Time - Series Forecasting in Retail Industry using Bidirectional, Stacked and Vanilla LSTM

Harshini S
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India
hs4351@srmist.edu.in

Lekhashree V
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India
ls7610@srmist.edu.in

S. Manohar
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India
manohars@srmist.edu.in

***Abstract*— Recently, interest in deep learning research and its applicability to practical issues has grown significantly. Developing a time-series analysis model to comprehend sales and profits/losses, as well as forecast future values, is crucial for businesses and companies, whether they operate online or offline. The objective of this study is to construct a time-series analysis model that can comprehend sales and profits/losses while forecasting future values. To achieve an effective analysis, we have chosen Long Short-Term Memory (LSTM) deep learning architectures, including Stacked LSTM, Vanilla LSTM, and Bidirectional LSTM (Bi-LSTM). LSTM, in contrast to conventional recurrent neural networks, can handle time steps of varying sizes without encountering the issue of vanishing gradients. Additionally, they overcome the limitation of the stationarity assumption that is present in models like ARIMA, making them a more flexible and powerful tool for time-series analysis. The three distinct LSTM models are used to train the dataset and are compared with each other with respect to their accuracy measures. The conclusion of the thesis suggests that utilizing the Stacked LSTM deep learning architecture can greatly enhance the accuracy of sales prediction using financial data. Also, the thesis includes the forecast for the next 12 months. The implications of this thesis are significant for businesses and companies, as accurate sales prediction can help in making informed decisions related to production, inventory management, and marketing strategies. Furthermore, the findings of the thesis can also contribute to the realm of deep learning research, particularly concerning of time-series analysis.**

Keywords*— Deep Learning, Long Short – Term Memory, Vanilla LSTM, Stacked LSTM, Bidirectional LSTM

I. INTRODUCTION

The time series is very extensive and utilized in various fields such as language processing and speech popularity, traffic analysis, weather forecasting, unemployment rate analysis and so on. many other fields. Some sequential modelling strategies include estimating certain parameters to satisfy a hypothetical form of time collection, including autoregression (AR), autoregressive moving average (ARMA) and common move-associated autoregression (ARIMA). Due to the noisy and complex nature of such time collection, the important patterns cannot be captured by classical methods, which mostly depend only on linear regression and parameter estimation. Most economic time series tend to exhibit non-linear trends in their structure. Forecasting sales behaviour is a challenging task without the use of complex and nonlinear modelling tools. However, deep learning offers a solution by allowing prediction and classification operations based on intricate and hard-to-decipher educational data. Deep Neural Networks (DNNs) have shown good overall performance in several additional application areas, including signal processing, image types and speech prevalence. Therefore, applying leveraging Long Short-Term Memory (LSTM) to financial time-series forecasting is a potential strategy worth pursuing, as DL is well-suited it and for many studies have developed different deep learning strategies for predicting time series information. Among these methods, LSTM has attracted interest due to its ability to recall previous inputs and use current information to evaluate network weights.

The goal of this study is to create a sales forecasting application for the retail industry by leveraging LSTM time series models. By using deep learning techniques like LSTM, this application can improve the accuracy of sales predictions and enable businesses to make informed decisions about allocating resources, managing cash flow, and forecasting short-term and long-term performance. LSTM models are specifically designed to address the problem of long-term dependency, making it easier for them to retain information for extended periods.

To develop a comprehensive sales forecasting model that can adapt to changing market trends and consumer behavior, this study will compare the performance of Stacked LSTM, Vanilla LSTM & Bi-LSTM. By analyzing historical sales data and applying deep learning algorithms, this model aims to provide accurate sales predictions for the retail industry.

By using deep learning algorithms to analyze sales data, businesses can make informed decisions about resource allocation and cash flow management, leading to better overall performance and long-term success.

The study centres the following:

1. Data collection.
2. To pre-process the dataset, analyse and extract required variable.
3. Build, train, and test sales forecasting models.
4. Compare performance evaluation between Stacked LSTM, Vanilla LSTM and Bi-LSTM models.
5. Forecast the next 12 months using the best model.

The structure of the study is as follows:

Section I discusses about introduction; section II discusses about literature review related to time series analysis and deep learning models. Section III discusses about model building using Vanilla LSTM, Stacked LSTM and Bi-LSTM. Section IV discusses about conclusion and future scope of the study.

II. LITERATURE REVIEW

There has been a significant increase in the application of LSTM deep learning architecture to time series forecasting, with the aim of improving accuracy, transparency, and efficiency. Several research works have suggested the utilization of LSTM in time series forecasting due to its capability to handle time steps with variable lengths and overcome the vanishing gradient issue. These findings have important implications for improving the accuracy and reliability of time series forecasting, particularly in domains such as finance, economics, and meteorology.

Gopalakrishnan.T et al. [1] implemented the linear regression using cost function and gradient descent. They obtained real time sales dataset from 2011-2013 to predict sales for 2014. The study compares actual values with the predicted sales values to calculate the accuracy rate and to validate the prediction. Rishi Raj Sharma et al. [2] discuss about the implementation of ARIMA model along with EVDHM (Eigen Value Decomposition Hankel Matrix) for non-stationary time series which is defined by the Philips-Perron Test (PPT). Generic Algorithm (GA) has been used to optimize the ARIMA parameters with minimizing AIC (Akaike Information Criterion) values. Based on the historical dataset, Mehat Vijh et al. [3] developed ANN (Artificial Neural Network) and Random Forest to forecast the next day stock's closing price. Comparative investigation using RMSE, MAPE, and MBE shows that ANN provides superior prediction. Regression techniques like Linear Regression and Polynomial Regression were used by Saud Shaikh et al. [4] to analyze and forecast the COVID-19 outbreak in India. Polynomial Regression outperforms other models, according to analysis of the models using R squared score and error values. Using a variety of machine learning models, including the Decision Tree (DT), Generalized Linear Model (GLM), Gradient Boost Tree, Sanjay. N. Gunjal et al. [5] performed Big-Mart sales prediction (GBT). Using error metrics like RMSE, MSE, and MAE to compare the models, it is found that GBT exhibits good accuracy.

In order to estimate the sales based on the Big-Mart dataset, Varshini S. and Dr. D. Preethi [6] analyzed machine learning models such as XGBoost Regressor, Random Forest Regressor, ANN, and SVR (Support Vector Regression). According to the evaluation criteria of RMSE, R2 score, and MAPE, Random Forest performs better. To analyze and forecast the Big-Mart Sales, Nayana R et al. [7] used the following ML models: Linear Regression, Ridge Regression, Polynomial Regression, and XGBoost Regression. XGBoost and Ridge Regression provide higher predictions based on accuracy rate. Meng-Chen Hsieh et al. [8] analyzed the impact of the supplier sharing their knowledge with the retailer on improving their own inventory-related expenses and forecasting, and how it influences the supplier's demand. The researchers discovered that when the retailer uses a suboptimal exponential smoothing (SES) forecast, the supplier can recover the retailer's actual shocks and that with a thorough record of the retailer's orders, the supplier can determine the real ARMA model that creates the retailer's demand pattern. Random Forest Regression, Support Vector Machine and Artificial Neural Networks were utilized by Xianghui Yuan et al. [9] to evaluate the profitability of multiple integrated stock selection models that employ different feature selection techniques and algorithms for predicting stock price trends. Findings indicate that applying Random Forest yields greater performance. Demand forecasting was carried out by Hossein Abbasimehra et al. [10] using various ML models, including ETS, ARIMA, ANN, SVM, KNN, basic RNN, and single layer LSTM. According to the calculated RMSE and SMAPE values, the LSTM outperforms the other models.

III. METHODOLOGY

First, a dataset consisting of approximately 10,000 sales records spanning a four-year period from 2019 to 2022 was obtained from Kaggle. The dataset includes several variables such as order ID, customer information, category, sub-category, city, order date, region (North, South, East, and West), sales, discount, profit, and state. To do forecasting based on time series, the primary step is to acquire data across the required time period. However, the obtained data may include mistakes, missing values, or duplicates. Therefore, the next step is to pre-process the data by handling missing values and converting datatypes to useable format (i.e.) to datetime type. When data is ready, exploratory data analysis can be performed to gain deeper insights into the data. For predicting sales, Stacked, Vanilla and bi-LSTM models are created. The created models are evaluated, displayed, and are used to forecast the future.

A. Data Preprocessing

The dataset has been prepared by managing missing values and organising the data. First, the "Order Date" property is turned into a datetime object. Statistical data analysis has been performed on the dataset, and the snacks categorical value has been separated as a new data frame. The sales on the same order dates in different regions of the cities in Tamil Nadu have been grouped together and added by the function called the sum aggregate. The snacks data frame has been set with order date as its index. Order dates related to the same year have been grouped by month, and the average return of all orders is analysed for every month and stored in series format. Exploratory data analysis has been performed using this data.

Value counts of both category and sub-category have been visualized. A Seasonal Decompose graph has been plotted, identifying the dataset as a seasonal time-series dataset. Monthly sales have been plotted to better understand the month-wise mean sales. The resulting series had 48 months of sales, which was split into a train-test ratio of 75:25. Both the train and test data were normalized using MinMax scaler transform to be passed into the model. A time series generator is a Python class that generates batches of data for the LSTM model during training is created. The TimeseriesGenerator class takes in several arguments, including the input data, the target data, the length of the input sequence, and the batch size, and outputs a generator object that can be used to feed data into the LSTM model during training. Input sequence - In this case, the LSTM will take in a sequence of 12 data points as input and output one data point as output. The study works with univariate time-series model so the number of features is 1 which is the sales count. The length parameter is set to 12, it means that the generator is designed to use the preceding 12 months of data to forecast the profit for the subsequent month. Additionally, the batch size has been set at 20. The TimeseriesGenerator object generates batches of data for the LSTM model during training. Each batch consists of a sequence of 12 data points and their corresponding target values. By using a generator to feed data into the model during training, we can avoid loading the entire dataset into memory at once, which can be important for large datasets

B. Model Building

1. Vanilla LSTM

Vanilla LSTM is a basic version of the LSTM architecture that has a single hidden layer of LSTM cells. The LSTM cells in the hidden layer take in a sequence of input data and output a sequence of hidden states, which can be used for predicting the next value in the sequence. The sequential model provides a straightforward approach to stack layers of a neural network on top of one another, without being reliant on the exact tensor shape or layer arrangement within the model. Subsequently, the Sequential constructor can be instantiated to create the model. Fig.1 represents Vanilla LSTM model.

First a NumPy array of shape (12, n) is generated, where n is set to 10. This initializes a matrix of zeros with 12 rows and 10 columns. Then, for each column in the matrix (for each iteration of the loop), it creates a vanilla LSTM model with an input shape of (12, 1), consisting of a single LSTM layer containing 50 neurons, along with two dense layers, each containing 100 neurons. Both the LSTM and dense layers employ the Rectified Linear Unit (ReLU) activation function. Additionally, there is an output layer containing a single neuron. In order to train the model, it was compiled using the Adam optimizer and the loss function called mean squared error (MSE) has been utilized.

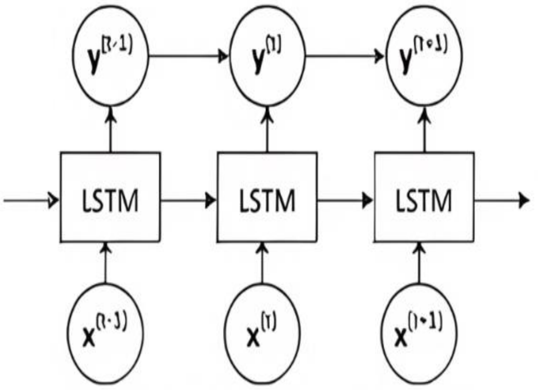

Fig. 1 Vanilla LSTM Model

2. Stacked LSTM

The Stacked LSTM model is composed of numerous LSTM layers arranged on one another. Each layer processes the input sequence, and its output is employed as the input for the subsequent LSTM layer. This design allows the model to understand the intricate temporal dependencies and identify long-term trends in the data, hence boosting the accuracy of the model's predictions. Fig.2 represents Stacked LSTM model.

First a NumPy array of shape (12, n) is generated, where n is set to 10. This initializes a matrix of zeros with 12 rows and 10 columns. Then, for each column in the matrix (for each iteration of the loop), it creates a stacked LSTM model with an input shape of (12, 1), consisting of two LSTM layers with 50 neurons each and define return_sequences=True to pass on the output of each LSTM layer to the next layer, two dense layers with 100 and 50 neurons respectively, both LSTM & dense layer with ReLU activation function. And an output layer with one neuron. Adam optimizer is implemented to the model for compilation and the loss function called mean squared error (MSE) has been employed.

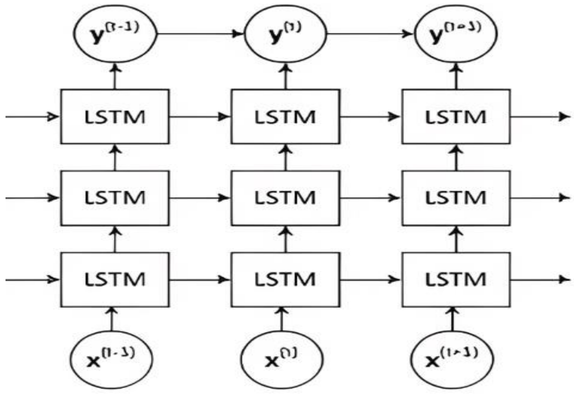

Fig. 2 Stacked LSTM Model

3. Bidirectional LSTM

The Bi-LSTM model has the potential to examine a series of data in both forward and backward orientations. This model includes of two LSTM layers. The forward LSTM layer follows the classic sequence processing paradigm, where inputs are treated sequentially and each output is transmitted to the subsequent time step. On the other hand, the backward LSTM layer operates in opposite way, beginning from the final input and working towards the first input. The ultimate output for each time step arises from integrating the outputs of these two layers through concatenation. Therefore, the final output incorporates information from both sides of the input sequence, enabling the model to recognise subtle patterns and connections. Fig.3 represents Bi-LSTM model.

First a NumPy array of shape (12, n) is generated, where n is set to 10. This initializes a matrix of zeros with 12 rows and 10 columns. Then, for each column in the matrix (for each iteration of the loop), a Sequential model is created. Then, a Bi-LSTM layer with 50 units and ReLU activation function is used to enhance the model. A Dense layer with a single output unit is added to the model after the LSTM layer. This output layer produces the predicted value for the next timestep. To compile the model, an optimizer called “Adam” is employed along with the loss function MSE- mean squared error.

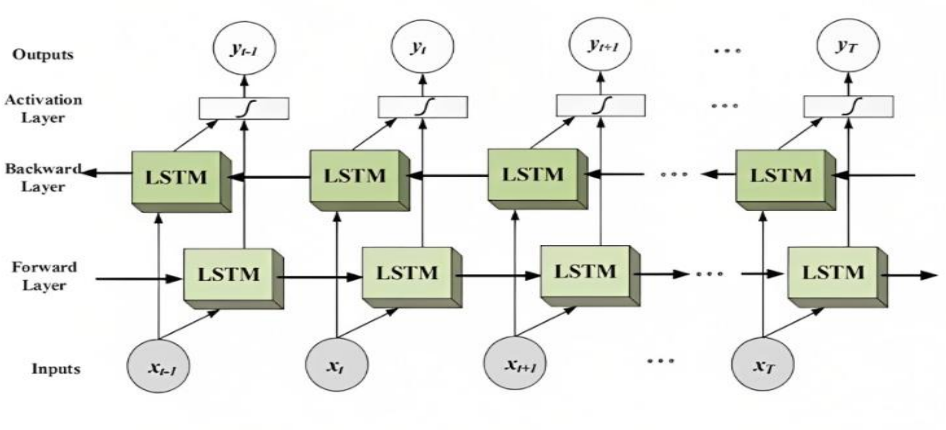

Fig. 3 Bidirectional LSTM Model

C. Training and Testing

The above 3 models are trained over the course of 200 epochs using the generator object.

After training, iterative predictions are made, where one time step is predicted based on the previous prediction. The resultant predictions are then transformed back to their original scale using a scaler object. Next, the prediction matrix column corresponding to the current iteration is updated with the predicted values for the 12-time steps. This process is repeated 10 times, generating 10 sets of predictions for the next 12-time steps using the respective LSTM model trained with the generator object. Finally, the 10 sets of predictions are converted into a one-dimensional array of length 12 by calculating the mean of each row.
Each of the three LSTM models goes through the above validation testing process individually.

D. Evaluation

To check the efficiency of the model, the "performance" function is applied. This function calculates and produces the following metrics between the forecasted sales and actual sales for the last 12 months contained within the dataset. They are:

- Mean Squared Error
- Root Mean Squared Error
- Mean Absolute Percentage Error

A lower value of MSE and RMSE indicates better performance of the model in predicting the sales values. Similarly, a lower value of MAPE indicates that the model can predict the sales values with higher accuracy.
Table I shows the performance metrics of three different LSTM models for sales forecasting for the next 12 months. The first column specifies the model used and the other three columns show the corresponding MSE, RMSE, and MAPE values.

In terms of performance, the Stacked LSTM model outperformed the other two models, as it had the lowest MSE and RMSE and MAPE. The Vanilla LSTM model had the highest MSE, RMSE and MAPE, indicating that it performed the worst of the three models. However, the Bi-LSTM model had a slightly lower MSE, RMSE and MAPE than the Vanilla LSTM model, indicating that it performed better in terms of relative error. Overall, the Stacked LSTM model is the most suitable for the given data and task.

TABLE I. PERFORMANCE METRICS OF THREE DIFFERENT LSTM MODELS

S. No	MODEL	MSE	RMSE	MAPE
1	Vanilla LSTM	17428.60	132.02	7.83
2	Stacked LSTM	7419.42	86.17	5.16
3	Bi-LSTM	14924.49	122.17	7.30

E. Forecasting

The predictions for the next 12 months are generated using the stacked LSTM model because of its high performance. An empty array is created to store the predictions. We then loop through each month in the prediction horizon (12 months) and create a list to store the predicted values for that month. Next, a batch of test data is created from the last 12 months of actual data, and used the stacked model to predict the next value in the sequence. This predicted value is appended to the list of predicted values, and the batch of test data is updated to include the predicted value. It then iterates over several times, predicting future sales using the LSTM model and storing the results in a list. After completing the predictions, the mean of the predicted values is calculated for each time step in the forecast horizon and store in a NumPy array. Finally, the array of mean values is reshaped into a 1-dimensional array and stored as the final set of predictions. This array represents our forecasted sales for the next 12 months.

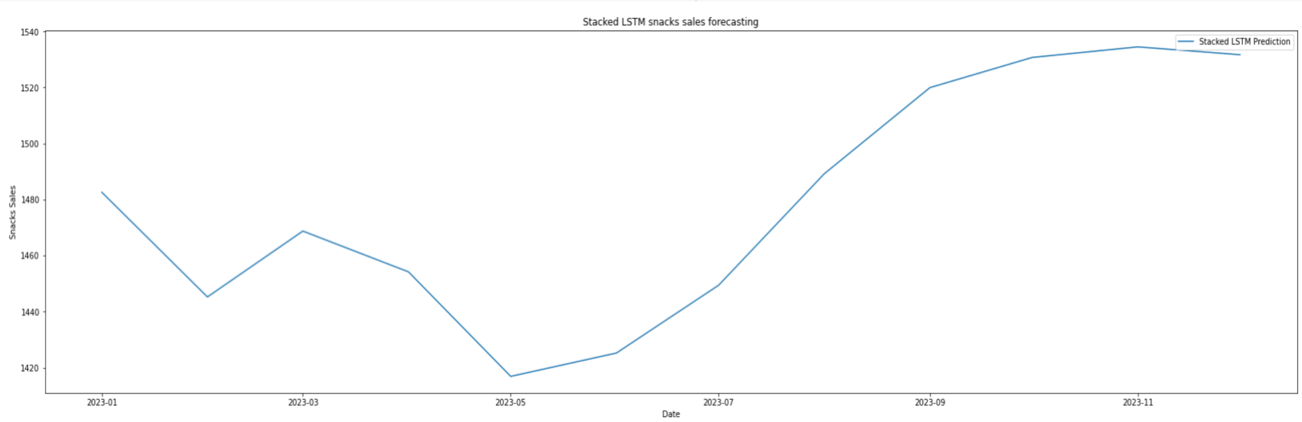

Fig. 4 Forecast using Stacked LSTM.

IV. Conclusion and Future scope

Three alternative deep learning neural network models such as Stacked, Vanilla and Bi-LSTM were used in this study to analyze sales data. The objective was to predict sales for a dataset from a supermarket, specifically for the category of Snacks for the year 2023. Calculating the average monthly sales, the dataset was divided into 25% for validation testing and 75% for training. Following the definition of a time-series generator, the three LSTM models were constructed and fitted using the generator. As a result of having lower MSE, RMSE and MAPE than the Vanilla LSTM and Bi-LSTM models, the Stacked LSTM model outperformed them. Finally, sales for the year 2023 were predicted using the Stacked LSTM model.

Although LSTM models have demonstrated superior performance over ARIMA, SARIMA, and RNN models, they require a significant number of computational resources, such as processing power, memory, or time, to complete when dealing with large and complex datasets. Thus, it is advisable to terminate training as soon as a satisfactory level of accuracy is attained, as further increases in epoch count may not consistently enhance accuracy. Additionally, it should be noted that forecast accuracy diminishes with increasing time intervals between data points due to the loss of temporal data and limited data. Consequently, future research could investigate alternative deep learning models or the combination of stochastic and deep learning models, depending on the data characteristics. To support sales forecasting decision-making, companies may consider developing a web or mobile application.

REFERENCES

[1] G.T., R. Choudhary and S. Prasad, "Prediction of Sales Value in Online shopping using Linear Regression," 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2018, pp. 1-6, doi: 10.1109/CCAA.2018.8777620.

[2] R. R. Sharma, M. Kumar, S. Maheshwari and K. P. Ray, "EVDHM-ARIMA-Based Time Series Forecasting Model and Its Application for COVID-19 Cases," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-10, 2021, Art no. 6502210, doi: 10.1109/TIM.2020.3041833.

[3] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar, Stock Closing Price Prediction using Machine Learning Techniques, Procedia Computer Science, Volume 167, 2020, Pp 599-606, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.326.

[4] S. Shaikh, J. Gala, A. Jain, S. Advani, S. Jaidhara and M. Roja Edinburgh, "Analysis and Prediction of COVID-19 using Regression Models and Time Series Forecasting," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 989-995, doi: 10.1109/Confluence51648.2021.9377137.

[5] Nana, Dr & Kshirsagar, Dhananjay & Dange, Bapusaheb & Khodke, Dr & Kulkarni, Dr. (2022). Machine Learning Approach for Big-Mart Sales Prediction Framework. International Journal of Innovative Technology and Exploring Engineering. 11. 69-75. 10.35940/ijitee.F9916.0511622.

[6] Nayana R, Chaithanya G, Meghana T, Narahari K S, Sushma M, 2022, Predictive Analysis for Big Mart Sales using Machine Learning Algorithms, International Journal of Engineering Research & Technology (IJERT) RTCSIT – 2022 (Volume 10 – Issue 12)

[7] Varshini S., Dr. D. Preethi, "An Analysis of Machine Learning Algorithms to Predict Sales", International Journal of Science and Research (IJSR), Volume 11 Issue 6, June 2022, pp. 462-466, https://www.ijsr.net/get_abstract.php?paper_id=SR22601144946

[8] M. -C. Hsieh, A. Giloni and C. Hurvich, "The propagation and identification of ARMA demand under simple exponential smoothing: forecasting expertise and information sharing," in IMA Journal of Management Mathematics, vol. 31, no. 1, pp. 307-344, Dec. 2019, doi: 10.1093/imaman/dpaa006.

[9] X. Yuan, J. Yuan, T. Jiang and Q. U. Ain, "Integrated Long-Term Stock Selection Models Based on Feature Selection and Machine Learning Algorithms for China Stock Market," in IEEE Access, vol. 8, pp. 22672-22685, 2020, doi: 10.1109/ACCESS.2020.2969293.

[10] Hossein Abbasimehr, Mostafa Shabani, Mohsen Yousefi, An optimized model using LSTM network for demand forecasting, Computers & Industrial Engineering, Volume 143, 2020, 106435, ISSN 0360-8352, https://doi.org/10.1016/j.cie.2020.106435.