using ML methods in supply chain management over time. In 2007, Carbonneau et. al [4] concluded in their comparison study that ML techniques have relatively poor performance as compared to traditional methods like exponential smoothing in forecasting distorted demand (single time series). However, later in 2020, Spiliotis et. al [8] did a similar comparison between the performance of statistical methods and ML methods trained in a series-by-series fashion for daily SKU demand forecasting and concluded that ML methods provide significantly less biased and more accurate forecasts. Additionally, it is shown that when time series features are used in addition to historical data, it improves the forecasting performance. This change in perspective happened majorly due to the advancement of technology in the field of data analytics. Maryam et. al. [10] further showed how traditional statistical methods for demand forecasting are inaccurate, so machine learning algorithms such as extreme learning machine, gradient boosting, and multi-layer perceptron are used instead to examine the accuracy of demand forecasting.

Tugay et. al.[9] in their work explores demand prediction techniques through machine learning methods, specifically emphasizing Stacked Generalization. The study highlights the effectiveness of combining various machine learning methods, particularly Stacked

Generalization, in improving demand prediction models. Mitra et. al [7] in their work compare various demand forecasting models for a multi-channel retail company and introduce a novel hybrid machine learning approach. The study explores the effectiveness of different methods, providing insights into improved forecasting techniques for retail businesses. Abbate et. al [3] in their research explore demand forecasting for delivery platforms through neural network techniques. It delves into the application of neural networks to predict demand patterns, providing valuable insights for efficient delivery management. The research emphasizes leveraging advanced computational methods for accurate demand predictions in the context of delivery services.

As for more recent developments, Yun Dai et.al. [5] present the increasing use of deep learning for time series forecasting in recent years. Long Short-Term Memory (LSTM) networks are particularly effective for this task, as they can learn long-term dependencies in the data. A few examples were studied such as using LSTM to forecast sales of a retail store, and finding that it outperformed traditional forecasting methods such as ARIMA and exponential smoothing. Another application was to capture the seasonality and trends by forecasting sales of a fashion retailer and then accurately predicting sales even during periods of high demand. Suleka et. al. [6] enhance LSTM as a type of recurrent neural network (RNN) which is used in deep learning. LSTMs can retain long-term dependencies, which makes them useful for tasks such as sales forecasting, where the value of a particular day's sales may be influenced by the sales of previous days with improved accuracy. Abbaspour Ghadim Bonab, A [2] performs a study that compares the accuracy of five ML models in forecasting a univariate time series, whose results were based on the accuracy of the forecasting according to the values of forecasting errors. The RMSE and MAE error measures as well as the R, correlation coefficient were used to assess the forecasting accuracy of the models.

3 Problem Description and Model

3.1 Problem Description

Limitations of Conventional Forecasting Techniques:
Forecasting future demand holds significant importance across various industries, enabling informed decision-making regarding inventory management, resource allocation, and production planning. However, conventional methodologies encounter challenges in precisely capturing the intricate patterns inherent in contemporary datasets, as mentioned below.

- **Temporal Dependencies and Sequential Patterns:** Temporal dependencies and sequential patterns pose challenges for conventional statistical models when analyzing demand data. These models often fall short of accommodating shifting trends, seasonal variations, and irregular fluctuations commonly observed in real-world demand situations.

- **Nonlinear and Complex Relationships:** The behaviour of demand often showcases intricate, non-linear relationships influenced by various complex factors including consumer behaviour, market trends, and external forces. Conventional models, built on linear assumptions, encounter difficulties in comprehending and accurately predicting outcomes within these intricate environments.

- **Data Volatility and Variability:** The ever-changing market dynamics and shifting consumer preferences introduce volatility and variability in demand data. Traditional methods face challenges in adjusting to these fluctuations, resulting in less-than-optimal forecasting accuracy.

3.2 Solution to the problem: LSTM

Given these challenges and limitations, it becomes essential to consider the implementation of Long Short-Term Memory (LSTM) neural networks for demand forecasting. Based on the literature review, LSTMs, a specialized variant of recurrent neural networks, demonstrate exceptional aptitude in capturing extensive dependencies, managing sequential data, and comprehending intricate patterns. Their inherent capacity to retain and selectively discard information across prolonged sequences offers a promising solution to rectify the deficiencies present in traditional methodologies. Due to the following reasons, we chose the LSTM model for forecasting in our research (Refer Figure 3).

- **Capturing Long-Term Dependencies:** LSTM models can recognize and leverage long-term dependencies in time-series data, which is particularly beneficial for demand forecasting.
- **Handling Non-Linear Relationships:** LSTM models can capture non-linear relationships between variables, providing more accurate predictions compared to linear models like ARIMA.
- **Dealing with Seasonality:** LSTM models can effectively handle seasonal patterns in demand data, a crucial factor in many industries.
- **Adaptability to Changing Patterns:** LSTM models can adapt to changing patterns in demand data, making them more robust and reliable for forecasting.

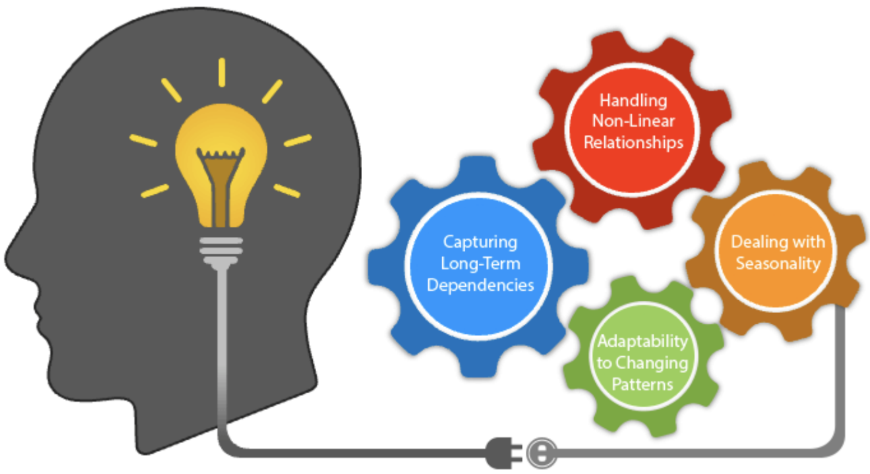

Figure 3: Advantages of LSTM

4 Data Architecture

4.1 Data Set

The dataset utilized for our analysis spanned a comprehensive timeframe, covering a period of 5 years from January 1, 2013, to December 31, 2017. This dataset specifically focused on the sales records associated with various items across different stores. Within this dataset, there were four primary columns or attributes (Refer Figure 4):

- **Date:** This column recorded the specific dates corresponding to each sales transaction. Notably, the recorded dates did not factor in any holiday effects or instances of store closures. Each entry in this column represented a date on which sales were recorded.

- **Store:** The Store column comprised distinct Store IDs, uniquely identifying each of the ten stores considered in the dataset. Each transaction was linked to a specific store location through its corresponding Store ID.

- **Item:** This column contained Item IDs that served as unique identifiers for the different products or items available for sale. There were a total of 50 unique items represented within this dataset, each linked to specific sales transactions.

- **Sales:** This numerical column denoted the number of items sold for a particular product at a specific store on a given date. It reflected the sales volume or quantity of items transacted, offering insight into the demand or popularity of specific items across various stores over time.

Raw Data			
Date	Store	Item	Sales
1/1/13	1	1	13
1/1/13	1	2	33
1/2/13	1	1	11
1/2/13	1	2	43
1/2/13	1	3	30
1/1/13	2	1	12
1/1/13	2	2	41
1/2/13	2	1	16
1/2/13	2	2	33
1/2/13	2	3	32

Figure 4: Sample of Raw Data

The dataset provided a comprehensive overview of the sales dynamics, detailing the number of items sold across different stores for each distinct item over the specified 5-year duration. This raw dataset structure allowed for further analysis and exploration to derive insights into sales patterns, trends, and correlations between various factors such as date, store, item, and sales volume.

4.2 Understanding the data

To gain a deeper comprehension of the dataset, we conducted visual representations by plotting the aggregated sales data against the date, presented in Figure 5. Upon initial examination of this plot, distinct patterns emerged: noticeable seasonal variations in sales and a consistent upward trend in sales volume with the progression of each passing year were observed.

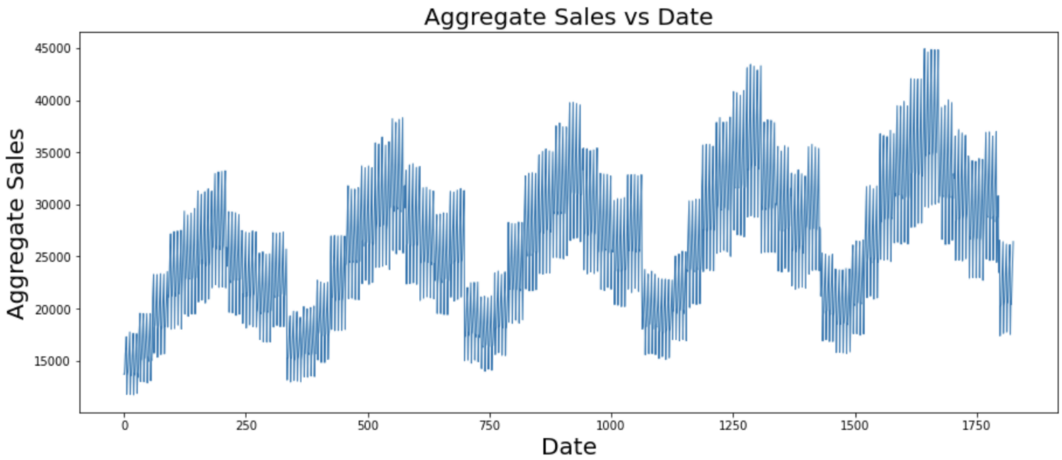

Figure 5: Aggregate Sales vs Date

Additionally, we generated separate plots illustrating the aggregate sales concerning stores (depicted in Figure 6) and items (depicted in Figure 7). These plots enabled us to understand seasonal patterns in the sales trends concerning both stores and items. This observation suggested consistency in the seasonality of sales across different stores and various items, consequently mitigating any potential bias inherent in the overall dataset.

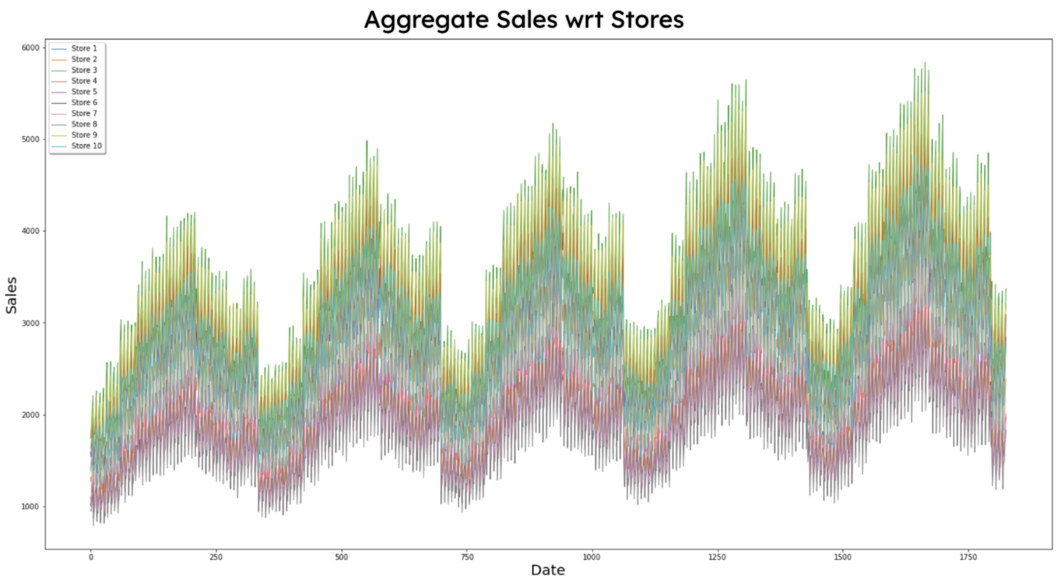

Figure 6: Aggregate Sales vs Date wrt Stores

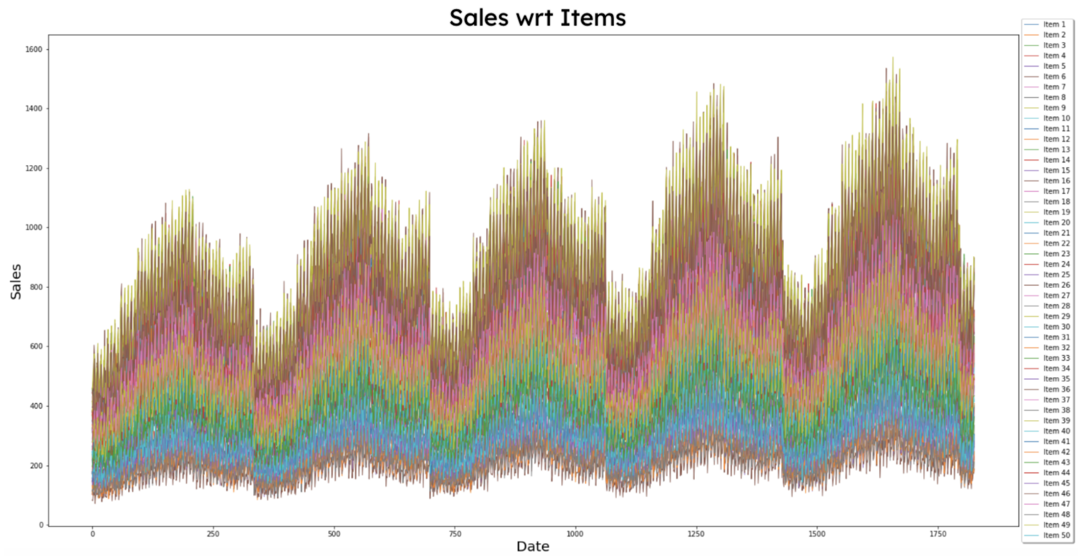

Figure 7: Aggregate Sales vs Date wrt Items

By visualizing sales trends across time, stores, and items, we aimed to uncover and validate patterns, thereby facilitating a more comprehensive and unbiased understanding of the dataset's underlying dynamics.

5 Experimental Design

5.1 Long Short-Term Memory (LSTM) algorithm developed

In this experimental design, the primary objective is to predict future sales leveraging historical data through the implementation of an LSTM (Long Short-Term Memory) model. The first step was to have every column in the correct format, hence we converted the 'date' column to a datetime object to visualize overall daily sales patterns across time.

- **Feature Engineering:** After converting the date column, we started with feature engineering, where we transformed the data into a time series problem using a seriestosupervised function to create time series features, with experimentation around different window sizes to capture temporal dependencies effectively. The goal is to prepare the data for training a machine learning model, where past observations are used to predict future observations. The subsequent train-test split ensures a clear separation between training and validation sets, with an emphasis on sufficient overlap between the two periods.

- **Exploratory Data Analysis (EDA):** Furthermore, daily sales with respect to stores and items were plotted to identify variations and trends. On looking at the plots, we saw seasonal trends in sales against stores and items and aggregated amounts. Hence, we found a correlation between stores and items and decided to aggregate the total sales by using feature correlation analysis. As part of exploratory data analysis, statistical properties of the time series data are examined, and autocorrelation and partial autocorrelation functions are visualized to identify potential lags. As a result, a few lags that gave the best results included 89, 59 and 29 days with "29 days" being the best in terms of autocorrelation plots. Hence we decided to use a lag of "29 days" (Refer Figure 8) to forecast the next 90 days (Refer Figure 9).

Lag: 29 days		
Item (t-29)	Store (t-29)	Sales (t-29)
1.0	1.0	19.0
1.0	1.0	15.0
1.0	1.0	10.0
1.0	1.0	16.0
1.0	1.0	14.0

Figure 8: Lag - 29 Days

Forecast: 90 days		
Item (t+90)	Store (t+90)	Sales (t+90)
1.0	1.0	33.0
1.0	1.0	15.0
1.0	1.0	21.0
1.0	1.0	29.0
1.0	1.0	19.0

Figure 9: Forecast - next 90 days

- **LSTM Model Algorithm:** The LSTM model architecture is then designed using Keras, featuring an input layer with LSTM units and activation functions (Used - Relu), and an output layer with a dense unit for regression. Multiple layers and dropouts may be incorporated for regularization purposes. Our model was trained, and performance was monitored by evaluating training and validation loss over epochs. We chose a loss function of Mean Squared Error and optimizer as Adam. Following this, hyperparameter tuning was conducted, exploring different window sizes, LSTM units, and learning rates to optimize model performance.

- **Hyperparameter Tuning:** Hyperparameter tuning is a crucial step in optimizing the performance of the LSTM model. The goal is to systematically explore different combinations of hyperparameters to identify the configuration that yields the best results. Several hyperparameters can be fine-tuned to enhance the model's predictive capabilities. Firstly, the window size, which represents the number of past time steps considered for prediction, can significantly impact the model's ability to capture temporal patterns. Experimenting with various window sizes, such as 7 days or 30 days, allows for an assessment of how well the model generalizes to different temporal contexts. In our case, we used different lags such as 89, 59 and 29 days but we got the best result with "29 days". Adjusting the number of LSTM units is another critical hyperparameter. A higher number of units may allow the model to capture more complex patterns, but it also increases the risk of overfitting. Conversely, a smaller number of units may result in underfitting. Hyperparameter tuning involves finding the optimal balance for the specific dataset. Based on our data, we used 50 units in the first layer of our LSTM model.

- **Model Training:** During training, the LSTM network learns to adjust the parameters within the LSTM cells to minimize the difference between its predictions and the actual demand data. The network uses a loss function to measure this difference, and optimization algorithms, such as gradient descent, are used to adjust the parameters. The first RMSE that we got without any data pre-processing came out to be 25.6 and hence we did some fine-tuning in our model using the results we got earlier from EDA and hyperparameter tuning. We also explored regularization to prevent overfitting, techniques like dropout and recurrent dropout were applied to LSTM layers.

- **Data Pre-processing:** The research aims to train the LSTM model with historical sales data, using various architectures to experiment with different configurations. It included testing the model on a larger dataset, allowing for more accurate forecasts. Additionally, data preprocessing was done to remove outliers and impute missing values, feature elimination to remove correlated features, and in-time predictions to assess the model's performance on unseen data. The steps of Data Pre-processing are further explained in section 5.2 in detail with an impact of it on the final result of prediction.

- **Deployment and Forecasting:** Once trained and evaluated, the LSTM model was deployed to make future demand forecasts by feeding historical data into the network and generating predictions for upcoming time steps. The ultimate goal to improve the accuracy of time series forecasting was achieved after the Data Preprocessing step was implemented.

5.2 Data Preprocessing

Before training the LSTM model, we conducted thorough data preprocessing to ensure the quality and reliability of the dataset once it is trained. After doing date conversion and studying the plots of sales wrt stores and items, we adjusted the time range of the dataset such that it was filtered to include only data from January 1, 2017, onward. This adjustment was likely made to focus the analysis and training on a specific time range, potentially excluding earlier data that might not be relevant to the model's training. The following preprocessing steps were taken -

1. Outlier Detection and Handling: We implemented outlier detection techniques, such as Z-score or IQR, to identify and address outliers in the dataset. Data points exceeding a certain threshold were considered outliers and removed or corrected. In our case, the data was filtered to be only after January 1, 2017, onwards because of the seasonality in the trends (Refer Figure 10).

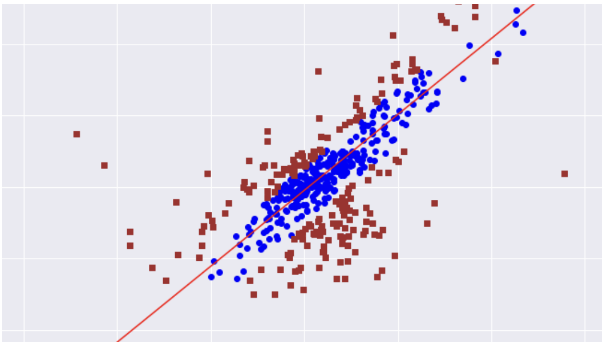

Figure 10: Outlier Detection (marked with red)

2. Missing Value Imputation: Missing values were detected from 2017 onwards and imputed using appropriate techniques, such as mean imputation to ensure that the dataset was complete.

3. Feature Elimination: To improve the model's efficiency and reduce the risk of overfitting, feature elimination was performed. We used correlation analysis to identify highly correlated features. In our case, stores and items performed the same, in terms of seasonality with respect to the sales so both of these features exhibited high correlation and hence were eliminated from our time series data, as they could introduce multicollinearity and reduce model interpretability. Hence, unnecessary columns were dropped from the dataset related to 'item' and 'store' at different time points, as well as the original 'item(t)' and 'store(t)' columns.

4. Time Series Transformation: We grouped the data by 'item', 'store', and 'date', calculating the mean sales for each group. This transformation was done to convert the data into a time series format that is more suitable for training an LSTM model. The seriestosupervised function was applied to convert the time series data into a supervised learning problem. This function created lag features, representing past observations, and prepared the data for training the LSTM model to predict future sales. Rows were filtered to include only those where the 'store' and 'item' values matched the corresponding values from the shifted columns. This step ensured consistency in the data used for training.

5. In-Time Predictions: Because we didn't have out-of-time data we decided to do in-time predictions for the last 4 weeks of 2017. In-time predictions involved training the model on a historical subset of the data and testing its performance on a subsequent time period within the same dataset. This simulated the model's real-time forecasting capabilities on unseen data. The code we designed extracted the labels for the prediction task where labelscol was set to 'sales(t+lagsize)', representing the future sales target variable (future sales). The traintestsplit from scikitlearn is used to split the dataset into training and validation sets where Xtrain and Ytrain represent the features and labels for the training set, while Xvalid and Yvalid represent those for the validation set. The testsize parameter was set to 0.4, indicating a 60-40 split between the training and validation sets.

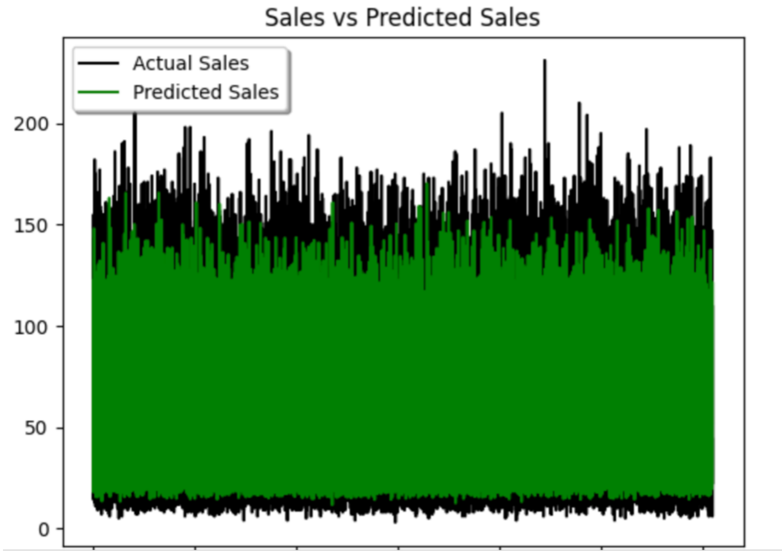

Figure 11: Actual vs Predicted Sales

5.3 Model Evaluation and Results

The evaluation of predictive models is crucial for assessing their accuracy and performance. The LSTM model's performance is evaluated using appropriate metrics like Root Mean Square Percentage Error (RMSE) by comparing the predicted demand values to actual demand values for in-time predictions. Using this experimentation in the time-series model, we were able to reduce the RMSE to 19.6 in the next iteration after data pre-processing. RMSE quantifies the relative accuracy of the forecasts, considering both the magnitude and direction of errors. A lower RMSE indicates more accurate predictions, as it signifies that the percentage errors between predicted and actual values are smaller. This improved model with lower RMSE will contribute to more effective inventory management and better decision-making.

1. Model Evaluation plot: Once our model training was done, we predicted the sales using the lag of '29 days' for the next 90 days and further calculated RMSE using the meansquarederror function from scikitlearn. Once we computed the mean squared error between the actual (Yvalid) and predicted sales then numpy.sqrt was applied to obtain the square root hence resulting in the RMSE which told us how well the model performed on the validation/test dataset. Furthermore, we created a visual representation of the "Sales vs Predicted Sales" graph telling us that how well the LSTM model's predictions align with the actual sales values (Refer Figure 11). The actual sales line is represented by black (Yvalid) whereas the predicted sales are represented using the green line. Each point on the black line corresponds to the true sales value while each point on the green line represents the sales values predicted by the LSTM model for the same time periods (Last 4 weeks of 2017) as the actual sales.

2. Interpretation of result from the plot: Ideally, the green line (predicted sales) should closely follow the path of the black line (actual sales). A close alignment indicates that the model is accurately capturing the underlying patterns in the data. The graph allows us to visually inspect how well the model predicts various patterns and trends in the sales data. As shown, the black line is a little over the predicted sales but follows the same pattern, hence we are underpredicting than the actual sales. Instances, where the green line deviates significantly from the black line, suggest potential areas where the model is less accurate in its predictions. Our model is not overfitting because the green line doesn't follow the black line too closely hence the model is not fitting the training data way too precisely. Conversely, we couldn't see substantial gaps between the lines, hence there is no indication of underfitting. In summary, the "Sales vs Predicted Sales" graph serves as a valuable tool for assessing the LSTM model's predictive capabilities and gaining insights into its performance on unseen data.

3. Quantitative Assessment from the RMSE: While visual inspection provides qualitative insights, the Root Mean Squared Error (RMSE) or other quantitative metrics should be considered for a more precise evaluation of the model's accuracy. If the model predictions closely align with actual sales, it suggests that the LSTM model is effective in capturing and learning from the historical patterns in the data. A final RMSE of 19.6 that we got indicates that, on average, the model's predictions deviate by approximately 19.6 units from the actual sales values. This provides a measure of the magnitude of errors across the dataset. Since in our case, the baseline RMSE (25.6) is higher than the RMSE of our model which states that our LSTM algorithm is performing better. For time series forecasting, an RMSE of 19.6 seems acceptable, especially if the sales data exhibits seasonality over time.

6 Conclusion and Future Research Directions

In conclusion, while an RMSE of 19.6 suggests room for improvement and the graphical visualization offers insights into the model's ability to capture sales patterns. The next steps involve an iterative approach to refinement, incorporating feedback from the RMSE and graphical analysis, and aligning the model with business objectives. The continuous improvement of the model will contribute to its effectiveness in predicting sales with greater accuracy.

If there are discrepancies, further analysis and potential model refinement may be necessary. While an RMSE of 19.6 provides a good assessment, it's advisable to explore opportunities for model refinement for the future scope. This might involve tweaking hyperparameters, adjusting the model architecture, or exploring advanced techniques. In our case, an iterative process of improvement can be undertaken by making adjustments to the model based on insights gained from the data and performance evaluation from previous iterations. We can also investigate specific instances or time periods where the model's errors are higher in the future. Understanding these patterns and reasons for high errors can guide us in targeted improvements.

Further exploration of hyperparameters, such as adjusting the number of LSTM units even more, tuning learning rates, and incorporating dropout layers, can be performed to optimize model performance. Additionally, experimentation with additional features or transformations may enhance the model's ability to capture complex patterns in the sales data. In fact, besides RMSE, we can also explore other evaluation metrics, such as Mean Absolute Error (MAE) or custom metrics tailored to business objectives, which can offer a more comprehensive understanding of model performance.

References

[1] Ammar Aamer, LuhPutu Eka Yani, and IMade Alan Priyatna. Data analytics in the supply chain management: Review of machine learning applications in demand forecasting. *Operations and Supply Chain Management: An International Journal*, 14(1):1–13, 2020.

[2] A. Abbaspour Ghadim Bonab. Comparative study of demand forecasting based on machine learning methods with time series approach. *Journal of applied research on industrial engineering*, 9(3):331–353, 2022.

[3] R Abbate, P Manco, M Caterino, M Fera, and R Macchiaroli. Demand forecasting for delivery platforms by using neural network. *IFAC-PapersOnLine*, 55(10):607–612, 2022.

[4] Real Carbonneau, Rustam Vahidov, and Kevin Laframboise. Machine learning-based demand forecasting in supply chains. *International journal of intelligent information technologies (IJIIT)*, 3(4):40–57, 2007.

[5] Yun Dai and Jinghao Huang. A sales prediction method based on lstm with hyper-parameter search. *Journal of Physics: Conference Series*, 1756(1):012015, 2021.

[6] Suleka Helmini, Nadheesh Jihan, Malith Jayasinghe, and Srinath Perera. Sales forecasting using multivariate long short term memory network models. *International Journal of Information Technology*, 14(4):1937–1947, 2022.

[7] Arnab Mitra, Arnav Jain, Avinash Kishore, and Pravin Kumar. A comparative study of demand forecasting models for a multi-channel retail company: a novel hybrid machine learning approach. In *Operations Research Forum*, volume 3, page 58. Springer, 2022.

[8] Evangelos Spiliotis, Spyros Makridakis, Artemios-Anargyros Semenoglou, and Vassilios Assimakopoulos. Comparison of statistical and machine learning methods for daily sku demand forecasting. *Operational Research*, pages 1–25, 2020.

[9] Resul Tugay and Sule Gunduz Oguducu. Demand prediction using machine learning methods and stacked generalization. *arXiv preprint arXiv:2009.09756*, 2020.

[10] Maryam Zohdi, Majid Rafiee, Vahid Kayvanfar, and Amirhossein Salamiraad. Demand forecasting based machine learning algorithms on customer information: an applied approach. *International Journal of Information Technology*, 2022.