NORTHWESTERN UNIVERSITY | SCHOOL OF PROFESSIONAL STUDIES

Time Series Demand Forecasting Using LSTM

Yunze Chen, Yining Feng

November 22, 2020

Abstract

Lean manufacturing is an essential goal for manufacturing companies to sustain their competitive advantages. The demand forecast is the solution to achieve such a goal, following resource utilization optimization through better production planning. With the increasing availability of extensive amounts of historical data and the need to perform accurate demand forecasting, a powerful forecasting technique that infers the stochastic dependency between past and future values is highly needed. To help our client build a solution to forecast demand effectively, we developed four models to estimate demand based on historical patterns: a Long Short-Term Memory (LSTM) network, an encoder-decoder LSTM network, a hybrid neural network with a convolutional neural network (CNN) as the encoder and an LSTM network as the decoder, as well as a ConvLSTM Encoder-Decoder hybrid network. While there may be many available models for dealing with a time series problem, our multi-step time series forecasting approach is relatively new and highly sophisticated to its counterpart. By comparing this study to the other existing models and techniques, we are now closer to solving at least one of many complications apparent in the manufacturing industry.

1. Introduction & Problem Statement

In a manufacturing contract company, the most crucial aspect of the whole business process is cost management. The industry has brought a lot of measurements and tools to reduce the overall cost. One concept that is particularly important in today's industry is lean production. Through lean management, what adds value becomes clear by removing or reducing everything that does not add value. Eliminating waste is one of the strategies for removing non-value-added costs. In the manufacturing environment, waste can include overstaffing and overstock raw material. To reduce waste, the company should only hire a fair number of operators and keep a minimal material inventory level. Lean manufacturing brought up higher requirements for demand prediction. Better the prediction of demand, lower the possibility of waste in overpreparation. A research conducted by Retail Week revealed that Tesco achieved a 100 million Euro saving by a reduction in wastage stock (Clarke, 2013). These vast savings were made possible by getting experts in the retail space to work together with highly skilled data analysts to build models that help predict demand more precisely.

Traditional ways to perform demand forecasts include formulas inducted from experience and ML methods like ARIMA and SARIMA. Nevertheless, these traditional methods' performance is still questionable because more complex, high-dimensional, and noisy real-world time-series data often cannot be described with analytical equations based on parameters. With each traditional method having its limitation, we want to investigate the possibility of improving the demand forecast using a deep learning model. We want to use an LSTM based neural network to generate a model with the past three years of data to predict a target customer's demand. It is necessary to have the next week's prediction, next month, and the next 14 weeks from a business viewpoint. The next week is an immediate need; the next month is a reasonable planning period; the next 14 weeks is the strategic outlook.

2. Literature Review

The analysis and prediction of time series have always been a critical technique in supply chain analytics. Advanced data science techniques for predictive analytics in the manufacturing industry can be classified as traditional machine learning (ML) techniques and deep learning (DL) techniques. Many data-driven approaches have been applied in the existing literature toward time-series forecasting or classification, including autoregressive integrated moving average (ARIMA), support vector machines, statistical analysis, and instance-based learning techniques. The application of traditional ML techniques on large datasets consistently exposes these models' inherent vulnerabilities, such as the lack of structured means to determine some key parameters of the model, multicollinearity, and varying data aggregation.

On the other hand, DL techniques are increasingly employed in time series prediction. DL models, including recurrent neural networks (RNN), support vector machines (SVM), K-nearest neighbors (KNN), and adaptive neuro-fuzzy inference systems (ANFIS), have been frequently used for the problem of time series prediction. Specifically, Recurrent Neural Networks (RNNs), a deep learning model, establishes the reputation to cope with time series by recurrent neural connections. However, for any standard RNN architecture, the influence of a given input on the hidden layers and eventually on the neural network output would either decay or blow up exponentially when cycling around recurrent connections. For this problem, Long ShortTerm Memory (LSTM) has been exclusively designed by changing the hidden neurons' structure in traditional RNN. Today, research and applications of LSTM for time series prediction has been applied in various areas of application such as stock market prices forecasting (Cao, Li, & Li, 2019), anomaly detection (Filonov, Lavrentyev, & Vorontsov, 2016), tourism demand forecasting (Law, Li, Fong, & Han, 2019), forecasting of petroleum production (Sagheer & Kotb, 2019), weather forecasting (Chen, et al., 2015), etc.

Some recent studies have attempted to tackle demand forecasting problems using deep learning models. In the manufacturing industry, Ma et al. (2020) used long short-term memory models, which outperformed backpropagation neural networks, autoregressive moving average, and support vector regressions as the optimal approach for predictive production planning. In the transportation industry, Ke et al. (2017) proposed a novel deep learning approach of convolutional long short-term memory network (ConvLSTM) to forecast short-term passenger demand of on-demand ride service. The comparison results suggested that the ConvLSTM method provides better predictive performance than conventional time-series prediction models and state-of-art machine learning algorithms. In addition to travel demand forecasting, deep learning approaches have recently been used to predict market demand in the retail industry. Wang et al. (2018) innovatively integrated CNN and LSTM in one end-to-end DL structure, named the convolutional LSTM (ConvLSTM), which provided a brand-new idea predicting the market potential of areas without retailers in Guiyang, China.

Current literature revealed that LSTM networks typically outperform other conventional models in time series forecasting. The architecture of the LSTM network is generally composed of units called memory blocks. The memory block contains memory cells and three controlling gates, i.e., input gate, forget gate, and output gate. The memory cell is designed for preserving the knowledge of previous steps with self-connections that can store (remember) the temporal state of the network while the gates control the update and flow direction of information. The underlying mechanism behind the LSTM network mainly comes from the "gate" components that are designed for learning when to let prediction error in, and when to let it out (Lipton, Berkowitz, & Elkan, 2015). As such, LSTM gates provide an effective mechanism for quickly modifying the memory content of the cells and the internal dynamics in a cooperative way. In this sense, the LSTM network exhibits a superior ability to learn nonlinear statistical dependencies of real-world time series data compared to conventional forecasting models.

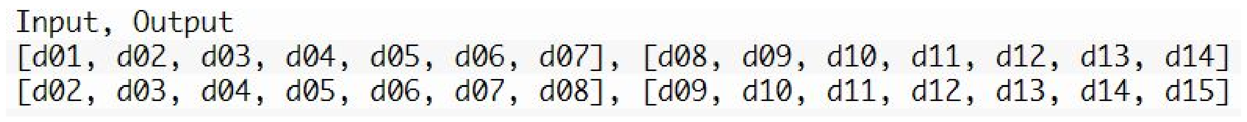
3. Research Design and Modeling Method

3.1 Data

The raw data comes from real order history from February 4th, 2017, to January 31st, 2020. There are 1,092 records in total. The dataset consists of dates in the year-month-day format and the order quantity of each date. The daily order quantity is always an integer value ranging from 9 to 77,812 units. An average daily order quantity is approximately 13,729 units. The *MinMaxScaler* function from the *scikit-learn* library is employed to normalize the input variable by transforming the input into the range [0,1], with the minimum and maximum value of the daily order variable as 0 and 1, respectively.

We also use Monte Carlo methods to generate multiple serialized data sets to simulate the noise for masking the original raw data. The goal of the Monte Carlo method is to approximate the expectations of outcomes given various inputs, uncertainty, and system dynamics. In this case, we generated normally distributed random numbers with a mean of 1 and a standard deviation of 0.1 in corresponding to each original data entry. This generated series is then multiplied to the original series to get one set of simulations. We run the Monte Carlo methods 100 times based on the original data, and in the end, we have 101 sets of series. That gives 108,500 records in total. For the next step, we create a function to iterate over the time steps and divide the data into an overlapping window of 7 days; the LSTM network's output from one time step is provided as an input in the subsequent time step. Figure 1 below shows examples of inputs and outputs as each iteration moves along one time step and predicts the subsequent seven days. We can do this by keeping track of start and end indexes for the inputs and outputs as we iterate across the length of the flattened data in terms of time steps.

Figure 1. Illustration of inputs and outputs of the LSTM network as each iteration moves along one time step and predicts the subsequent seven days.



```
Input, Output
[d01, d02, d03, d04, d05, d06, d07], [d08, d09, d10, d11, d12, d13, d14]
[d02, d03, d04, d05, d06, d07, d08], [d09, d10, d11, d12, d13, d14, d15]
```

3.2 Methods

In this study, we will employ four models utilizing LSTM for predicting the demand of a target customer based on the past three years of order data. The architecture of each model is illustrated in Diagram 1 in the Appendix. Input into the LSTM network involves a so-called *sequence length* parameter (i.e., the number of time steps) defined by the sample values over a finite time window. Thus, sequence length represents the change in the input vector over time; this is the time-series aspect of the input data. For time series analysis, there are three essential parameters, lookback, step, and delay. We want to try multiple combinations of parameters to get the best result. In this case, the lookback period should set to one week. The step is one day. As for the delay, we want to predict the next week.

We first developed a model with a single hidden LSTM layer with 200 nodes. The number of neurons in the hidden layer is unrelated to the number of time steps in the input sequences. The LSTM layer is followed by a fully-connected layer with 100 nodes that will interpret the features learned by the LSTM layer. Finally, an output layer will directly predict a vector with seven elements, one for each day in the output sequence.

The second model is an encoder-decoder LSTM model comprised of two sub-models: one called the encoder that reads the input sequences and compresses it to a fixed-length internal representation, and the other one called the decoder that interprets the internal representation and uses it to predict the output sequence. The encoder is an LSTM layer that consists of 200 neurons. It then connects to a layer of repeat vectors. This sequence of vectors will be presented to the LSTM decoder, which is also an LSTM hidden layer of 200 nodes. In the end, similar to our first model, we added a Dense layer of 100 nodes to interpret the features and finally linked to the Output layer. Such a Dense layer and Output layer exist in all four models.

The third model is a hybrid model using CNN and LSTM in an encoder-decoder architecture. We will define a simple but effective CNN architecture for the encoder that consists of two 1D convolutional layers followed by a Max Pooling layer, the results of which are then flattened. The first convolutional layer reads across the input sequence and projects the results onto feature maps. The second performs the same operation on the feature maps created by the first layer, attempting to amplify any salient features. We will use 64 feature maps per convolutional layer and read the input sequences with a kernel size of three time steps. The decoder remains the same as in previous models.

Last but not least, the fourth model, ConvLSTM, is an extension of the CNN-LSTM model, with convolutional structures in both the input-to-state and state-to-state transitions. Unlike previous models, the ConvLSTM model uses a convolutional layer directly as part of reading input into the LSTM units themselves. The input data of the ConvLSTM model is expected to have the shape of a 5D tensor, where each time step of data is defined as an image of *(row × columns)* data points. The ConvLSTM can then read across the time steps and perform the CNN process within each subsequence. By stacking the ConvLSTM encoder and the LSTM decoder structure, we can build an end-to-end trainable model for daily order forecast.

In all the LSTM layers mentioned above, we used Tanh as the activation function, Sigmoid as the recurrent activation function, zero dropouts, and enabled bias vector. With such parameters, we could utilize the CUDA and cuDNN GPU acceleration to reach better training performance.

We set the ratio of samples for training and testing to 70%:30%. On the other hand, we picked the mean squared error loss function because it is a good match for our chosen error metric of root-mean-square error (RMSE). We used the efficient Adam implementation of stochastic gradient descent and fit the model for 200 epochs with a batch size of 128.

Models are evaluated using a scheme called walk-forward validation. In this validation method, the model is required to make a one-week prediction, then the actual data for that week is made available to the model so that it can be used as the basis for predicting on the subsequent week. This method is realistic for how the model may be used in practice and beneficial to the models, allowing them to use the best available data.

4. Results

4.1 Overall Result

Root Mean Square Error (RMSE) is applied to estimate the prediction accuracy. RMSE measures the square root of the mean of the deviation squares, which quantifies the difference between the predicted values and the actual values. The ConvLSTM model performed the best overall because this neural network yielded the lowest RMSE value. However, there is a speed/accuracy tradeoff. While the RMSE of the encoder-decoder LSTM model is slightly larger than that of the ConvLSTM model, it takes 9 seconds to run one epoch for the ConvLSTM model, whereas it takes 3 seconds for the encoder-decoder LSTM to reach the same RMSE value of 0.001. Table 1 shows the comparison of the ConvLSTM with the other three models — baseline LSTM model, Encoder-decoder LSTM, and CNN-LSTM. The performance of these models are summarized as follows:

Table 1. Comparison of models' performance and running speed.

Models	Lowest RMSE Value	Running Time	Epochs to reach 0.001
Baseline LSTM Model	0.0075	< 1 second/epoch	N/A
Encoder-decoder LSTM	4.21×10^{-4}	3 seconds/epoch	37
CNN-LSTM	6.52×10^{-4}	2 seconds/epoch	87
ConvLSTM	3.04×10^{-4}	9 seconds/epoch	16

As Table 1 shows, encoder-decoder LSTM, CNN-LSTM, and ConvLSTM all outperform the baseline LSTM model in terms of prediction accuracy. While the baseline LSTM model can only yield an RMSE value as low as 0.0075, the other three variants of the neural network model can all generate prediction results with RMSE values below 0.001. The encoder-decoder LSTM model reached 0.001 with 37 epochs while CNN-LSTM reached the same RMSE with 87 epochs. ConvLSTM reaches the RMSE of 0.001 with only 16 epochs. Nevertheless, the time it takes to run an epoch for the baseline LSTM model is less than 1 second, far less than the time it takes for each of the other models to run one epoch. On the other hand, encoder-decoder LSTM models can reach 0.001 in 111 seconds while it takes the CNN-LSTM 174 seconds and ConvLSTM 144 seconds.

4.2 Analysis and Interpretation

When trying out different parameters with the models, we manipulated the number of Monte Carlo simulations, the number of epochs, batch size of each epoch, with or without scaling. As a result, the more Monte Carlo simulations, the better each epoch's result because the data size is dramatically increased. More epochs can also improve the final result. However, the training process will reach a point where further training becomes very inefficient. In this case, the point comes in when the models reach RMSE below 0.001. The batch size of each epoch can affect the speed of each epoch. The smaller batch size can cause the processor to take many batches to finish one epoch, while the larger batch size may cause that one batch to be too big to be fetched and processed. The batch size balances the processor's bandwidth and processing speed. Finally, scaling is very effective in this study case, with the records in a big range. Before scaling, the RMSE is in the range of millions, if not billions. Such RMSE is hardly a meaningful measurement for the models. The other side effect of not scaling is that the training process is prolonged.

The plot of RMSE of daily order prediction for each model is shown by Diagram 2, 3, 4, & 5, respectively, in the Appendix. Because our target customer has never placed any orders on Sundays, we exclude Sundays' prediction. The baseline LSTM model plot shows that Thursdays and Fridays are easier days to forecast than the other days. Similarly, the plot for the CNN-LSTM model also shows the same results as the baseline LSTM model. On the other hand, the encoder-decoder LSTM model's plot shows that Fridays are easier to forecast than the other days. Conversely, the ConvLSTM model plot shows that Thursdays are easier days to forecast than the other days. Predictions generated by all models unanimously show that Saturday at the end of a standard week is the most challenging day to forecast. In reality, our target customer usually places orders during weekdays and only residual orders on Saturdays, and therefore there could be a wide variation among order amounts on Saturdays.

To compare the four models horizontally with the result, we want to find the model with the best accuracy. Among all four models, the baseline LSTM model does not provide the desired accuracy while the CNN-LSTM model does reach the desired accuracy but both of its lowest accuracy and total time to reach the 0.001 RMSE are worse than the other two models. The encoder-decoder LSTM model is the most efficient as it has the lowest time consumption and second-lowest final accuracy. ConvLSTM model is the most accurate model as it has the lowest final accuracy. It also used the least epochs to reach the RMSE of 0.001. However, each epoch of the ConvLSTM model does take a longer time than all other models.

5. Conclusions

5.1 Conclusions

In the manufacturing production environment, the business aspect of this study, we would pick the ConvLSTM model to predict our demand forecast. ConvLSTM model is the most accurate model based on the result. Furthermore, in the enterprise environment, computational power is not the most concerning issue. On the other hand, the encode-decode LSTM model can also be an option for the business intelligence team to use as a quick prototype model as it is the most efficient one.

5.2 Directions for Future Work

To improve the model, we can try to get more historical data and run more Monte Carlo simulation iterations. We may also manipulate the LSTM layer in the model to improve accuracy. However, as cuDNN only supports limited types of LSTM, this direction may cause the training process to be slower.

From a business point of view, we only predicted a week's demand. We can also predict the next month and the next 14 weeks for the operations. In our prediction, we also need to inverse the transformation of the scaling. This way, the numbers regain their business meaning to be further used by the operations.

References

Brownlee, Jason. “Multi-Step LSTM Time Series Forecasting Models for Power Usage.” Machine Learning Mastery. Machine Learning Mastery Pty. Ltd., October 10, 2018. https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/

Cao, Jian, Zhi Li, and Jian Li. “Financial time series forecasting model based on CEEMDAN and LSTM.” *Physica A: Statistical Mechanics and its Applications* 519 (2019): 127-139.

Chen, Xingjian, Shi Zhourong, Hao Wang Dit-Yan Yeung, and Wai-Kin Wong Wang-chun Woo. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.” *arXiv preprint arXiv:1506.04214* (2015).

Clarke, Lindsay. “Analysis: How Tesco and Otto Are Using Data to Forecast Demand.” Retail Week. Ascential Information Services Limited, October 10, 2013. https://www.retail-week.com/analysis-how-tesco-and-otto-are-using-data-to-forecast-demand-/5053784.article.

Filonov, Pavel, Andrey Lavrentyev, and Artem Vorontsov. “Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model.” *arXiv preprint arXiv:1612.06676* (2016).

Ke, Jintao, Hongyu Zheng, Hai Yang, and Xiqun Michael Chen. “Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach.” *Transportation Research Part C: Emerging Technologies* 85 (2017): 591-608.

Law, Rob, Gang Li, Davis Ka Chio Fong, and Xin Han. "Tourism demand forecasting: A deep learning approach." *Annals of Tourism Research* 75 (2019): 410-423.

Lipton, Zachary C., John Berkowitz, and Charles Elkan. "A critical review of recurrent neural networks for sequence learning." *arXiv preprint arXiv:1506.00019* (2015).

Ma, Shuaiyin, Yingfeng Zhang, Jingxiang Lv, Yuntian Ge, Haidong Yang, and Lin Li. "Big data driven predictive production planning for energy-intensive manufacturing industries." *Energy* 211 (2020): 118320.

Sagheer, Alaa, and Mostafa Kotb. "Time series forecasting of petroleum production using deep LSTM recurrent networks." *Neurocomputing* 323 (2019): 203-213.

Wang, Luyao, Hong Fan, and Yankun Wang. "Sustainability analysis and market demand estimation in the retail industry through a convolutional neural network." *Sustainability* 10, no. 6 (2018): 1762.

Appendix

Figures

Diagram 1. Breakdown of four neural networks' architectures for demand prediction.

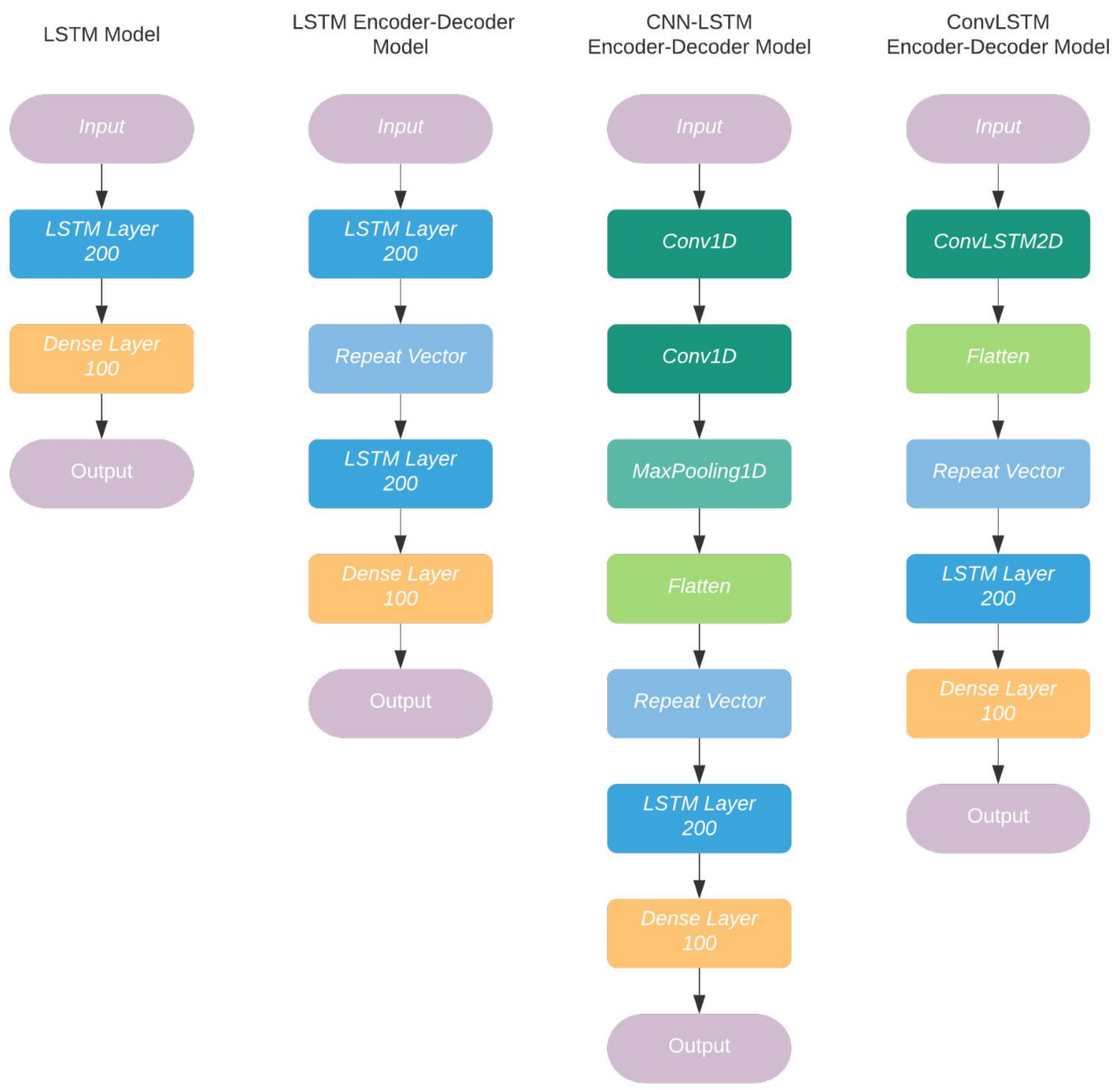


Diagram 2. The plot of RMSE of daily order prediction for the baseline LSTM model.

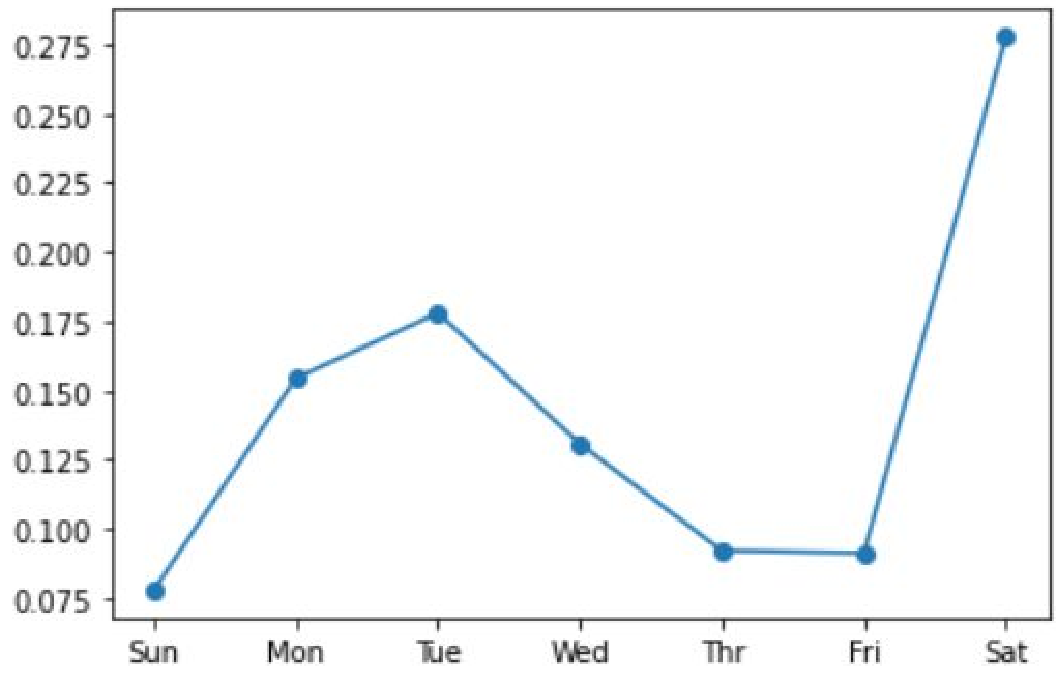


Diagram 3. The plot of RMSE of daily order prediction for the encoder-decoder LSTM model.

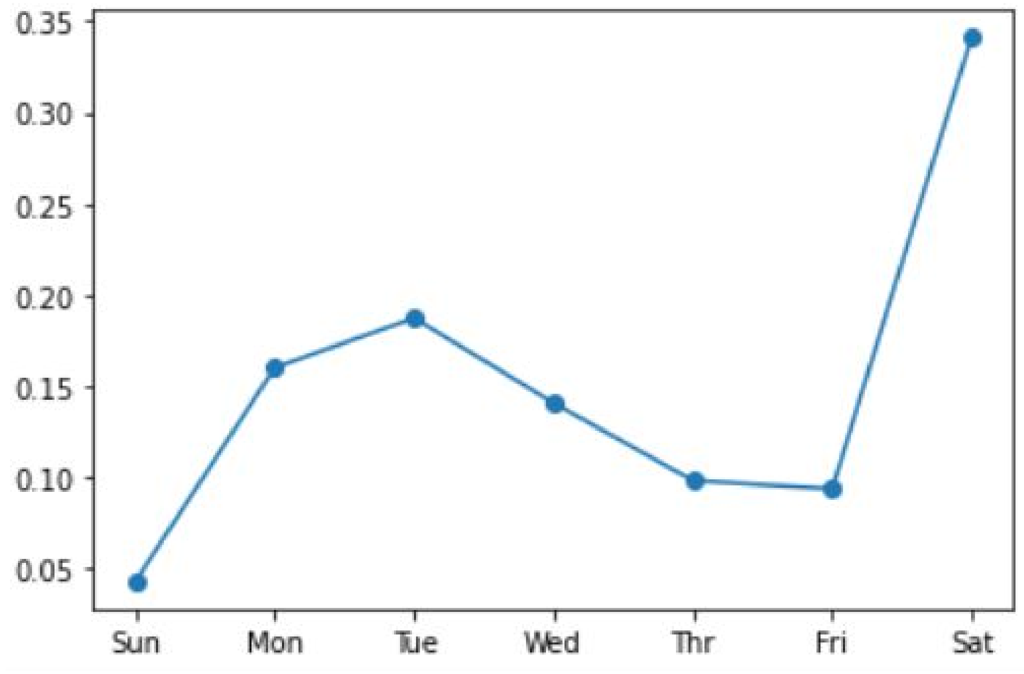


Diagram 4. The plot of RMSE of daily order prediction for the CNN-LSTM model.

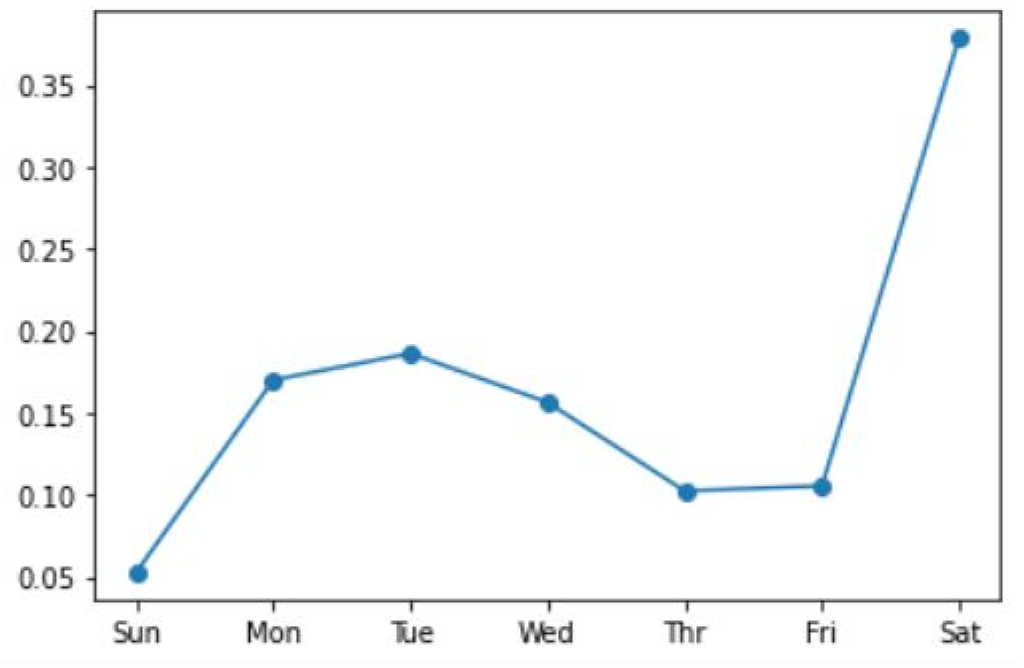
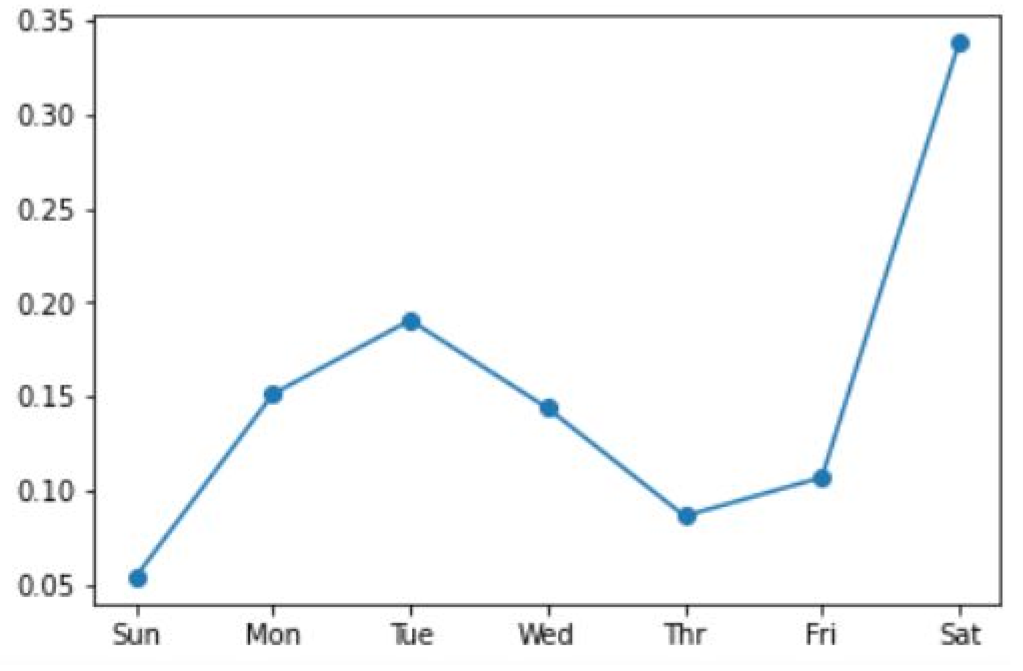


Diagram 5. The plot of RMSE of daily order prediction for the ConvLSTM model.


Sun
Mon
Tue
Wed
Thr
Fri
Sat