Retail Demand Forecasting Using Sequence to Sequence Long Short-Term Memory Networks

Mon Myat Phyu
University of Information Technology
Yangon, Myanmar
monmyatphyu@uit.edu.mm

Myat Thiri Khine
University of Information Technology
Yangon, Myanmar
myatthirikhine@uit.edu.mm

***Abstract*—Demand forecasting is crucial for a retail business as it can greatly affect everything ranging from promotion, pricing, product assortment and inventory. Building a reliable and useful demand forecasting model is still a challenging task. Machine learning techniques used for demand forecasting including Random Forest Regressor and Support Vector Regressor are inadequate when dealing with time series. Recent works show that Long Short-Term Memory (LSTM) networks can learn non-linear relationships and time-series specific information from retail time series data. In this paper, a methodology based on Sequence to Sequence Long Short-Term Memory (Seq2Seq LSTM) network is proposed to tackle short-term retail demand forecasting problem. The Seq2Seq architecture commonly used for language translation is adapted to retail demand forecasting to improve LSTM's ability of learning long-range temporal dependencies from retail time series data. Experiments are evaluated with different input sequence lengths on store item sales dataset with daily resolution data. Bayesian Optimization is conducted to tune models' hyperparameters and examine whether it could enhance the prediction accuracy of the models. In order to gauge the robustness of the proposed forecasting model, it is compared against Standard LSTM and Vanilla RNN.**

Keywords— retail demand forecasting, multi-step ahead forecasting, sequence to sequence long short-term memory, long short-term memory, recurrent neural network

I. Introduction

Retail demand forecasting can be defined as predicting future demand for a product or service by using its past demand and other influential factors that can affect demand. Alternatively, it can be considered as a kind of time series forecasting task. At an operational level, demand forecasting focuses on product-level forecasts, which is crucial for making operational decisions including inventory management, pricing, promotion and so on. Inaccurate demand forecasting may result in inventory surplus, lost sales, poor customer experience and revenue losses. Hence, it is important to improve the accuracy of demand forecasts.

For decades, many researchers have carried out different researches and studies to be able to produce more accurate forecasts. It is challenging to directly incorporate trend and seasonality, which usually contain in retail time series into a forecasting model. Traditional univariate forecasting techniques such as ARIMA do not perform well on non-linear time series data. More advanced machine learning techniques including Random Forest Regressor (RFR) and Support Vector Regressor (SVR) can handle non-linearity but they are incapable of learning time series-specific information [3].

Moreover, traditional Neural Networks are unable to persist information from one time step to another, which becomes a major shortcoming of Neural Networks when processing sequential data. Recurrent Neural Networks (RNNs) are getting more and more attention for solving time series forecasting problem in recent years. RNN has connections between nodes that constitute a directed graph along a temporal sequence and allow them to exhibit temporal dynamic behaviour of a sequence. They are well-suited to sequential data like time series data as they can preserve the information from previous time steps along with the current input when processing sequential data. However, the Vanilla RNN encounters the vanishing gradient problem, which makes it inadequate to persist long term dependencies.

Therefore, Long Short-Term Memory (LSTM), a special kind of RNN was introduced to solve the vanishing gradient problem of Vanilla RNN. It has the advantage of being able to persist long-range temporal dependencies by enforcing constant error flow through constant error carrousels (CECs) inside its memory cell [4]. Seq2Seq is a general framework but not a specific model [5]. Even though Seq2Seq models were initially used for language translation, they have also been successfully applied to multi-step ahead forecasting problem [6], [7], [8], [9] because their ability to map sequences of varying lengths is beneficial for multi-step ahead forecasting. Seq2Seq LSTM consists of two LSTM networks: one LSTM acts as an encoder, which maps the input sequence to a fixed-length vector and another LSTM acts as a decoder, which generates output sequence from that vector [10].

It is interesting to explore the suitableness of Seq2Seq LSTM to retail time series data. Hence, this paper proposes a methodology based on Sequence to Sequence LSTM (Seq2Seq LSTM) network for short-term retail demand forecasting. The Seq2Seq architecture, commonly used for language translation is adapted to retail demand forecasting to improve demand forecasting accuracy. This study focuses on forecasting demand of individual item for a week ahead. Firstly, Seq2Seq LSTM models with four different input sequence lengths are built using default hyperparameters. Then, the best combination of hyperparameters achieved from Bayesian Optimization is used instead of default hyperparameters to examine the predictive performance of different Seq2Seq LSTM models. Lastly, the best Seq2Seq LSTM model was compared with Standard LSTM and Vanilla RNN models. All experiments were conducted using store item sales dataset, which contains daily resolution data.

The rest of the paper is structured as follows. Section II introduces related work. Section III addresses the problem description. Section IV provides background theory. Section V describes the dataset and explains the pipeline of the proposed methodology in details. Section VI explains experiments and discusses experimental results. Section VII concludes the paper and directs for future work.

II. Related Work

Retail forecasting can be distinguished into strategic, tactical and operational forecasts [1]. A large amount of literature can be found under retail forecasting. Many researchers have proposed different approaches to deal with different kinds of retail forecasting problems. Data-driven

approaches including statistical methods, machine learning techniques, artificial neural networks and recurrent neural networks are commonly used in the literature because of the availability of large volumes of data.

Ching-Wu Chu and Guoqiang Peter Zhang [11] carried out a comparison of linear models and nonlinear models such as ARIMA and Neural Networks for aggregate retail sales forecasting. Their results suggested that the neural network model that used deseasonalized time series data performed the best. Real Carbonneau, Kevin Laframboise, Rustam Vahidov [12] investigated the relevance of advanced machine learning techniques for demand forecasting and compared against traditional techniques. They found that traditional techniques including Moving Average and Naïve did not perform well while advanced techniques such as Recurrent Neural Networks (RNN) and SVM provided the best performance.

Sen Lin, Eric Yu and Xiuzhen Guo [13] compared the performance of Frequency Domain Regression (FDR), SVR and Linear Regression for store sales forecasting. Their experiments showed that SVR was the highest performing method. Yuta Kaneko and Katsutoshi Yada [14] constructed a deep learning model for retail store sales prediction and deep learning model with L1 regularization outperformed logistic regression model. Oscar Chang, Ivan Naranjo, Christian Guerron, Dennys Criollo, Jefferson Guerron and Galo Mosquera [15] proposed a Deep Neural Network (DNN) that includes one autoencoder layer and two shallow nets for forecasting weekly sales of pharmaceutical products.

Adarsh Goyal et al. [5] built LSTM models for multi-step ahead demand forecasting using univariate demand time series data. The result showed that LSTM models achieved lower MAPE compared to other baseline models including Feedforward Neural Network and Exponential Smoothing models. Ajinkya Athlye and Angad Bashani [16] implemented different deep learning models for forecasting best-selling products. They also considered other external factors including holiday and temperature. Deep learning models, ANN, RNN, GRU and LSTM were compared against traditional models such as AR and ARIMA. Among all models, LSTM performed the best with the least Mean Absolute Percentage Error (MAPE).

Kasun Bandara et al. [2] built a global multivariate multi-step ahead demand forecasting model using LSTM network. Cross-series information of related demand time series was incorporated into the model to consider the non-linear relationships between products under e-commerce product assortment hierarchy. Their proposed method outperformed ETS and naive. Suleka Helmini, Nadheesh Jihan, Malith Jayasinghe and Srinath Perera (2019)[3] appraised an LSTM model for sales forecasting by incorporating features that can be known in the current moment. They claimed that the improved LSTM model had a significant improvement over the initial LSTM model and both LSTM models outperformed Extreme Gradient Boosting (XGB) and Random Forest Regressor (RFR).

This study mainly focuses on operational forecasts in retail forecasting, which consider product level forecasts for a short forecasting horizon. The success of LSTM in recent studies [2], [3], [5], [17] inspires us to use LSTM in our study. The proposed methodology mainly differs from the recent studies by adapting Seq2Seq architecture from language translation to retail demand forecasting and according to the overall methodology used to tackle short-term retail demand forecasting problem.

III. Problem Definition

This section defines short-term retail demand forecasting problem that will be tackled in this paper.

Suppose $X_i = \{x_1, x_2, \dots, x_{L-1}, x_L\} \in \mathbb{R}^L$ be daily sales time series of item i which has length L. $Z_i = \{z_1, z_2, \dots, z_{L-1}, z_L\} \in \mathbb{R}^{L \times D}$ be a set of additional features extracted which has D feature dimension. Given time t, n-step ahead forecasting of demand of an item using its past sales data and additional features can be written as follows.

$$y_{t+n} = f(\{x_{t-m+1}, \dots, x_{t-1}, x_t\}, \{z_{t-m+1}, \dots, z_{t-1}, z_t\}) \quad (1)$$

where f is a predictive model, n is the length of forecasting horizon and m is the length of past consecutive timesteps used for n-step ahead forecasting. As forecasting demand for a week ahead is considered here, n is equal to 7.

IV. Background

This section describes Long Short-Term Memory and Seq2Seq Long Short-Term Memory Networks.

A. Long Short-Term Memory

In order to overcome the vanishing gradient problem of Recurrent Neural Network (RNNs), Long Short-Term Memory (LSTM) was introduced. An LSTM network comprises memory cells illustrated in Fig. 1. Each memory cell is a recurrent unit itself which uses the input of the current time step x_t and the output from the previous time step h_{t-1}, the old cell state C_{t-1} along with three gates, namely, forget gate f_t, input gate i_t and output gate o_t to compute a new hidden output h_t and cell state C_t. The detailed computations of an LSTM unit are given as follows:

$$f_t = \sigma(W_f . x_t + U_f . h_{t-1} + b_f) \quad (2)$$

$$i_t = \sigma(W_i . x_t + U_i . h_{t-1} + b_i) \quad (3)$$

$$\tilde{C}_t = tanh(W_c . x_t + U_c . h_{t-1} + b_c) \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (5)$$

$$o_t = \sigma(W_o . x_t + U_o . h_{t-1} + b_o) \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7)$$

where W' s, U' s are weight matrices, and b' s are bias vectors, which are model parameters learned during training. σ is sigmoid activation function and $tanh$ is hyperbolic tangent activation function.$+$, $.$, $\odot$ denotes addition, inner product and Hadamard product or element-wise product, respectively.

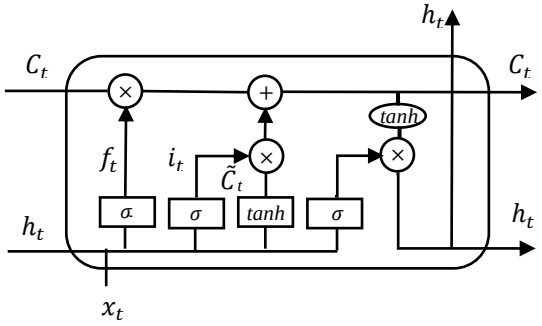

Fig. 1. *An LSTM Cell*

The forget gate layer decides that the information from previous cell state needs to be removed or not. Input gate layer determines what information will be added to the new cell state. The $tanh$ layer computes candidate values that will be added to the new cell state. The output layer calculates which parts of the new cell state will be produced as output h_t. The information flow is regulated inside the memory cell and relevant information is passed through longer sequences when processing sequential data. In such a way, LSTM can preserve long term dependencies between sequences.

B. Sequence to Sequence Long Short-Term Memory

As illustrated in Fig. 2, Seq2Seq LSTM constitutes encoder LSTM, encoder state vector and decoder LSTM. The first component, encoder LSTM processes the input sequences, learns an internal representation of the input sequences and generates a state vector. Then, the state vector is utilized as an initial state of decoder LSTM to produce the output sequences.

The combination of encoder LSTM and decoder LSTM increases LSTM's ability of learning long-range temporal dependencies between input and output sequences. Additionally, Seq2Seq architecture provides LSTM to be more flexible while mapping an arbitrary length of input sequences to an arbitrary length of output sequences.

V. METHODOLOGY

This section describes the description of the dataset and explains the pipeline of the proposed methodology illustrated in Fig. 3 in details.

A. Dataset Description

The dataset is comprised of 500 daily time series. Each time series is 5 years' worth sales of an item from January 2013 to December 2017. It contains 4 variables namely, date, item, store and sales. Date variable describes date being observed. Item and store variables are unique identifiers of each item and store, respectively. Sales variable represents quantities sold of each item in each store on a specific day. For the implementation and evaluation purpose, an item among 500 different items was randomly chosen. Daily sales demand for the chosen item over five years is shown in Fig. 4. There is neither holiday effects nor store closures. It can be seen that the series has upward trend and distinct seasonality.

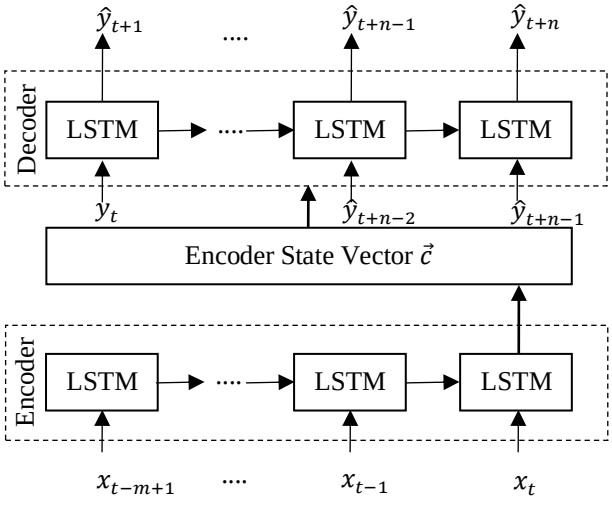

Fig. 2. *Sequence to Sequence Long Short-Term Memory*

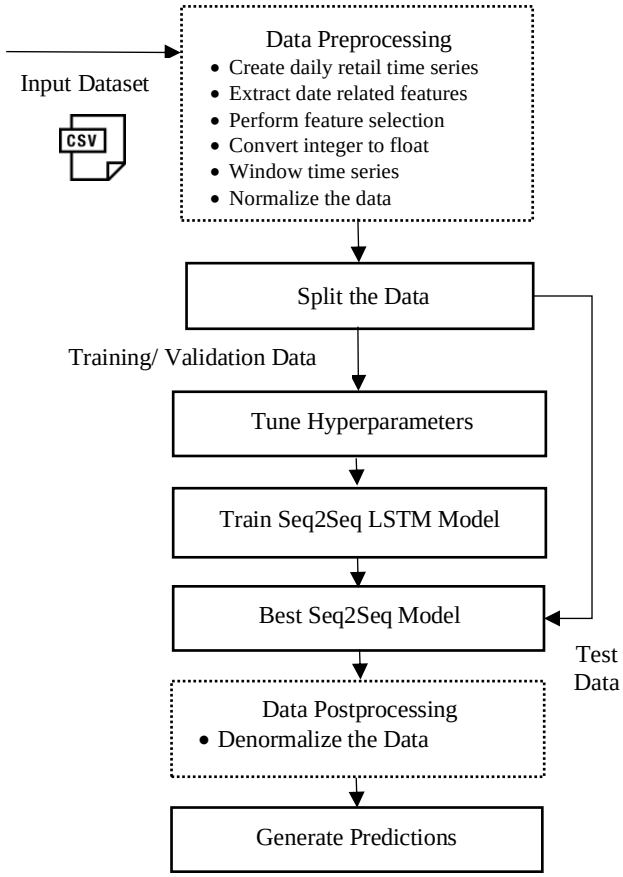

Fig. 3. *The pipeline of the proposed methodology*

The dataset is collected from [18].

B. Data Preprocessing

Firstly, a daily retail time series for the selected item was created by grouping the data by store and item. It contains 1826 data points. In addition to sales features, features that may be helpful in learning the behaviour of retail time series such as year, quarter, week of year, month of year, day of week and day of month were extracted from date variable. Among all features, it is observed that only quarter, month of year, day of week and sales features have contributions to the forecasting accuracy according to empirical analysis. Thus, day of month, week of year and year features were omitted since adding them can only increase model complexity. After that, the values of all features were converted into float data types since neural networks fit better with floating-point numbers.

The data was then transformed into encoder input, decoder input and decoder output sequences to train Seq2Seq LSTM network in a supervised fashion. Given a time series $X_i = \{x_1, x_2, \dots, x_L\} \epsilon \mathbb{R}^{L \times D}$, where L is the length of the time series and D is its dimension, the data at time step t is defined as:

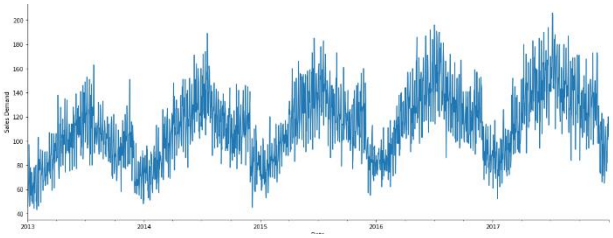

Fig. 4. *Sales Demand of the Selected Item over Five Years*

$$x_t = [day\ of\ week_t, month\ of\ year_t, quarter_t, sales_t] \quad (8)$$

Sliding window approach was applied to generate encoder input and decoder output sequences. It transforms X_i into pairs of input sequence which has input window size m and output sequences which has output window size n. After windowing, $(L - m - n + 1)$ samples were obtained. The input sequences were taken as encoder input sequences. As for decoder output sequences, only sales feature of the output sequences was kept and defined output window size n as 7 since this study aims at predicting demand of an item for a week ahead. In order to get decoder inputs, the last time step of encoder input sequences (only sales feature) and all decoder output sequences except the last time step were concatenated.

Then, min-max normalization [19], which can be calculated as describes in (9), was employed to scale each feature values into values ranging from 0 and 1. It can prevent large-scale feature such as sales from dominating small-scale feature like day of week.

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (9)$$

where $x_{normalized}$ is the normalized feature and x_{min}, x_{max} are the minimum and maximum values of the normalized feature, respectively.

C. Data Splitting

After the data was preprocessed, it was split into three sets: 80 percent for training, 10 percent for validation and the remaining 10 percent for testing. The training set was used for training the model. The validation set, which is also known as the development set was used for doing an unbiased evaluation of a model fit on the training set while training the model and tuning the hyperparameters of the model. Lastly, the test set was used for evaluating a completely trained model.

D. Model Implementation

In this study, a simple Seq2Seq LSTM architecture as shown in Fig. 2 was used. The encoder LSTM layer was provided with the preprocessed encoder inputs for m previous time steps $x_{t-m+1}, \dots, x_t$. Each cell takes encoder inputs, one step at a time, previous cell state and hidden state and processes them as discussed in section IV. After the encoder LSTM processes the whole input sequence, it produces an encoder state vector $\vec{c}$ which is also known as context vector as it contains encoded context from the whole input sequence. The decoder LSTM layer uses that encoder state vector as its initial state and decoder input y_t as its initial input. It is followed by a dense layer with sigmoid activation to transform the hidden output produced from decoder LSTM to a prediction for the next time step $\hat{y}_{t+1}$. This whole process is iterated for n times to generate $\hat{y}_{t+1}, \dots, \hat{y}_{t+n-1}, \hat{y}_{t+n}$ which are demand predictions of the item for a week ahead. The preprocessed decoder inputs $y_t, \dots, y_{t+n-1}, y_{t+n-1}$ were fed, one step at a time during training the network, instead of feeding predicted decoder outputs $\hat{y}_{t+1}, \dots, \hat{y}_{t+n-1}, \hat{y}_{t+n-1}$ by following teacher forcing technique. However, during inference time, decoder inputs except y_t were replaced by predicted decoder outputs.

Mean Squared Error (MSE) was used as the loss function which computes the average squared difference of actual values and predicted values. The objective of training the network is to minimize the loss function after each epoch which can be expressed as:

$$\underset{\theta}{\operatorname{argmin}}\, L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (10)$$

Standard backpropagation through time was applied to train the network using Adam [20], a gradient-based optimization algorithm. Adam was used as optimizer as it performs better than other stochastic optimization methods with faster convergence and better accuracy. Model parameters θ, weight matrices and bias vectors, were eventually updated during the training process.

Early stopping strategy was also applied to prevent the network from overtraining that may lead to overfitting the training dataset. Once the training process was completed, the final model was evaluated on the test set.

E. Data Postprocessing

The predictions produced from the sigmoid layer are still the scaled values. In order to get its original scales, the predictions were denormalized by using inverse transform method [19].

VI. Experimental and Results

This section describes experiments conducted to evaluate our proposed methodology for short-term retail demand forecasting and discusses the results.

A. Experiments

TensorFlow [21], an open-source machine learning library, was used to implement the models. The proposed methodology was evaluated on five years' worth of sales data for the selected item described in Section V, which contains 1826 consecutive daily sales data. When it comes to choosing the length of input sequence m for forecasting a week ahead, there can be many possible values. Here, only four different input sequence lengths $m = [14, 28, 56, 112]$ were considered, which are equal to looking back to the past 14, 28, 56 and 112 consecutive days, respectively, in order to observe that if longer input sequence length could increase the accuracy of the model.

The default hyperparameters used to train those four Seq2Seq LSTM models were as follows:

- LSTM's cell and hidden dimension: 64
- Learning rate: 0.001
- Batch size: 32
- Epochs: 200

1) Hyperparameter Tuning: Hyperparameters play a crucial role as it controls a training algorithm's behaviour during the training process. The performance of a model can be greatly affected by the choice of hyperparameters. Here, Bayesian Optimization using Gaussian Process was employed to tune the hyperparameters of the model automatically and find out whether the best combination of hyperparameters provided by Bayesian Optimization can give higher accuracy than the default hyperparameters. Bayesian Optimization constructs a surrogate model that approximates the true objective function based on accumulated observations. Then, it uses an acquisition function to find the next point to evaluate from potentially promising region. Bayesian Optimization was chosen because it is computationally efficient than other exhaustive hyperparameter optimization techniques including Random

Search and Grid Search [3]. Hyperparameters tuned and their search spaces for Seq2Seq Models are shown in Table I. The default hyperparameters were set as an initial point and skopt.gp-minimize [22] was executed for 100 times. The objective is to minimize MSE values on the validation set. The expected improvement (EI) was used as the acquisition function. The best combinations achieved for Seq2Seq models with different input sequence lengths using Bayesian Optimization are described in Table II.

The experiments were also conducted on Vanilla RNN and Standard LSTM to examine the robustness of the best performing Seq2Seq LSTM model. Both Vanilla RNN and Standard LSTM took encoder inputs and decoder outputs as their input and target sequences. The input sequences were passed through one RNN layer and LSTM layer, respectively. And then each layer is followed by a dense layer with n nodes to predict n-step ahead directly instead of iteratively producing predictions, one step at a time as in Seq2Seq LSTM model, following multi-input multi-ouput strategy. The same default hyperparameters (cell dimension was excluded in Vanilla RNN) as Seq2Seq LSTM were used and Bayesian Optimization was also employed for both Vanilla RNN and Seq2Seq LSTM models.

2) Error Measure: Root Mean Square Error (RMSE) was used as a performance matrix. The overall RMSE for a week ahead predictions was computed as follows:

$$Overall\ RMSE = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{n}(y_{ij}-\hat{y}_{ij})^2}{N\times n}} \quad (11)$$

where y is actual sales, $\hat{y}$ is predicted demand, N is number of samples in the test set, and n is the length of the forecasting horizon. RMSE values for each day were also calculated to gauge the predictive performance of the model at each time step while forecasting for multi-step ahead. The RMSE value for the j^{th} day was calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_{ij}-\hat{y}_{ij})^2}{N}} \quad (12)$$

The RMSE values were calculated on the original scale of sales demand rather than on the scaled values. The lower the RMSE values are, the better the predictive performance of the model is.

TABLE I. HYPERPARAMETERS AND THEIR SEARCH SPACES FOR SEQ2SEQ LSTM MODEL

Hyperparameter	Search Space
LSTM's cell and hidden dimension	32-128
Learning rate	0.0001-0.01
Batch size	16-128
Epoch	100-200

TABLE II. BEST COMBINATION OF HYPERPARAMETERS ACHIEVED USING BAYESIAN OPTIMIZATION

Seq2Seq Model with Input Sequence Length (m)	Best Combination
14	[33, 0.0030, 16, 103]
28	[127, 0.0017, 126,193]
56	[35, 0.0023, 126, 198]
112	[126, 0.0017, 20, 102]

B. Results and Discussion

Table III shows the best achieved overall RMSE values of 8 different Seq2Seq LSTM models with four different input sequence lengths and two different hyperparameter sets. The lowest overall RMSE achieved is shown in bold. Considering Seq2Seq LSTM models with default hyperparameters, the model with the longest input sequence, m = 112, achieves the lowest overall RMSE and the values of RMSE decreases as the input sequence length increases. Therefore, it can be said that the longer input sequences can provide better accuracy. Besides, it is observed that models with best combination of hyperparameters from Bayesian Optimization show improvements over models with default hyperparameters for input sequence lengths, m = 14 and m = 56. However, RMSE values increase for input sequence length, m = 112 and m = 28 which means that the best hyperparameter combination achieved from Bayesian Optimization does not always enhance the prediction accuracy of the model. This may be due to the configuration settings we used for Bayesian Optimization. Different configurations of Bayesian Optimization will still need to be explored.

Input sequence length, m = 112, was used for Vanilla RNN and Standard LSTM models as it is the input sequence length that achieved the best performance for Seq2Seq LSTM model. Two different models were developed for both Vanilla RNN and Standard LSTM: one with the default hyperparameters and the other with Bayesian's best combination of hyperparameters for respective models. Table IV displays the best achieved overall RMSE for Seq2Seq LSTM, Standard LSTM and Vanilla RNN models. Comparing two LSTM models against Vanilla RNN model, it is found that both Seq2Seq LSTM and Standard LSTM models obtain significantly better RMSE values than Vanilla RNN as expected, with 11.44% and 9.12% decrease in RMSE since Vanilla RNN has difficulties in learning long-term dependencies between sequences. RMSE decrease in percentage can be obtained by calculating the difference between the overall RMSE of Vanilla RNN and Seq2Seq LSTM or Standard LSTM which is then divided by overall RMSE of RNN and multiply by 100.

Considering the two LSTM models, the performance of Seq2Seq LSTM is improved, with 2.55% reduction in RMSE compared to Standard LSTM which proves that the use of

TABLE III. OVERALL RMSE VALUES OF SEQ2SEQ LSTM MODELS

m	Overall RMSE	
	Default Hyperparameters	***Bayesian's Best Combination***
14	15.9255	15.0489
28	15.2429	15.2585
56	14.1367	13.6372
112	**13.3393**	15.4029

TABLE IV. BEST ACHIEVED OVERALL RMSE VALUES OF SEQ2SEQ LSTM, STANDARD LSTM AND VANILLA RNN

Model	Overall RMSE
Seq2Seq LSTM	13.3393
Standard LSTM	13.6878
Vanilla RNN	15.0616

encoder LSTM and decoder LSTM improves LSTM's ability of learning long-range temporal dependencies from retail time series data. A visualization of RMSE values for each day for Seq2Seq LSTM, Standard LSTM and Vanilla RNN models for a week ahead prediction is depicted in Fig. 5. It can be seen that RMSE values increase from time step 1 to 7 for all models, which means that the predictive performance of all models decreases as the forecasting horizon expands. Among all three models, Vanilla RNN performs the worst at each time step. Even though Seq2Seq LSTM uses iterative strategy for multi-step ahead forecasting, which has the drawbacks of error accumulations, it performs slightly better than Standard LSTM, which use multi-input multi-output strategy, at each time step for short-term retail demand forecasting.

VII. Conclusion and Future Work

Demand forecasting is crucial for retail business because it can help the business to be more cost-efficient and improve customer experience. Retail demand forecasting task may be a bit different depending on how demand forecasts will be used. In this study, we focus on short-term retail demand forecasting. A methodology based on Seq2Seq LSTM network is proposed by adapting Seq2Seq LSTM from language translation to retail demand forecasting to improve demand forecasting accuracy. Seq2Seq LSTM can not only improve the ability of LSTM while learning long-range temporal dependencies from retail time series data but also provide the flexibility in modelling multi-step ahead forecasting. The proposed methodology was evaluated with different input sequence lengths and different hyperparameter sets on store item sales dataset with daily resolution data and compared against Standard LSTM and Vanilla RNN models. It is found that longer input sequences provide better accuracy but the predictive performance of the models does not always increase by using the best combination of hyperparameters achieved from Bayesian Optimization. Additionally, the proposed Seq2Seq LSTM model outperforms Standard LSTM model and Vanilla RNN models in our short-term retail demand forecasting task.

In this study, we do not consider external factors that can affect demand such as retail price, promotion, weather and microeconomic indicators, etc. As future work, it would be interesting to incorporate that information into Seq2Seq LSTM network. Moreover, deeper architectures and different learning schemes of Seq2Seq LSTM network for multi-step ahead forecasting will still need to be investigated. Application of Seq2Seq LSTM to other time series forecasting problems and real-world applications will be explored as well.

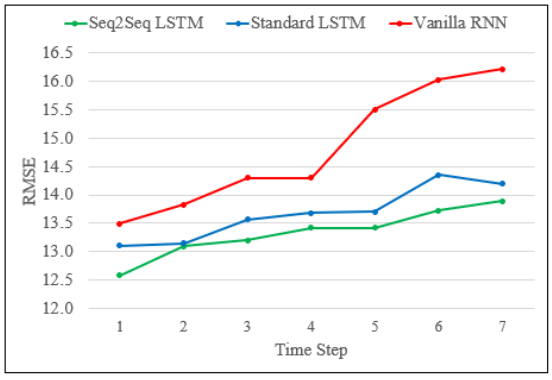

Fig. 5. *RMSE Values for Each Day and for Each Model*

References

[1] S. Ma, R. Fildes and S. Kolassa, "Retail forecasting: research and practice," MPRA Paper 89356, University Library of Munich, Germany, 2019.

[2] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran and B. Seaman, "Sales Demand Forecast in E-commerce Using a Long Short-Term Memory Neural Network Methodology," Neural Information Processing. ICONIP 2019, https://doi.org/10.1007/978-3-030-36718-339.

[3] S. Helmini, N. Jihan, M. Jayasinghe and S. Perera, "Sales forecasting using multivariate long short term memory network models," PeerJ Preprint, 7, e27712, 2019.

[4] F. A. Gers, J. Schmidhuber and F. Cummins, "Learning to forget: continual prediction with LSTM," 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), Edinburgh, UK, 1999, pp. 850-855 vol.2, doi: 10.1049/cp:19991218.

[5] A. Goyal, S. Krishnamurthy, S. Kulkarni, R. Kumar, M. Vartak and M. Lanham, "A Solution to Forecast Demand Using Long Short-Term Memory Recurrent Neural Networks for Time Series Forecasting," 2018.

[6] D. L. Marino, K. Amarasinghe and M. Manic, "Building energy load forecasting using Deep Neural Networks," IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, 2016, pp. 7046-7051, doi: 10.1109/IECON.2016.7793413.

[7] S. Du, T. Li and S. Horng, "Time Series Forecasting Using Sequence-to-Sequence Deep Learning Framework," 2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), Taipei, Taiwan, 2018, pp. 171-176, doi: 10.1109/PAAP.2018.00037.

[8] L. Sehovac, C. Nesen and K. Grolinger, "Forecasting Building Energy Consumption with Deep Learning: A Sequence to Sequence Approach," 2019 IEEE International Congress on Internet of Things (ICIOT), Milan, Italy, 2019, pp. 108-116, doi: 10.1109/ICIOT.2019.00029.

[9] L. Sehovac and K. Grolinger, "Deep Learning for Load Forecasting: Sequence to Sequence Recurrent Neural Networks With Attention," in IEEE Access, vol. 8, pp. 36411-36426, 2020, doi: 10.1109/ACCESS.2020.2975738.

[10] I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks," In Proc. of Advances in Neural Information Processing Systems (NIPS), pp. 3104–3112, 2014.

[11] C. Chua and G. P. Zhangb, "A comparative study oflinear and nonlinear models for aggregate retail sales forecasting, " Int. J. Production Economics 86 (2003), 217–231.

[12] R. Carbonneau, K. Laframboise and R. Vahidov. "Application of machine learning techniques for supply chain demand forecasting." Eur. J. Oper. Res. 184 (2008): 1140-1154.

[13] S. Lin, E. Yu and X. Guo. "Forecasting Rossmann Store Leading 6-month Sales CS 229 Fall 2015." (2015).

[14] Y. Kaneko and K. Yada, "A Deep Learning Approach for the Prediction of Retail Store Sales," 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, 2016, pp. 531-537, doi: 10.1109/ICDMW.2016.0082.

[15] O. Chang, I. Naranjo, C. Guerron, D. Criollo, J. Guerron, G. Mosquera, "A Deep Learning Algorithm to Forecast Sales of Pharmaceutical Products," 2017.

[16] A. Athlye and A. Bashani, "Multivariate Demand Forecasting of Sales Data," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 6, no. 10, pp. 198-211, Oct 2018.

[17] S. Du, T. Li and S. Horng, "Time Series Forecasting Using Sequence-to-Sequence Deep Learning Framework," 2018 9th International dsSymposium on Parallel Architectures, Algorithms and Programming (PAAP), Taipei, Taiwan, 2018, pp. 171-176, doi: 10.1109/PAAP.2018.00037.

[18] "Store Item Demand Forecasting," Available at https://www.kaggle.com/c/demand-forecasting-kernelsonly/data.

[19] "Min Max Scaler," Available at https://scikit-learn.org/.

[20] D. P. Kingma and J. Ba, "Adam: a method for stochasticoptimization," arXiv:1412.6980, 2014.

[21] "TensorFlow," Available at https://www.tensorflow.org/.

[22] "Scikit-Optimize," Available at https://scikitoptimize.github.io/stable/.