# Sales forecasting using multivariate long short term memory network models

**Suleka Helmini** [1] , **Nadheesh Jihan** [1] , **Malith Jayasinghe** [Corresp., 1] , **Srinath Perera** [2]

[1] WSO2 Research, WSO2, Colombo, Sri Lanka

[2] CTO Office, WSO2, Colombo 03, Sri Lanka

Corresponding Author: Malith Jayasinghe
Email address: malithj@wso2.com

In the retail domain, estimating the sales before actual sales become known plays a key role in maintaining a successful business. This is due to the fact that most crucial decisions are bound to be based on these forecasts. Statistical sales forecasting models like ARIMA (Auto-Regressive Integrated Moving Average), can be identified as one of the most traditional and commonly used forecasting methodologies. Even though these models are capable of producing satisfactory forecasts for linear time series data they are not suitable for analyzing non-linear data. Therefore, machine learning models (such as Random Forest Regression, XGBoost) have been employed frequently as they were able to achieve better results using non-linear data. The recent research shows that deep learning models (e.g. recurrent neural networks) can provide higher accuracy in predictions compared to machine learning models due to their ability to persist information and identify temporal relationships. In this paper, we adopt a special variant of Long Short Term Memory (LSTM) network called LSTM model with peephole connections for sales prediction. We first build our model using historical features for sales forecasting. We compare the results of this initial LSTM model with multiple machine learning models, namely, the Extreme Gradient Boosting model (XGB) and Random Forest Regressor model(RFR). We further improve the prediction accuracy of the initial model by incorporating features that describe the future that is known to us in the current moment, an approach that has not been explored in previous state-of-the-art LSTM based forecasting models. The initial LSTM model we develop outperforms the machine learning models achieving 12% - 14% improvement whereas the improved LSTM model achieves 11\% - 13\% improvement compared to the improved machine learning models. Furthermore, we also show that our improved LSTM model can obtain a 20% - 21% improvement compared to the initial LSTM model, achieving significant improvement.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

# Sales Forecasting using Multivariate Long Short Term Memory Networks

**Suleka Helmini[1], Nadheesh Jihan[1], Malith Jayasinghe[1], and Srinath Perera[1]**

[1]**WSO2, Inc., Mountain View, CA, USA**

Corresponding author:
First Author[1]

Email address: malithj@wso2.com

## ABSTRACT

In the retail domain, estimating sales before actual sales become known plays a key role in maintaining a successful business. This is due to the fact that most crucial decisions are bound to be based on these forecasts. Statistical sales forecasting models like ARIMA (Auto-Regressive Integrated Moving Average), can be identified as one of the most traditional and commonly used forecasting methodologies. Even though these models are capable of producing satisfactory forecasts for linear time series data they are not suitable for analyzing non-linear data. Therefore, machine learning models (such as Random Forest Regression, Extreme Gradient Boosting) have been employed frequently as they were able to achieve better results using non-linear data. The recent research shows that deep learning models (e.g. recurrent neural networks) can provide higher accuracy in predictions compared to machine learning models due to their ability to persist information and identify temporal relationships. In this paper, we adopt a special variant of Long Short Term Memory (LSTM) network; LSTM with peephole connections for the sales forecasting tasks. We first introduce an LSTM model that solely depends on historical information for sales forecasting. We appraise the accuracy of this initial LSTM against two state-of-the-art machine learning techniques, namely, Extreme Gradient Boosting (XGB) and Random Forest Regressor (RFR) using 8 randomly chosen stores from the Rossmann data-set. We further improve the prediction accuracy of the initial LSTM model by incorporating features that describe the future that is known to us in the current moment, an approach that has not been explored in previous state-of-the-art LSTM based forecasting models. The initial LSTM we develop outperforms the two regression techniques achieving 12% - 14% improvement whereas the improved LSTM achieves 11% - 13% reduction in error compared to the machine learning approaches with the same level of information as the improved LSTM. Furthermore, using the information describing the future with the LSTM model, we achieve a significant improvement of 20% - 21% compared to the LSTM that only uses historical data.

## INTRODUCTION

Time series forecasting involves performing forecasts on data with a time component. Forecasting typically considers historical data and provides estimations based on them for the future. Sales forecasting is a time series forecasting task. It is the process of predicting future sales values. In the retail domain, estimating sales before actual sales become known plays a key role in maintaining a successful business. This is due to the fact that most crucial decisions are bound to be based on these forecasts. Before technology dominated the world, the forecasting process was done manually by an experienced individual in the domain. This intuition required a lot of experience and was prone to human error. Due to this reason, individuals started realizing the need for automating the sales forecasting process. Thus, research and experiments were carried out with statistical, machine learning, deep learning and ensemble techniques to achieve more accurate sales forecasts.

Statistical sales forecasting models like Auto-Regressive Integrated Moving Average (ARIMA), can be identified as one of the most traditional and commonly used forecasting methodologies. Even though these models are capable of producing satisfactory forecasts for linear time series data they are not suitable for analyzing non-linear data (Zhang, 2003). Therefore, machine learning models were employed frequently as they were able to achieve better results using non-linear data. The use of state-of-the-art

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

machine learning models like Support Vector Regression (SVR), Extreme Gradient Boosting (XGB) and Random Forest Regressor (RFR) can be seen in the literature. Though the behaviour of SVR models with sales forecasting has been studied extensively (Carbonneau et al., 2008; Xiangsheng Xie, 2008; Gao et al., 2009) analysis on XGB and RFR model's behaviour is not as common. However, even though machine learning models are capable of handling non-linear information they are not tailored towards capturing time series specific information.

In recent years, types of Recurrent Neural Networks (RNN) have been frequently employed for sales forecasting tasks and have shown promising results (Bandara et al., 2019; Chniti et al., 2017; Carbonneau et al., 2008). This is mainly due to RNNs having the ability to persist information about previous time steps and being able to use that information when processing the current time step. When performing a time series forecasting task, it is important to remember what the model saw in the previous time steps when processing the current data in order to capture the complex correlations and patterns. Furthermore, compared to other sales forecasting methods, using RNNs eliminate the need to perform manual traditional modelling methods like stability checking, auto-correlation function checking and partial auto-correlation function checking, thus simplifying the modelling process (Yunpeng et al., 2017). Müller-Navarra et al. (2015) proposes neural network architectures for sales forecasting of a real-world sales data-set and empirically proves that partial recurrent neural networks can outperform statistical models. Carbonneau et al. (2008) have used RNN and SVM for demand-forecasting and achieve higher accuracy compared to conventional regression techniques. Although the basic RNN architecture can persist short term dependencies due to it being prone to vanishing gradients it is unable to persist long term dependencies. Long Short Term Memory (LSTM) network is a type of RNN that was introduced to persist long term dependencies. This helps in persisting information of many previous time-steps and allow to derive correlations from the information of older time-steps compared to a traditional RNN. It is evident that LSTM networks have often been used in identifying correlations between cross series Bandara et al. (2019); Chniti et al. (2017). Recently, it has been shown that multivariate LSTM with cross-series features can outperform the univariate models for similar time series forecasting tasks. Chniti et al. (2017) propose to forecast the prices of mobile phones while considering the correlations between the prices of different phone models by multiple providers in the cell phone market, as a cross-series multivariate analysis. Their technique achieves a significant accuracy gain compared to an SVR model that uses the same information as lag features. Bandara et al. (2019) proposes a similar multivariate approach, they have used cross-series sales information of different products to train a global LSTM model to exploit demand pattern correlations of those products.

In this paper we adopt a special variant of LSTM called "LSTM with peephole connections" (Lipton, 2015; Gers et al., 1999) that can more accurately capture the time-based patterns in sales forecasting tasks. We first present a multivariate LSTM model (based on peephole connections) in which we use historical features for sales forecasting with daily sales values. We compare the results of this initial LSTM model with multiple machine learning models, namely, XGB and RFR. We then further improve the prediction accuracy of the initial model by incorporating features that describe the future that is known to us in the current moment, an approach that has not been explored in previous state-of-the-art LSTM based forecasting models. These new features were added in addition to the historical information and daily sales values. Similar to the initial model, we compare the results of the improved LSTM model with the improved machine learning models and ultimately analyze how the improved LSTM performed compared to the initial LSTM model. The initial LSTM model that we developed outperformed machine learning models achieving 12% - 14% improvement whereas the improved LSTM model achieved 11% - 13% improvement compared to the improved machine learning models. Furthermore, we also show that our improved LSTM model can obtain a 20% - 21% improvement compared to the initial LSTM model, achieving significant improvement.

In order to evaluate the forecasting accuracy of the models, we used the Rossmann data-set [1]. It can be seen that the Rossmann data-set has been used frequently for sales forecasting in numerous occasions (Lin et al., 2015; Pavlyshenko, 2016; Doornik and Hansen, 1994) Rossmann is a company that governs over 3000 drug stores in 7 European countries and this data-set contains sales information of 1,115 stores located across Germany. The data-set offers convoluted sales patterns and also offers many different unique features of stores like competition distance, promotion interval and competition open since a month which facilitates in exploring novel forecasting methodologies. All stores in the data-set were

[1] https://www.kaggle.com/c/rossmann-store-sales


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

divided into 4 types. In our analysis we randomly chose 2 stores from each type, thus doing the evaluation based on 8 stores. We were unable to evaluate all 1115 stores due to resource and time limitations.

The rest of the paper is organized as follows. In the methodology section, we discuss our LSTM model and the forecasting pipeline of the LSTM analysis. In the machine learning models section, we discuss the two machine learning models and their analysis pipeline. In the next section, we present our obtained results. The discussion section elaborates on the obtained results. Related work section discusses existing literature in the domain and the final section concludes the paper.

# METHODOLOGY

This section provides the methodology we used to build the LSTM models. Let us first define the problem we attempt to tackle in the paper.

Consider a set of $d$ temporal attributes $\mathrm{X}_t = \{x_{t,j}\}_{j=1}^{d}$ that describes a store and its operations for a given time $t$ (e.g. day, availability of a promotion etc), which leverage the number of sales $\mathrm{S}_t$. A typical sales forecasting task involves estimating $\mathscr{F}_m^n$ such that,

$$\mathrm{S}_{t+n} = \mathscr{F}_m^n\Big([\mathrm{X}_{t-m}, \mathrm{X}_t], [\mathrm{S}_{t-m}, \mathrm{S}_t], (\mathrm{Z}_t, \mathrm{Z}_{t+n}]\Big)$$

Here $n > 1$ and $m > 0$ are corresponding to the number of steps from the current time to the predicted future and number of steps that are taken into account from the history to predict the future, respectively. However, in our specific task, we only consider the scenarios where $n = 1$, which forecast the daily sales of the very next day. Moreover, $\mathrm{Z}_t$ are the set of attributes from $\mathrm{X}_t$ that is known to us always prior to $n$ time steps.

It should be emphasized that we do not use any time-invariant features in our analysis since we consider sales as a whole, not for individual products. Therefore, time-invariant information will not add any value in this context.

## LSTM Network Architecture

LSTM (Hochreiter and Schmidhuber, 1997) is a descendent of traditional neural networks. Although traditional neural networks have its perks, it suffered from a major flaw of not being able to persist information about previous time-steps, thus losing possible information about correlations. The RNN (Lipton, 2015) solved this issue as it is equipped with an architectural component called the “hidden state”. This acts as a memory and helped the RNN to persist information of the previous time-steps. Due to RNN being subject to vanishing gradients rather heavily, it could only retain short term dependencies. The LSTM model was introduced to mitigate this issue. It has a “hidden state” as well but in addition to that it also has an architectural component named the “cell state”. The hidden state helps in retaining short term dependencies and the “cell state” helps in retaining long term dependencies. LSTM architecture also introduces several gates as the forget gate, input gate and output gate. The forget gate and the input gate controls which part of the information should be removed or reserved and the output gate generates an output according to the processed information (Yunpeng et al., 2017). In our work, we used a special variant of LSTM called “LSTM with peephole connections” (Lipton, 2015; Gers et al., 1999; Gers et al., 2003). This incorporates the previous state of the cell state into the LSTM input and the forget gate Bandara et al. (2019). The peephole connections help in boosting the performance of timing tasks like counting objects and emitting a meaningful output when a defined number of objects have been seen by the network. This ability helps the network to learn to accurately measure intervals between events (Lipton, 2015), which is useful in time-series analysis to learn the contribution of certain intervals towards the final prediction. As an example, consider a feature “day of the week” which will be fed to an LSTM network. We can expect some fixed number of sales for each day that contributes to the total sales for that day, that is solely determined by the day of the week. Therefore, the model must now learn to count the days of the week as they repetitively appear and produce a suitable output reflecting the number of sales that occurs as a repetitive pattern.

In our work, we initially introduce using historical information (HF) with daily sales values and later on in our improved model we incorporate information about the known future (FF) into the features of the initial model. We expect the features to have correlations illustrated in Figure 1. Ultimately, we can predict the number of sales by understanding the existence and intensity of such relationships. Our selection of LSTM is based on its ability to capture all these relationships without any additional effort apart from its


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

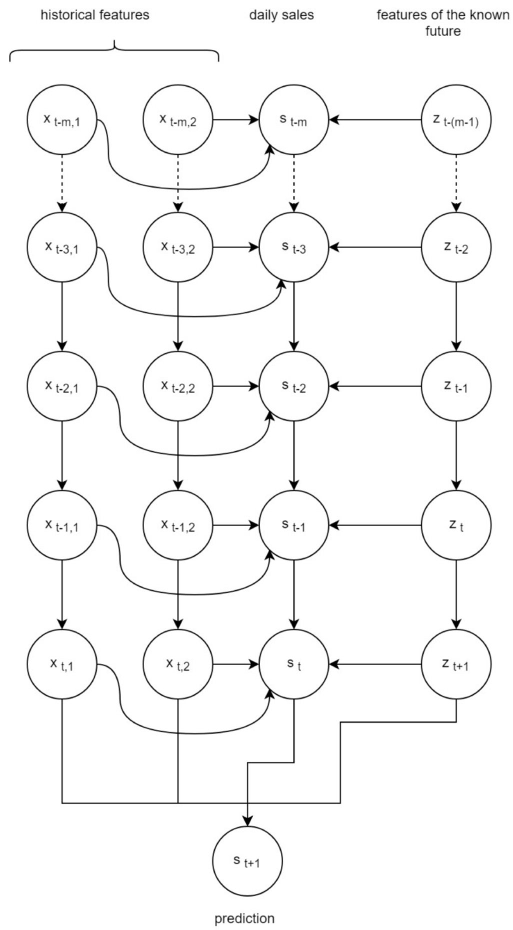


**Figure 1.** feature correlation graph

reputation in time-series analysis. Using the hidden state and cell state of LSTM, it can learn relationships in the temporal axis for each HF feature ($\{X_i\}_{i=t-m}^{t}$, $\{S_i\}_{i=t-m}^{t+1}$) and FF feature ($\{Z_i\}_{i=t-m}^{t}$). On the other hand, LSTM also captures the correlations between number of sales ($S_t$) for each time step with the HF features ($\{x_{tj}\}_{i=1}^{d}$) and FF features ($\{z_{tj}\}_{i=1}^{\bar{d}}$) at the same time step. Moreover, the peephole connections help in extracting crucial insights from temporal intervals in HF, FF and daily sales information. Finally, we can model the relationship between all the information captured and the sales value that is being predicted $S_{t+1}$ using additional layers in between the LSTM layer and the output layer.

We used the same basic architecture for both the initial and the improved LSTM models. The first layer of our model's architecture comprises of an LSTM layer with peephole connections. This aids in capturing all the time series specific information about our data. The output of the LSTM model may have remaining non-linearities, to capture these we then employed two dense hidden layers. Then we implemented a dropout layer to reduce possible chances of over-fitting through regularizing the output. Finally, the output layer is put to structure the model's output to derive the desired prediction. To reduce the magnitude in the change of the learning rate as the training progresses, we used exponential decay, a learning rate decay algorithm. This increased the ability of the model to converge. Adam optimizer was used as the optimization function in our model as it is widely known to perform better than backpropagation methods. Moreover, we used the mean squared error function to calculate the loss of each training step. We implemented this LSTM model using TensorFlow library [2].

[2]https://www.tensorflow.org/

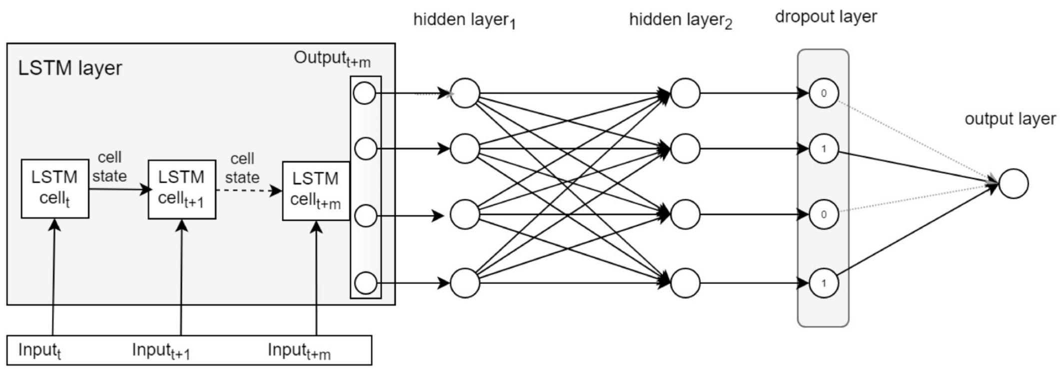


**Figure 2.** LSTM architecture

## Features

This section presents the features we used when training the models. We conducted our analysis as a multivariate forecasting task thus, we employed several other features apart from using the historical daily sales values. For the initial stage, we wanted to study how historical data can be employed to forecast the number of sales analogous to traditional time series forecasting tasks. The original data-set included attributes like date, state holiday, promotion availability, school holiday, store open/close information and the number of customers. We decomposed the composite attribute date into three separate features like day, month and year. Moreover, we further simplified the day to indicate the day of the week, as most of the sales trends are directly co-related to the day of the week. Through empirical analysis we identified that day of the week, promotion availability information and school holiday information were the best combinations of historical features that maximize the forecasting accuracy. We have omitted any information related to the number of customers because we do not know that information for the day being predicted, it is observed at the end of that particular day along with the true number of sales. Therefore, we implemented the initial model based on these three features combined with daily sales values.

Then, we extend our initial model employing the information that described the future that is known to us at the ahead of a sufficient number of time steps (FF). Features like the day of the week and state holiday information can be considered as information from the future that is known by us even before years ahead. Of course, the government may unexpectedly declare state holidays under certain circumstances, yet such are rare occasions and still, we will learn such changes prior to adequate time. Therefore, we obtain FF features from the HF features by selecting the features that are known to us. In our specific scenario, any HF can be used as FF. It should be noted that we could use any feature that qualifies as FF though we consider only the features already identified as HF. Through empirical analysis, we identified that promotion availability and school holiday information to provide the best accuracy with the validation split. We used these features to train the improved models apart from the HF features that was used with the initial model. Hence, we now have 6 features for our improved models, namely, sales value at time step t, day of the week at time step t, promotion availability information at time step t, school holiday information at time step t, promotion information at time-step t+1 and school holiday information at time step t+1.

## Data Preparation

For both initial and improved models, first and foremost, we divided the entire data-set (with 942 data samples) for each store into three splits as training, validation and testing set. Here we consider the last two months of data as the validation and testing split, allocating exactly one month per each split. Then each of these splits was scaled to values between 0 and 1 using min-max scaling. For the features that have known bounds, we use them (i.e. the lower bound and upper bound of the day of the week are respectively 1 and 7, number of sales are non-negative etc) and rest of the bounds are found based on the minimum or maximum value reported with training split. It should be pointed out that scaling is crucial in this analysis since each feature was operating in significantly different intervals (e.g. sales values ranged between 1000 - 30,000, values of a day of week ranged between 1 -7 etc). Therefore, raw values would


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

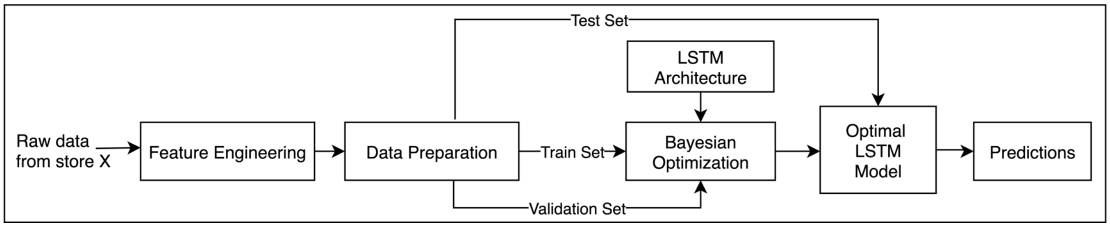


**Figure 3.** pipeline of LSTM analysis

| Hyperparameter | Search Space |
|---|---|
| number of steps | 2 - 14 |
| LSTM size | 8 - 128 |
| batch size | 5 - 65 |
| Initial learning rate | 0.0001 - 0.1 |
| learning rate decay | 0.7 - 0.99 |
| initial number of epoch | 5 - 50 |
| maximum number of epoch | 60- 200 |
| number of nodes in the first hidden layer | 4 - 64 |
| number of nodes in the second hidden layer | 2 - 32 |
| dropout rate | 0.1 - 0.9 |
| activation of first hidden layer | ReLU , Tanh |
| activation of the second hidden layer | ReLU , Tanh |
| activation of LSTM | ReLU , Tanh |

**Table 1.** hyperparameter search space for initial and improved models for Bayesian optimization

have given more influence to the larger sales values over the day of the week, which have affected the accuracy of our models considerably.

However, we kept the original sales values for validation and testing sets. During our evaluations, we re-scaled the predicted outputs to its original scale in order to compute the error metrics for non-scaled sales.

## Hyperparameter Optimization

We realized that LSTM model requires tuning too many hyperparameters and manually tuning each hyperparameter for the enormous search space is not a feasible task. The evaluation included 8 stores and needed tuning 13 hyperparameters for two different LSTM models thus, forcing us to tune $13 \times 8 \times 2$ hyperparameters if we can run each experiment exactly once. Therefore, the need to automate the hyperparameter optimization process became mandatory.

To automate the hyperparameter optimization process we employed a Bayesian optimization based on the Gaussian Process (GP)[3]. Bayesian optimization finds a posterior distribution as the function to be optimized during the parameter optimization, then uses an acquisition function to sample from that posterior to find the next set of parameters to be explored (Brochu et al., 2010). Since Bayesian optimization decides the next point based on more systematic approach considering the available data it is expected to yield achieve better configurations faster compared to the exhaustive parameter optimization techniques such as Grid Search and Random Search. Therefore, Bayesian optimization is more time and resource efficient compared to those exhaustive parameter optimization techniques, especially when we are required to optimize 13 parameters including 3 parameters with a continues search space. Table 1 illustrate the optimized hyperparameter and the search spaces used for each hyperparameter in each experiment. In our implementation, we will be striving towards minimizing the regression error metric of the model.

Figure 3 presents the complete pipeline used in our experiments to construct the LSTM models. We perform feature engineering as explained in section Features on top of the raw data which is followed by

[3]https://scikit-optimize.github.io/#skopt.gp_minimize


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

| Hperparameter | Grid Search Values |
|---|---|
| learning rate | 0.1, 0.01, 0.75 |
| maximum depth | 2, 5 |
| subsample | 0.5, 1, 0.1, 0.75 |
| colsample by tree | 1, 0.1, 0.75 |
| n estimators | 50, 100, 1000 |

**Table 2.** hyperparameter search space explored for XGB with Grid Search

| Hperparameter | Grid Search Values |
|---|---|
| n estimators | 50, 100, 1000 |
| maximum depth | 2, 5 |

**Table 3.** hyperparameter search space explored for RFR with Grid Search

data preparation elaborated in the previous section. Then we construct the LSTM model with the best hyperparameter configuration using train and validation sets following the automatic hyperparameter optimization explained in this section. Finally, our pipeline outputs the optimal LSTM model, which we use for the evaluations.

## MACHINE LEARNING MODELS

To compare the results we obtained from the LSTM model we conducted the same evaluation on two state-of-the-art ensemble machine learning models that are capable of dealing with non-linearities in data. They are the RFR (Breiman, 2001) and the XGB regression (Chen and Guestrin, 2016). RFR makes use of multiple decision trees and bagging techniques that involve training each decision tree on a different data sample where sampling is done using replacement. The work-flow of RFR is as follows: At each step of building an individual tree, it finds the best split of data. Then while building a tree it uses a bootstrap sample from the data-set. Finally, it aggregates the individual tree outputs by averaging. XGB is a tree boosting based model that is highly scalable. When using gradient boosting for regression, the weak learners are regression trees, and each regression treemap an input data point to one of its leaves that contains a continuous score. The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. Both stages of the analysis carried out when evaluating the LSTM model were done when evaluating both the machine learning models. The feature selection, scaling and data splitting of the initial and the improved stages were also carried out the same way as described in the LSTM forecasting methodology. However, when including FF features into the machine learning models, lagging the data was not necessary as machine learning models have no notion of time steps.

### Hyperparameter Optimization

This section discusses the pipeline of hyperparameter optimization, training, validating and testing of both the initial and the improved machine learning models. Both XGB and RFR and both the initial and the improved models used the same pipeline.

Similar to the LSTM model's methodology, we employed a hyperparameter optimization for both the initial and the improved models. XGB and RFR have a set of hyperparameters that affect its performance. Even Though the number of parameters is not as many as in the LSTM model, manually tuning each of these parameters for 8 stores is a rather tedious task. Thus, we decided to implement a Grid Search for the hyperparameter optimization task. We have used the Grid Search approach here as the number of hyperparameter values to be optimized was small so that the process would not be overly time-consuming. We defined the value bounds for the hyperparameters that the Grid Search algorithm should explore. The Grid Search was implemented the same way for both machine learning algorithms. The optimized hyperparameters in XGB were learning rate, maximum depth, subsample, colsample by tree and $n$ estimators. The optimized hyperparameters in RFR were max-depth and $n$ estimators. Shown in Table 2


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

| Store | Store type | RFR (RMSE) | XGB (RMSE) | LSTM (RMSE) |
|---|---|---|---|---|
| 749 | a | 791.97 | 738.65 | **627.03** |
| 85 | b | 804.03 | 803.154 | **617.93** |
| 519 | c | 763.27 | **757.78** | 826.60 |
| 725 | d | 789.02 | 650.35 | **584.66** |
| 165 | a | 382.47 | 391.17 | **342.22** |
| 335 | b | **1,290.12** | 1,312.85 | 1,772.99 |
| 925 | c | 987.47 | **980.65** | 1,065.23 |
| 1089 | d | **930.71** | 984.88 | 1,161.25 |

**Table 4.** initial model: comparison using RMSE values

| Store | Store type | RFR (MAE) | XGB (MAE) | LSTM (MAE) |
|---|---|---|---|---|
| 749 | a | 535.33 | 503.07 | **483.04** |
| 85 | b | 646.71 | 630.65 | **473.94** |
| 519 | c | **556.26** | 579.89 | 641.26 |
| 725 | d | 681.15 | 539.44 | **481.85** |
| 165 | a | 316.70 | 312.11 | **276.85** |
| 335 | b | 954.77 | **944.06** | 1,346.65 |
| 925 | c | 763.74 | **758.11** | 878.56 |
| 1089 | d | **654.04** | 703.00 | 854.89 |

**Table 5.** initial model: comparison using MAE values

and Table 3 are the hyperparameter search values used for each hyperparameter in both approaches for XGB and RFR.

Before initiating the execution, the optimal *m* for each store needed to be found in order to achieve a better accuracy as the forecasting is heavily dependent on *m* when using machine learning models. For this task, we implemented a mechanism to exhaustively check through a defined range of values (2 to 14) for the optimal *m* for each store. We used the validation set for this task.

The *m* that provided the lowest error metric value for each store for the validation set was identified as the optimal m. After obtaining the optimal *m* for each store, we split the data using the derived *m* and ran the train input data set through the Grid Search of the RFR model and the XGB model and derived the validation predictions using the validation input data to determine the optimal hyperparameter values that gave the lowest error metric value for the validation set for both models. We then initialized the model with the obtained respective optimal hyperparameter values and ran the test set through the models to obtain the final predictions. This process was executed for both initial and improved models for all 8 stores.

## EXPERIMENTAL RESULTS

This section provides the analysis of results for the initial and improved LSTM models. To evaluate both LSTM and machine learning models, we used Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) as error metrics. We have employed RMSE for the hyperparameter optimization task of both LSTM and machine learning models. Considering $y_i$ and $\bar{y}_i$ respectively as the true sales and predicted sales , shown in Equations 1 and 2 are the respective equations for RMSE and MAE;

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}{n}} \tag{1}$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}(y_i - \bar{y}_i)}{n} \tag{2}$$

Table 4 and table 5 show the RMSE and MAE values for initial models. Table 6 and table 7 shows the RMSE and MAE values for the improved models.Considering these tables, the values in bold show the


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

| Store | Store type | RFR (RMSE) | XGB (RMSE) | LSTM (RMSE) |
|---|---|---|---|---|
| 749 | a | 716.96 | 674.59 | **494.44** |
| 85 | b | 750.56 | 719.27 | **683.38** |
| 519 | c | 765.21 | **665.88** | 732.08 |
| 725 | d | 603.38 | 565.42 | **541.01** |
| 165 | a | 393.96 | 415.41 | **347.57** |
| 335 | b | 1,208.06 | 1,455.11 | **949.28** |
| 925 | c | **865.40** | 914.29 | 986.86 |
| 1089 | d | 985.14 | 921.87 | **816.24** |

**Table 6.** improved model:comparison using RMSE values

| Store | Store type | RFR (MAE) | XGB (MAE) | LSTM (MAE) |
|---|---|---|---|---|
| 749 | a | 464.65 | 427.00 | **328.23** |
| 85 | b | 603.45 | 573.71 | **546.00** |
| 519 | c | 558.90 | **501.68** | 516.71 |
| 725 | d | 475.45 | 446.61 | **431.00** |
| 165 | a | 310.90 | 313.19 | **261.81** |
| 335 | b | 907.29 | 1,072.77 | **766.97** |
| 925 | c | **643.29** | 676.97 | 768.71 |
| 1089 | d | 649.87 | 648.52 | **614.13** |

**Table 7.** improved model: comparison using MAE values

lowest RMSE/MAE values achieved for each store from the 3 algorithms. The orange colour portrays the comparison of results between the machine learning algorithms, thus the orange coloured cells show the lowest RMSE/MSE values when comparing the results of RFR and XGB. The yellow colour portrays the comparison of results between the LSTM and the machine learning algorithms, thus the yellow-coloured cells show the lowest RMSE/MAE value when comparing the LSTM model with the machine learning algorithms.

The graph in Fig 4 depicts how the predicted values of the initial LSTM model and the initial machine learning models compare with the true sales values of store 85 and the graph in Fig 5 depicts how the prediction values of the improved LSTM model and the improved machine learning models compare with the true sales values of store 335. Both the graphs illustrate the ability of the LSTM model to closely follow the spikes of the true values comparatively better than both XGB model and RFR model.

The graph in Fig 6 portrays how the initial LSTM model and the improved LSTM model compare with the true sales values of store 335. It can clearly be seen how the improved LSTM model follows the true values closely while the initial LSTM model shows deviations at most of the spikes of the graph.

## DISCUSSION

Let us first consider the performance of LSTM models compared to the conventional regression techniques. In the tables 4 and 5, we observe a significant improvement in both RMSE and MAE for initial LSTM with 4 stores (749, 85, 725, 165) out of 8 compared to both of the machine learning models. Furthermore, the improved LSTM model has achieved considerably better results for 6 stores (749, 85, 725, 165, 335, 1089) out of 8 compared to the machine learning models based on the error values from the tables 6 and 7. The results clearly suggest that the LSTM model has obtained a significant improvement over both of the two state-of-the-art regression techniques.

The better performance of LSTM is due to its superior ability to model time-series features. Machine learning algorithms have no notion of the different time steps of data or any kind of time series specific information, they merely perform a regression task on the given data, whereas the LSTM understands the concept of times steps and are strong tools used extensively in time-series forecasting (Bandara et al., 2019). LSTMs are capable of modelling long-range dependencies. The LSTM architecture contains a cell state in addition to a hidden state, that enables the LSTM to propagate the network error for much longer

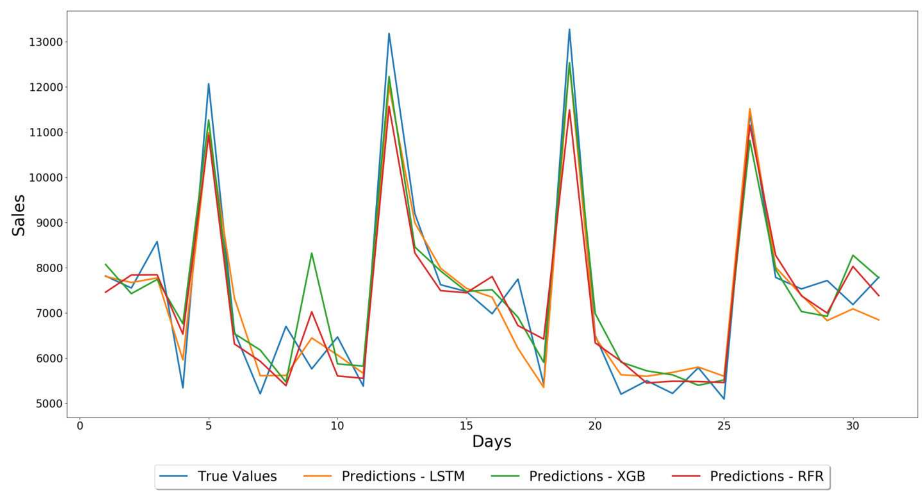


**Figure 4.** predicted values vs true values graph of store 85 - initial model

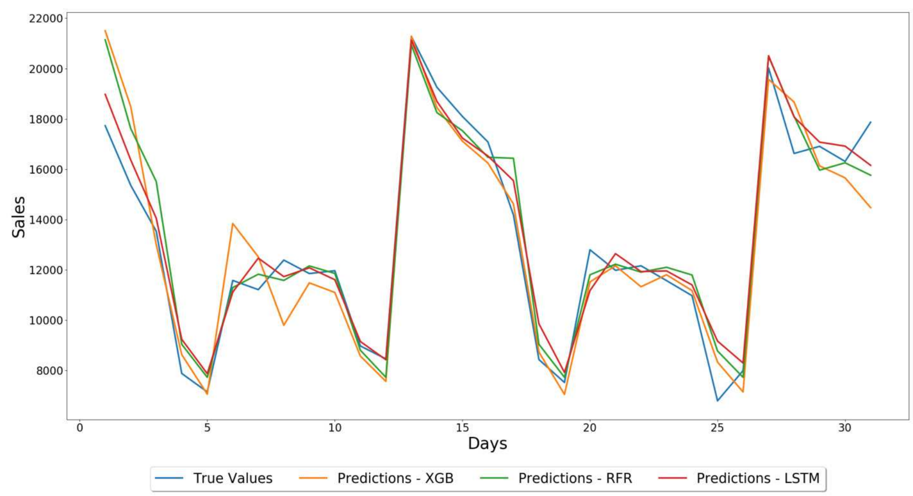


**Figure 5.** predicted values vs true values graph of store 335 - improved model

sequences while capturing their long-term temporal dependencies (Bandara et al., 2019; Chniti et al., 2017) LSTMS can also fit a wider range of data patterns compared to the traditional models (Yunpeng et al., 2017). These factors have enabled the LSTM to produce more accurate forecasts compared to two conventional machine learning models.

On the other hand, initial LSTM has shown the worst accuracy for the rest of the four stores (519, 335, 925, 1089). Even though RFR and XGB have obtained comparable performance against each other, the error values of initial LSTM model has significantly deviated from the RMSE and MAE of XGB and RFR. We believe this surprisingly poor accuracy of LSTM is a result of the over-fitting of the LSTM due to insufficient data. It should be noticed that we only use 881 $(942-31-30)$ data samples to train each model, yet LSTM is known to yield better results with larger data-sets. On the other hand, RFR and XGB are specifically designed to work with small data-sets minimizing the over-fitting. Therefore, we can justify the poor accuracy of LSTM with the rest of the stores as an indication of over-fitting due to insufficient data.

Interestingly, the improved LSTM outperforms the initial LSTM for 6 stores (749, 519, 725, 335, 925, 1089) and 7 (749, 519, 725, 165, 335, 925, 1089) stores respectively based on the RMSE (table 8) and MAE (table 9). The reduction in error is significant (20%-21%) when considered the FF features for sales forecasting. Moreover, we see similar improvements with improved XGB and RFR compared to the initial XGB and RFR. Our observations emphasize the significance of using information describing the future to


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

| Store | Store type | LSTM - Initial | LSTM - Improved |
|---|---|---|---|
| 749 | a | 627.03 | 494.44 |
| 85 | b | 617.93 | 683.38 |
| 519 | c | 826.60 | 732.08 |
| 725 | d | 584.66 | 541.01 |
| 165 | a | 342.22 | 347.57 |
| 335 | b | 1,772.99 | 949.28 |
| 925 | c | 1,065.23 | 986.86 |
| 1089 | d | 1,161.25 | 816.24 |

**Table 8.** comparison of LSTM results of the initial model and the improved model - RMSE

| Store | Store type | LSTM - Initial | LSTM - Improved |
|---|---|---|---|
| 749 | a | 483.04 | 328.23 |
| 85 | b | 473.94 | 546.00 |
| 519 | c | 641.26 | 516.71 |
| 725 | d | 481.85 | 431.00 |
| 165 | a | 276.85 | 261.81 |
| 335 | b | 1,346.65 | 766.97 |
| 925 | c | 878.56 | 768.71 |
| 1089 | d | 854.89 | 614.13 |

**Table 9.** comparison of LSTM results of the initial model and the improved model - MAE

anticipate daily sales. For an example, knowing whether the day being foretasted has a promotion can provide essential information to the models because the anticipation of such unpredictable events is not possible even with state-of-the-art time series models such as LSTM (unless the promotions follow a certain time-series).

Discussing the machine learning models, in the initial model, 5 stores (749, 85, 519, 725, 925) have performed better with the XGB model and the remaining 3 stores (165, 335, 1089) have done better with RFR model when evaluated using RMSE metric. When evaluating with MAE metric, 6 stores (749, 85, 725, 165, 335, 925) have done better with the XGB model and 2 stores (519, 1089) have done better with the RFR model. Considering the improved model's results, 5 stores (749, 85, 519, 725, 1089) have done better with the XGB model and the remaining 3 (165, 335, 925) stores have done better with the RFR model out of 8 stores when evaluating with the RMSE metric. However, unlike in the initial stage analysis, the same stores that have a better improvement with XGB compared to RFR when evaluating with the RMSE metric have also shown a better improvement with XGB than with RFR when evaluating with the MAE metric. According to the obtained results, when comparing the two machine learning models, we can state that the XGB model has outperformed the RFR model. The reason for XGB obtaining better results compared to the RFR is mainly because of it's boosted trees being derived by optimizing an objective function which makes it easier to solve all objective functions that a gradient can be written for. These type of tasks are harder for RFR models to achieve. Furthermore, XGB performs the optimization in a function space rather than in parameter space, which makes the use of custom loss functions much easier than in RFR models.

## RELATED WORK

A significant amount of work has been done to improve the task of sales forecasting. These approaches are mainly based on statistical models, machine learning, neural networks, ensemble techniques, and RNN/LSTM based approaches. In our literature analysis, we discuss RNN/LSTM based approaches in detail since they are much closer to our approach. Let's consider each approach. Among statistical methods, the traditional Auto-Regressive Integrated Moving Average (ARIMA) model has been used as the baseline in most studies for sales forecasting (Müller-Navarra et al., 2015; Pavlyshenko, 2016; Gurnani et al., 2017). However, the traditional ARIMA models cannot handle multivariate features (Bandara


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

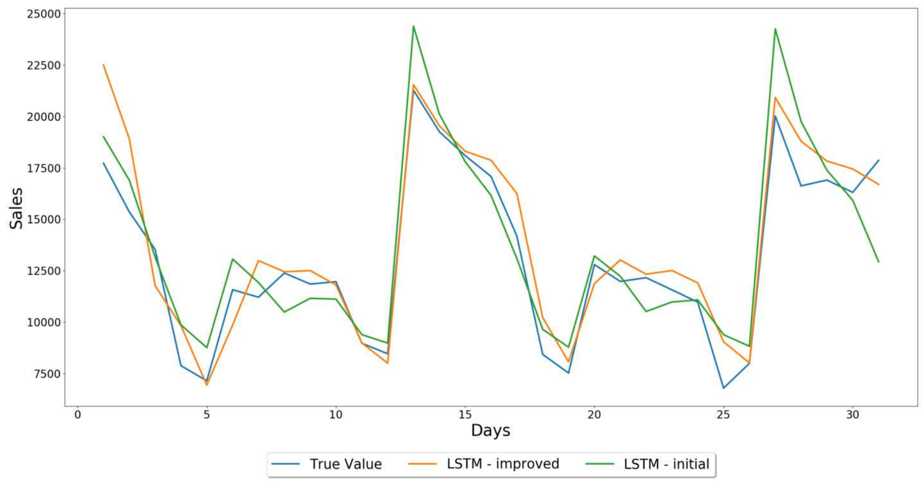


**Figure 6.** predictions of improved LSTM model vs predictions of initial LSTM model - Store 335

et al., 2019) and also shows poor performance in handling seasonality and trend (Gurnani et al., 2017). Xiangsheng Xie (2008) and Wu et al. (2012) have adopted two variants of ARIMA; Seasonal ARIMA and Vector Auto-Regressive Moving Average (ARMAV) with the linear trend to handle above properties in sales forecasting tasks. Gurnani et al. (2017) show that ARIMA with external regressors is most suitable to model the linearity in time series data, yet fail to capture non-linear patterns (Zhang, 2003).

On the other hand, though machine learning and regression techniques are not specifically built for time-series forecasting, they have been considered as promising contenders compared to most of the statistical methods due to their ability to handle both linear and non-linear tasks by considering time-series features as lag features (Doornik and Hansen, 1994). For example, most of the work in sales forecasting based on Rossmann data-set have adopted various machine learning techniques to model such non-linear patterns effectively. Doornik and Hansen (1994) performs sales forecasting analysis for Rossmann data-set using linear regression, softmax regression and Support Vector Machine (SVR) where SVR managed to significantly outperform softmax regression. Lin et al. (2015) also explored sales forecasting using SVR and Frequency Domain Regression (FDR) with the Rossmann data-set. His findings show that SVR with polynomial kernel outperformed FDR as it achieved the best balance between overfitting and underfitting. Pavlyshenko (2016) explores different linear, machine learning and probabilistic models for sales forecasting. He discussed the advantages of using probabilistic models such as Bayesian inference and copulas modelling for the risk assessment of forecasted sales. Moreover, Xiangsheng Xie (2008) also illustrated the superiority of machine learning approaches such as SVM over statistical methods for both short and long term forecasting of sales from the catering industry.

In recent years, deep neural networks have also been adopted for sales-forecasting due to their superior performance in modelling complex non-linear patterns compared to both statistical methods and most of the machine learning approaches. Qin and Li (2011) explores sales forecasting of a fast food manufacturing corporation using a backpropagation neural networks. They claim that the end result is better than the traditional regression analysis approaches. Omar and Liu (2012) tackles the sales forecasting task of magazines by introducing a back propagation neural network (BPNN) based architecture using historical sales data and popularity indexes of magazine article titles. They state that the BPNN algorithm outperforms other statistical algorithms and that by providing additional information on the popularity index gives better accuracy numbers. Li et al. (2012) also illustrate that backpropagation neural networks can yield satisfactory results for vehicle sales forecasting. As traditional BPNN algorithms were providing promising results, studies were conducted on improving BPNN networks by adding different extensions to it. Jiang (2012) proposed an improved back propagation neural network with a conjugate gradient algorithm that shortens training time and improves the forecasting precision for sales forecasting of a corporation. A sales forecasting based on fuzzy neural networks (FNN) was proposed by Liu and Liu (2009) and the study claims that FNNs with weight elimination can outperform traditional artificial neural networks. Gao et al. (2009) discusses rearranging Holt-Winters model to build a neural


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

network on top of it and he has empirically proven that the neural network approach can yield better results than the traditional Holt-Winters model (Makridakis et al., 1984). Kaneko and Yada (2016) constructed a sales prediction model using deep learning and L1 regularization which when given the sales of a particular day, predicts changes in sales on the following day. Their experiments show that deep learning is highly suitable for constructing models that include multi-attribute variables compared to logistic regression.

Most of the work has established that the ensemble-based approaches to provide more accurate forecasts compared to individual models for sales forecasting tasks. ARIMA combined with XGB (Pavlyshenko, 2016), ARIMA with ARNN Gurnani et al. (2017), ARIMA with SVM (Gurnani et al., 2017), SARIMA with wavelet transform (Choi et al., 2011) and ARMAV with linear trend model (Wu et al., 2012) are some examples for combinations with statistical algorithms. In addition to the statistical combinations, there are also ensemble techniques that combine deep learning and machine learning algorithms. Chang et al. (2017) proposed a deep neural network algorithm for forecasting sales of a pharmaceutical company with an architecture that comprises of an autoencoder that generates the hidden layer abstractions and two other shallow neural nets which specializes in one week ahead predictions. Pavlyshenko (2019) has used regression-based approaches for sales forecasting rather than considering it as a time series forecasting task. They propose stacking several machine learning models and neural networks together into several layers to obtain forecasts and claims that this approach outperforms the individual performance of regression models and neural networks. Doganis et al. (2006) proposes a sales forecasting technique that combines the radial basis function (RBF), neural network architecture and a specially designed genetic algorithm for input selection. They claim that the proposed architecture gives better results compared to other ensemble methods like Linear AR-Linear MA, Neural Network AR-Neural Network MA, Neural Network AR-Linear MA, Linear AR-Neural Network MA and individual methods as well. Katkar et al. (2015) has introduced a sales forecasting method that uses fuzzy logic combined with a Naïve Bayesian classifier and the results show that it can achieve satisfactory results. Apart from ensemble methods, some studies have explored decomposing approaches where the sales forecasting tasks are decomposed to multiple, simple modelling components. Gurnani et al. (2017) has explored different statistical, machine learning, hybrid and decomposing methods. They proposed to break the series into three parts: seasonal, trend and remainder and analyzed each component using different machine learning and statistical algorithms. They demonstrated that decomposing the series and tackling individual aspects of the data separately can give better results than individual and hybrid methods. It is also worth mentioning that apart from the above-mentioned methodologies, there are also sales forecasting methodologies carried out using data mining (OZSAGLAM, 2015) and extreme learning approaches as well (Gao et al., 2014).

However, most recent and state-of-the-art sales forecasting approaches are mostly based on the ability to persist memory in deep neural networks using RNNs and LSTMs. Müller-Navarra et al. (2015) discusses the performance of 3 partial recurrent neural network architectures for sales forecasting of a real-world sales data-set and empirically proves that partial recurrent neural networks can outperform statistical models. Carbonneau et al. (2008) analyzed several different machine learning and deep learning approaches on a slightly different task from sales forecasting. They adopted RNN and SVM for demand-forecasting and achieve the best accuracy compared to a set of conventional regression techniques. Recently, it has been shown that multivariate LSTM with cross-series features to outperform the univariate models for similar time series forecasting tasks. Chniti et al. (2017) propose to forecast the prices of mobile phones while considering the correlations between the prices of different phone models by multiple providers in the cell phone market, as a cross-series multivariate analysis. Their technique achieves a significant accuracy gain compared to an SVR model that uses the same information as lag features. Bandara et al. (2019) also use a similar multivariate approach, they have used cross-series sales information of different products to train a global LSTM model to exploit demand pattern correlations of those products. Their multivariate LSTM model with the additional cross-series information significantly outperformed the traditional univariate LSTM models that consider each product individually. We derive our approach for sales forecasting based on the multivariate LSTM models due to their recent success in time-series forecasting in similar tasks. In cross series multivariate prediction the number of sales for store $a$ is predicted using the numbers of sales of stores that have a relationship with $a$. However, with the data-set we have, we cannot identify which stores have relationships to which store. Therefore we cannot consider cross-series correlations between similar entities as seen in previous approaches. Instead, we


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

have multiple features describing a single store thus, using a multivariate approach we attempt to find the relationship between those features and the number of sales for that particular store. We adopt a special variant of the LSTM model called peephole LSTM connections (Lipton, 2015; Gers et al., 1999) that can aid in identifying time-based patterns in our data-set better than a normal LSTM model. We train the model using historical information attached with the number of daily sales such as the day of the week and whether a particular day is a holiday, etc. In addition to such historical features, we improve our models by including the information that describes the future that is known to us at the current moment (i.e. even though the number of sales is unknown to us for the day being forecast, we still know the day of the week and whether that particular day is considered a holiday). This has not been explored in the previous state-of-the-art for sales forecasting techniques to our knowledge.

## CONCLUSION

In this paper, we adopt a special variant of Long Short Term Memory (LSTM) network; "LSTM with peephole connections" for the sales forecasting tasks. We expose the LSTM to two levels of information. We first introduce a multivariate LSTM model that solely depends on historical information for sales forecasting. We appraise the accuracy of this initial LSTM against two state-of-the-art machine learning techniques, namely, Extreme Gradient Boosting (XGB) and Random Forest Regressor (RFR) using 8 randomly selected stores from the Rossmann data-set. We further improve the prediction accuracy of the initial LSTM model by incorporating features that describe the future that is known to us in the current moment, an approach that has not been explored in previous state-of-the-art LSTM based forecasting models. The initial LSTM we develop outperforms the two regression techniques achieving 12% - 14% improvement whereas the improved LSTM achieves 11% - 13% reduction in error compared to the machine learning approaches with the same level of information as the improved LSTM, thus highlighting the superior capabilities of LSTM for sales forecasting. Furthermore, using the information describing the future with the LSTM model, we achieve a significant improvement of 20% - 21% compared to the LSTM that only uses historical data. Therefore, our analysis emphasizes the significance of using information describing the future for sales forecasting even with state-of-the-art time-series prediction models such as LSTM.

In the future, we are planning to explore the ability to incorporate multiple stores with a single LSTM to extract cross-series information to improve forecasting accuracy. We expect such features to improve time-series forecasting by comprehending the interdependencies between the stores such as competition, partnerships, market distribution etc. Moreover, it is interesting to investigate the importance of incorporating information that describes the future beyond the day being predicted. For instance, the customer buying behaviour for a particular day can significantly affect the fact whether the store is going to be closed in the following day. Yet, the time-series models may not be able to anticipate such relationships without explicitly providing information that represents the future even beyond the day that is being forecast. Therefore, we will be exploring such extensions with our technique in the future.

## REFERENCES

Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., and Seaman, B. (2019). Sales demand forecast in e-commerce using a long short-term memory neural network methodology. *CoRR*, abs/1901.04028.

Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1):5–32.

Brochu, E., Cora, V. M., and de Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599.

Carbonneau, R., Laframboise, K., and Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, 184(3):1140 – 1154.

Chang, O., Naranjo, I., and Guerron, C. (2017). A deep learning algorithm to forecast sales of pharmaceutical products.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.



PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

Chniti, G., Bakir, H., and Zaher, H. (2017). E-commerce time series forecasting using lstm neural network and support vector regression. In *Proceedings of the International Conference on Big Data and Internet of Thing*, BDIOT2017, pages 80–84, New York, NY, USA. ACM.

Choi, T.-M., Yu, Y., and Au, K.-F. (2011). A hybrid sarima wavelet transform method for sales forecasting. *Decision Support Systems*, 51(1):130 – 140.

Doganis, P., Alexandridis, A., Patrinos, P., and Sarimveis, H. (2006). Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering*, 75(2):196 – 204.

Doornik, J. and Hansen, H. (1994). A practical test for univariate and multivariate normality. Technical report, Nuffield College, Oxford, UK, Discussion paper.

Gao, M., Xu, W., Fu, H., Wang, M., and Liang, X. (2014). A novel forecasting method for large-scale sales prediction using extreme learning machine. In *2014 Seventh International Joint Conference on Computational Sciences and Optimization*, pages 602–606.

Gao, Y., Liang, Y., , Zhan, S., and Ou, Z. (2009). A neural-network-based forecasting algorithm for retail industry. In *2009 International Conference on Machine Learning and Cybernetics*, volume 2, pages 919–924.

Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 850–855 vol.2.

Gers, F. A., Schraudolph, N. N., and Schmidhuber, J. (2003). Learning precise timing with lstm recurrent networks. *J. Mach. Learn. Res.*, 3:115–143.

Gurnani, M., Korke, Y., Shah, P., Udmale, S., Sambhe, V., and Bhirud, S. (2017). Forecasting of sales by using fusion of machine learning techniques. In *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*, pages 93–101.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Jiang, X.-F. (2012). The research on sales forecasting based on rapid bp neural network. In *2012 International Conference on Computer Science and Information Processing (CSIP)*, pages 1239–1241.

Kaneko, Y. and Yada, K. (2016). A deep learning approach for the prediction of retail store sales. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 531–537.

Katkar, V., Gangopadhyay, S. P., Rathod, S., and Shetty, A. (2015). Sales forecasting using data warehouse and naïve bayesian classifier. In *2015 International Conference on Pervasive Computing (ICPC)*, pages 1–6.

Li, Z., Li, R., Shang, Z., Wang, H., Chen, X., and Mo, C. (2012). Application of bp neural network to sale forecast for h company. In *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 304–307.

Lin, S., Yu, E. S. K., and Guo, X. (2015). Forecasting rossmann store leading 6-month sales cs 229 fall 2015.

Lipton, Z. C. (2015). A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019.

Liu, Y. and Liu, L. (2009). Sales forecasting through fuzzy neural networks. In *2009 International Conference on Electronic Computer Technology*, pages 511–515.

Makridakis, S., C. Wheelwright, S., and Hyndman, R. (1984). *Forecasting: Methods and Applications*, volume 35.

Müller-Navarra, M., Lessmann, S., and Voß, S. (2015). Sales forecasting with partial recurrent neural networks: Empirical insights and benchmarking results. In *2015 48th Hawaii International Conference on System Sciences*, pages 1108–1116.

Omar, H. and Liu, D.-R. (2012). Enhancing sales forecasting by using neuro networks and the popularity of magazine article titles. pages 577–580.

OZSAGLAM, M. Y. (2015). Data mining techniques for sales forecastings. *International Journal of Technical Research and Applications*, 34.

Pavlyshenko, B. M. (2016). Linear, machine learning and probabilistic approaches for time series analysis. In *2016 IEEE First International Conference on Data Stream Mining Processing (DSMP)*, pages 377–381.

Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. *Data*, 4(1).

Qin, Y. and Li, H. (2011). Sales forecast based on bp neural network. In *2011 IEEE 3rd International*


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019

*Conference on Communication Software and Networks*, pages 186–189.
Wu, L., Yan, J. Y., and Fan, Y. J. (2012). Data mining algorithms and statistical analysis for sales data forecast. In *2012 Fifth International Joint Conference on Computational Sciences and Optimization*, pages 577–581.
Xiangsheng Xie, Jiajun Ding, G. H. (2008). Forecasting the retail sales of china's catering industry using support vector machines. In *2008 7th World Congress on Intelligent Control and Automation*, pages 4458–4462.
Yunpeng, L., Di, H., Junpeng, B., and Yong, Q. (2017). Multi-step ahead time series forecasting for different data patterns based on lstm recurrent neural network. In *2017 14th Web Information Systems and Applications Conference (WISA)*, pages 305–310.
Zhang, G. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159 – 175.


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27712v1 | CC BY 4.0 Open Access | rec: 8 May 2019, publ: 8 May 2019