ACTA INFOLOGICA 2023;7(1):59-70
dergipark.org.tr/acin

DOI: 10.26650/acin.1259067

RESEARCH ARTICLE

Short-Term Sales Forecasting Using LSTM and Prophet Based Models in E-Commerce

E-Ticarette LSTM ve Prophet Esaslı Modeller Kullanarak Kısa Dönemli Satış Tahmini

Alp Ecevit[1], İrem Öztürk[1], Mustafa Dağ[1], Tuncay Özcan[2]

[1] Private corporation

[2](Assoc. Prof.) Istanbul Technical University, Management Engineering Department, Istanbul Technical University, Istanbul, Turkiye

ORCID: A.E. 0000-0003-1685-7642;
İ.Ö. 0000-0002-0264-1798;
M.D. 0000-0003-0291-604X;
T.Ö. 0000-0002-9520-2494

Corresponding author:
Tuncay ÖZCAN
Istanbul Technical University, Management Engineering Department, Istanbul Technical University, Istanbul, Turkiye
E-mail address: tozcan@itu.edu.tr

Submitted: 02.03.2023
Accepted: 14.03.2023
Published Online: 14.04.2023

Citation: Ecevit, A., Ozturk, I., Dag, M., Ozcan, T. (2023). Short-term sales forecasting using LSTM and prophet based models in e-commerce. *Acta Infologica, 7*(1), 59-70. https://doi.org/10.26650/acin.1259067

ABSTRACT

The accuracy of sales forecasting is crucial for e-commerce businesses to optimize inventory management, pricing decisions, marketing strategies and staff scheduling. At this point, different approaches such as statistical models, fuzzy systems, machine learning and deep learning algorithms are widely used for sales forecasting. This study investigates the performance of the deep learning based the Long-Short Term Memory (LSTM) model and the Facebook Prophet model on short-term sales forecasting. The performance of the proposed models is compared with the seasonal autoregressive integrated moving average (SARIMA) using real-life data from an e-commerce site. For the comparative analysis of the proposed forecasting models, weighted average absolute percent error (wMAPE), root mean square error (RMSE) and R-squared are selected as performance measures. The numerical results show that the LSTM model outperforms the Prophet and SARIMA models in terms of forecast accuracy for hourly sales forecasting.

Keywords: Sales forecasting, e-commerce, LSTM, prophet

ÖZ

Satış tahmininin doğruluğu, e-ticaret işletmelerinin envanter yönetimini, fiyatlandırma kararlarını, pazarlama stratejilerini ve personel planlamasını en iyilemesi için çok önemlidir. Bu noktada, satış tahmini için istatistiksel modeller, bulanık sistemler, makine öğrenmesi ve derin öğrenme algoritmaları gibi farklı yaklaşımlar yaygın olarak kullanılmaktadır. Bu çalışma, derin öğrenme tabanlı Uzun-Kısa Süreli Bellek (LSTM) modeli ve Facebook Prophet modelinin kısa vadeli satış tahmini üzerindeki performansını incelemektedir. Önerilen modellerin performansı, bir e-ticaret sitesinden alınan gerçek hayat verileri kullanılarak mevsimsel otoregresif bütünleşik hareketli ortalama (SARIMA) ile karşılaştırılmıştır. Önerilen tahmin modellerinin karşılaştırmalı analizi için, performans ölçütleri olarak ağırlıklı ortalama mutlak yüzde hata (wMAPE), hata kareleri ortalamasının karekökü (RMSE) ve R-kare seçilmiştir. Sayısal sonuçlar, LSTM modelinin saatlik satış tahmini için tahmin doğruluğu açısından Prophet ve SARIMA modellerinden daha iyi performans gösterdiğini göstermiştir.

Anahtar Kelimeler: Satış tahmini, e-ticaret, LSTM, prophet

1. INTRODUCTION

Sales forecasting is a crucial task for e-commerce businesses as it allows them to anticipate future demand and make informed decisions on inventory management, pricing strategies, marketing campaigns and staff scheduling. Accurately predicting future sales allows retailers to optimize their stock and inventory levels, avoiding the costly consequences of both stockouts and excess inventory. Stockouts can lead to customer disappointment and loss of sales, whereas excess inventory can result in increased storage and holding costs, negatively impacting overall profits (Jing & Lewis, 2011). Therefore, retailers use various forecasting models to improve the accuracy of their sales forecasts. At this point, especially, machine learning and deep learning-based models are very popular.

This study presents the performance analysis of the deep learning based the Long-Short Term Memory (LSTM) model and the Prophet model on hourly sales forecasting. The performance of the proposed models is evaluated by comparison with the seasonal autoregressive integrated moving average (SARIMA). For this analysis, e-commerce data of a grocery retailer is used. In addition, weighted average absolute percent error (wMAPE), root mean square error (RMSE) and R-square of these models are calculated to measure prediction performance. In other words, the aim of this study is to perform a comparative analysis of three popular time series forecasting models such as LSTM, Prophet and SARIMA in order to identify the best model for short-term sales forecasting. Recent studies have shown that deep learning frameworks and Prophet have been effective in sales forecasting for e-commerce applications (Ensafi, Amin, Zhang, & Shah, 2022), whereas SARIMA remained a popular choice for traditional time series forecasting (Zhang, 2003). Each model has its own unique strengths and limitations, making it essential to thoroughly compare them in order to determine the most accurate and efficient model for sales forecasting. The findings of this study have practical implications for e-commerce businesses and contributes to the existing literature on sales forecasting in the field.

The rest of this paper is organized as follows: In the next section, a literature review of the existing methods for sales forecasting in e-commerce and retailing is presented. In Section 3, the methods used for sales forecasting and the performance metrics are explained. In Section 4, the dataset and the data preprocessing and application steps performed are described. In Section 5, the performance results obtained are presented and discussed. Finally, the conclusion of the paper is presented along with the suggested future work.

2. RELATED WORKS

The accuracy of sales forecasting significantly influences the marketing, pricing, inventory and scheduling decisions, especially for retailers and e-commerce companies. Loureiro, Miguéis, & da Silva (2018) acknowledged this statement by emphasized the crucial place of sales forecasting for businesses. Therefore, sales forecasting has been an interesting topic for practitioners and academics alike. There are numerous studies related to sales forecasting in the literature. Some of these studies are discussed in this section.

The importance of statistical models such as Autoregressive integrated moving averages (ARIMA) cannot be denied. However, these kinds of statistical forecasting frameworks have certain limitations. ARIMA models do perform well under the condition of data's linearity (Zhang, 2003; Loureiro et al., 2018; Ji, Wang, Zhao, & Guo, 2019; Bandara et al., 2019; Punia, Nikolopoulos, Singh, Madaan, & Litsiou, 2020). As the limitations of statistical models have been acknowledged and machine learning/ deep learning approaches started to become more efficient, Zhang (2003) proposed a hybrid approach that combined ARIMA and Artificial Neural Networks (ANN) intended to capture both the linear and nonlinear relationship between data points. In this study, various datasets have been implemented to test the hybrid model and this architecture has been evaluated with mean squared error (MSE) and mean absolute deviation (MAD). The hybrid model outperformed both ARIMA and ANN models based on the performance metrics mentioned above in all datasets. Sun, Choi, Au, & Yu (2008), showed the usage of neural networks in sales forecasting tasks. This study took advantage of a neural network framework called extreme machine learning (EML) and used it in the fashion retail industry by examining the relationship between sales amount and product-related features. Chang, Liu, & Fan (2009) used a hybrid structure that combined a k-means clustering algorithm with a neural network to predict the sales of circuit boards. In this study, the performance of the model was evaluated by

comparing it with different kinds of models such as back propagation neural networks, radial basis function neural networks. Different from previous studies, Yu, Choi, & Hui (2011) focused on the time-consuming aspect of Artificial Neural Networks and developed an extreme learning machine (ELM) model. Although it has been indicated in this study, that the extreme learning machines are not stable when compared to ANNs, the developed model has been used in the retail fashion industry which sacrificed performance over agility. Choi, Hui, Liu, Ng, & Yu (2014) proposed a hybrid approach of combining the grey model and the extreme learning machine using synthetic fashion retail data in order to predict the sales of the fast fashion industry. In this study, the proposed model was highly appropriate for industries where products are highly seasonal and are constantly changing i.e., there is a low amount of data available for every product point. Arunraj & Ahrens (2015) developed a hybrid autoregressive model to predict the sales of a banana retailer using features such as day of the year, the month of the year, holiday, weather effect, etc. In this study, focused the seasonal autoregressive integrated moving average with external variables (SARIMAX) and hybrid SARIMA and Quantile Regression (SARIMA-QR) that enabled high-low quantile predictions. The results of this study indicated that SARIMA-QR has been more successful in aspects such as business interpretability, accurate decision-making and insightfulness. Zhao & Wang (2017) proposed a CNN-based model that used a one-dimensional convolution layer to predict sales of an e-commerce site using variables such as user actions and content of the product. Loureiro et al. (2018) presented an Artificial Neural Network model to predict the sales of retail products using product-related data. In this study, the proposed model was compared with different models such as Random Forests, Decision Trees and Support Vector Regression.

Another state-of-the-art approach in time series forecasting tasks is to use Recurrent Neural Networks (RNN) and particularly Long Short-Term Memory (LSTM) models. Yu, Wang, Strandhagen, & Wang (2018) created a LSTM model for the product-based sales forecasting. The model achieved strong results in only one-fourth of the products. These results showed that the necessity of a larger amount of data points and a multivariate structure of both the model and the dataset. One of the more recent algorithms that have been used in time series forecasting tasks was Meta's Prophet (Taylor & Letham, 2017). Weytjens, Lohmann, & Kleinsteuber (2021) compared Prophet, ARIMA and LSTM models in predicting cash flows using transaction-related data. Rather than using mean squared error as a performance metric, this study adopted a domain-related measure called the Interest Opportunity Cost (IOC) which is used to evaluate the above models. In this study, numerical results showed that LSTMs both outperformed the statistical ARIMA model and Meta's Prophet. Ji et al. (2019) developed a hybrid model called C-A-XGBoost for an e-commerce company, which combined clustering, ARIMA and XGBoost algorithms that used features such as user actions, sales, weather and holidays. In order to evaluate the performance of the model, metrics such as mean squared error, root mean square error and mean absolute error were used, and numerical results showed that the proposed C-A-XGBoost model performed better than statistical forecasting algorithms. Bandara et al. (2019) created a model that combined LSTM with K-means using product-related, sales-related and time-related features for sales forecasting. In this study, the numerical results showed that the proposed LSTM-based time series forecasting model outperformed linear forecasting models. The usage of statistical methods for time series forecasting has been limited during the last few years and literature started to emphasize the argument that machine learning and deep learning models outperform models like ARIMA, SARIMA and linear regression. Punia et al. (2020) indicated that machine learning models are insufficient in handling the trend and seasonality in the data, and developed a hybrid model that combined LSTM and Random Forests. The performance of the developed model was compared with LSTM, ARIMAX and multiple regression. The proposed hybrid model performed slightly better than all of the other models. Another example of a forecasting study has been conducted by Chandriah & Naraganahalli (2021), created a LSTM-based model using the Modified-Adam algorithm to predict the demand for spare automobile parts. In this study, sales and stock-related features were used in order to train the model, and the model was evaluated using performance metrics of mean error and mean squared error. Ensafi et al. (2022) performed a comparative analysis of a Seasonal Autoregressive Integrated Moving Average (SARIMA), Prophet model, LSTM model and CNN model for predicting the sales of a furniture store. Numerical results showed that the LSTM model seemed to outperform all others, followed by the CNN model and the Prophet model. Zohdi, Rafiee, Kayvanfar, & Salamiraad (2022) used both machine learning and deep learning models to predict the demand for a supply chain management system. This study compared machine learning models such as K-nearest neighbors, decision trees, extreme gradient boosting and multi-layer perceptron

neural network. The numerical results of the study showed that multi-layer perceptron neural network had a stronger performance than the machine learning models mentioned above. Martínez, Charte, Frías, & Martínez-Rodríguez (2022) emphasized the importance of fast runtimes and developed a Generalized Regression Neural Networks (GRNNs) based framework for time series forecasting tasks. The main aim of this study was to create a fast and accurate neural network framework that was able to capture seasonality and trendiness from data. The numerical results of this study showed that GRNNs can be used in order to train time series forecasting models highly fast and accurately.

3. METHODOLOGY

In this section, mathematical details of the Prophet and LSTM algorithms are discussed along with selected performance measures.

3.1. Prophet

The Prophet procedure is a forecasting model developed by Facebook. There are libraries in Python and R for the implementation of Prophet. Meta's Prophet framework is an additive regression model that consists of four components: a trend function *g(t)* that models non-periodic changes such as the growth over time, a seasonality component *s(t)* that shows the periodic changes (i.e., monthly, yearly, weekly changes), a holiday component *h(t)* that merges the effects of holidays to the model and the *e(t)* parameter which represents the idiosyncratic changes that are not foreseen by the model (Taylor & Letham, 2017).

The calculation steps of the Prophet model are as follows (Taylor & Letham, 2017).

Meta's Prophet framework can be expressed by Equation (1).

$$y(t) = g(t) + s(t) + h(t) + e(t) \tag{1}$$

The Prophet has two trend models, one of them being a saturating growth model and the other one is a piecewise linear model. The nonlinear saturating growth model is a model that aims to model and forecast growth. As the authors indicate, the growth model is typically modeled using a logistic growth model. For example, in a case where the capacity of an online site users depends on the number of people that has access to internet, the typical growth model is given in Equation (2):

$$g(t) = \frac{C}{1 + exp\ (-k(t - m))} \tag{2}$$

Where *C* is the capacity of online site users, *k* is the growth rate, and *m* is the offset parameter.

However, this growth model is missing some aspects of growth. First, the capacity is not constant, so in the Prophet framework, constant *C* is replaced by a time-dependent *C* of *C(t)*. Second, the growth rate is not constant either. In order to catch the changes in trend, the model incorporates changepoints where the growth rate has the ability to change. In a case where there are S changepoints at times s_j, where *j* represents the values from 1 to *S*, the authors define a vector of rate adjustments: $\delta \in \mathbb{R}^S$. Where δ_j is the change in rate at time s_j. The rate mentioned above at time *t* is the base rate *k*, with the adjustments up to that time *t,* and it can be defined by Equation (3).

$$k + \Sigma_{j:t>s_j}\delta_j \tag{3}$$

The equation above can be defined more clearly by using a vector as in Equation (4);

$$a(t) \in \{0,1\}^S \tag{4}$$

where

$$a_j(t) = \begin{cases} 1, & if\ t \geq s_j, \\ 0, & otherwise \end{cases} \tag{5}$$

Rate *k* and offset parameter *m* both need to be adjusted when the former is adjusted in order to connect the endpoints of segments. The adjustment as change point *j* is as in Equation (6):

$$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma_l\right) * \left(1 - \frac{\left(k + \sum_{l<j} \delta_l\right)}{\left(k + \sum_{l \leq j} \delta_l\right)}\right) \tag{6}$$

Then, the rate at time *t* becomes $k + a(t)^T \delta$ as seen in Equation (7). So, the growth model is reached as follows:

$$g(t) = \frac{C(t)}{1 + \exp\left(-(k + a(t)^T \delta)\left(t - (m + a(t)^T \gamma)\right)\right)} \tag{7}$$

The second trend model which is called the linear trend with changepoints is a simple linear model with a fixed rate of growth. This model is more suited for problems where there is not a saturating growth, and can be defined by Equation (8).

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma) \tag{8}$$

Where *k* is the growth rate, δ has the rate adjustments and *m* is the offset parameter. To make function continuous, the growth rate can be expressed by Equation (9):

$$\gamma_j = -s_j \delta_j \tag{9}$$

The changepoints can be defined by the user or may be automatically selected by the model. The literature studies indicate it is advised to define a large number of changepoints. The changepoints can be shown as in Equation (10):

$$\delta_j \sim \mathcal{L}(0, \tau) \tag{10}$$

In this equation, τ directly controls the flexibility of the model.

When the model is going to start making predictions about the future using the data from the past, the *g(t)* will have a constant rate. The uncertainty in the forecast trend will be solved by extending the generative model forward. This generative model is composed of *S* changepoints over a history of *T* in which each has a rate change given in Equation (10).

Simulation of future changes is done by replacing τ with variance as in Equation (11).

$$\lambda = \frac{1}{S} \sum_{j=1}^{S} |\delta_j| \tag{11}$$

Future changepoints are randomly sampled in a way that the average frequency of changepoints in history is the same as the ones in the future as in Equation (12).

$$\forall j > T, \begin{cases} \delta_j = 0 \;\; w.p. & \frac{T-S}{T} \\ \delta_j \sim \mathcal{L}(0, \lambda) \;\; w.p. & \frac{S}{T} \end{cases} \tag{12}$$

Thus, uncertainty in the trend is calculated by making the assumption that the average frequency and magnitude of rate changes in history will be the same as in the future.

The second component which is the seasonality component is modeled using the Fourier series. Where *P* is the time period we want our time series to have (e.g. *P* = 7 for a weekly data or *P* = 365.25 for yearly data in daily-scaled data). The seasonality component of the model can be expressed by Equation (13).

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right)\right) \tag{13}$$

Fitting this seasonality component would require the construction of a matrix of seasonality vectors for each value of t in our data (?). For example, when looking at a yearly seasonality with *N*=10*,* Equation (14) is obtained.

$$X(t) = \left[\cos\left(\frac{2\pi(1)t}{365.25}\right), \ldots, +\sin\left(\frac{2\pi(10)t}{365.25}\right)\right] \tag{14}$$

In this case, the seasonal component can be expressed by Equation (15).

$$s(t) = X(t)\beta \tag{15}$$

In the generative model, Prophet takes $\beta \sim Normal(0,\sigma^2)$ in order to implement a smoothing prior to the seasonality. The more N is increased the more the model's ability to fit seasonality, which changes more rapidly, increases.

The third component ($h(t)$) is for including predictable instances in the model such as national holidays, Valentine's Day, etc. In order to use this feature, the user needs to provide the dates to the model by also specifying the source country of the holiday. For each holiday i, D_i represents the past and future occurrences of the holiday i which then is added by an indicator function that denotes whether the time t is during holiday i. As the last step, the model assigns a parameter of k_i to holiday i that represents the change in the forecast. This procedure is conducted in a similar way as seasonality components which is by generating a matrix of regressors as in Equation (16):

$$Z(t) = [1(t \in D_1), \ldots, 1(t \in D_L)] \tag{16}$$

The holiday component of the Prophet model can be expressed by Equation (17).

$$h(t) = Z(t) * \text{К} \tag{17}$$

and using a prior $\text{К} \sim Normal(0,v^2)$.

3.2. Long-Short Term Memory (LSTM)

Long-Short Term Memory (LSTM) is a particular type of Recurrent Neural Network that is constructed to deal with sequential data. They are generally used for speech recognition, handwriting recognition, and time series forecasting (Weytjens et al. 2021). However, in the traditional RNN there is a gradient vanishing and exploding problem. To overcome gradient vanishing and exploding problems, gated RNNs are introduced which have the ability to learn when and how to forget past data (Goodfellow et al. 2016). One of the most famous gated RNN frameworks is Long-short term memory and the flow of this network is illustrated in Fig. 1.

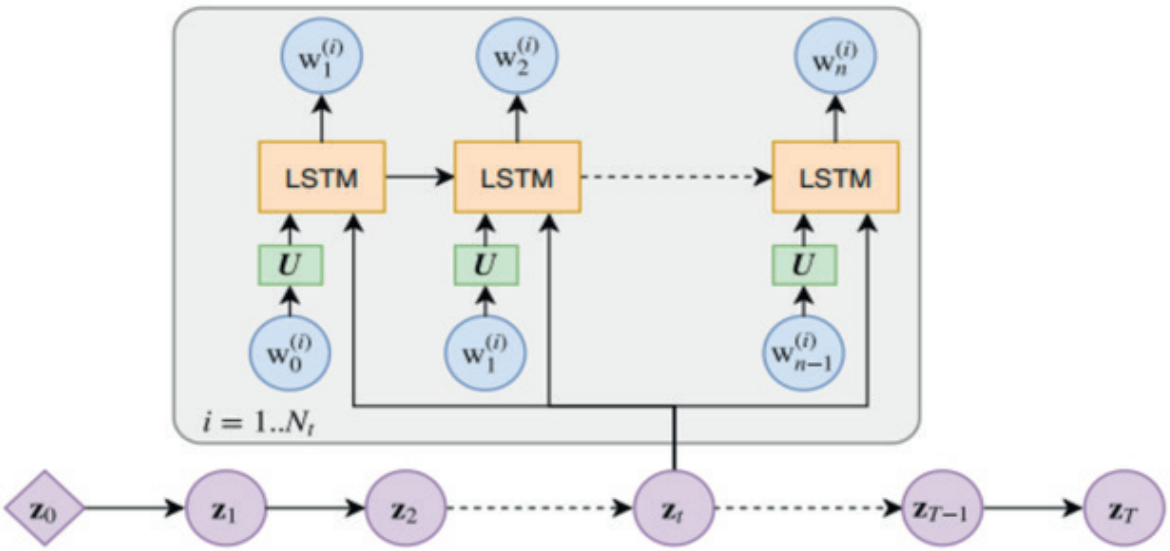

Figure 1. LSTM flow (Delasalles, Lamprier, & Denoyer, 2019)

LSTM includes three stages; forget gate, input gate, and output gate. At the forget gate of f_t,what percentage of the Long-term memory to remember is decided via the sigmoid activation function. Then, at the input gate i_t, short-term memory and the input are used to calculate the potential Long-term memory with hyperbolic tangent function (*tanh*). Similar to the forget gate, the percentage of potential long-term memory to remember is determined with the sigmoid activation function. As a result, Long term memory is updated. At the last stage, the new long-term memory is the input of the *tanh* function to find the potential short-term memory, again the percentage to remember is calculated with the sigmoid function. For a single time step, the model is operated using Equations (18)-(23) (Goodfellow et al. 2016).

In these equations, f_t, i_t, $o_t$represent the forget gate, the input gate and the output gate, respectively. Also, c_t and h_t indicate the cell state and the hidden state, respectively. is used for the sigmoid activation function, while is used for the hyperbolic tangent function.

$$f_t = \sigma_g\,(W_f \times x_t + U_f \times h_{t-1} + b_f) \tag{18}$$

$$i_t = \sigma_g(W_i \times x_t + U_i \times h_{t-1} + b_i) \tag{19}$$

$$o_t = \sigma_g(W_O \times x_t + U_o \times h_{t-1} + b_o) \tag{20}$$

$$c'_t = \sigma_c(W_c \times x_t + U_c \times h_{t-1} + b_c) \tag{21}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t \tag{22}$$

$$h_t = o_t \cdot \sigma_c(c_t) \tag{23}$$

3.3. Performance Measures

In this study, Root Mean Squared Error (RMSE), Weighted Mean Absolute Percentage Error (wMAPE) and R-squared were used as performance measures which are clearly explained in this section.

Mean squared error (MSE) is computed as the average of the squared differences between the predicted and observed values. Root Mean Squared Error (RMSE) is the square root of MSE. These two performance measures have the same units as the target variable. The main difference between these measures is that MSE effectively penalizes larger errors more severely. Hence, an overshoot may result in quite a high MSE value while RMSE smooths it out. The formulas for MSE and RMSE are given in Equations (24)-(25), respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{24}$$

$$RMSE = \sqrt{MSE} \tag{25}$$

In these equations, Y_i is the observed output vector at *i* th instance and $\hat{Y}_i$ is the predicted output vector at *i* th instance.

wMAPE is another metric used in this study to compare the performance of the proposed approaches. This decision is based on the limitations of commonly used metrics, such as the Mean Absolute Error (MAE) or L1 loss, which is computed as the mean of the differences between the actual measurements and the predictions. While MAE can be useful, it is not a meaningful metric by itself, as it depends on the magnitude of the data. To address these limitations, this study chose to use the Mean Absolute Percentage Error (MAPE) as a more meaningful metric. However, even MAPE has its own limitations, as it tends to become less reliable as the sales increase. For instance, forecasting a sale of $130 instead of $100 may not be as unacceptable as forecasting a sale of $13,000 instead of $10,000.

In this study, to overcome these limitations, wMAPE was chosen as the performance measurement metric. It weighs the error by adding the total sales, making it a more suitable metric for the study. MAE, MAPE and wMAPE can be calculated using Equations (26)-(28), respectively.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i\ | \tag{26}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} |\frac{(y_i - \hat{y}_i)}{y_i}\ | \tag{27}$$

$$WAPE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i|} \tag{28}$$

Another performance metric used in this study is R-squared. R-squared is one of the goodness of fit measures. It measures the proportion of variability in *Y* that can be explained using *X*. In other words, it represents the relationship between dependent variables and independent variables. It is equal to the squared correlation between the model and the response (dependent) variable (James et al, 2013). The R-squared can be calculated using Equations (29)-(31).

$$RSS = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{29}$$

$$TSS = \frac{1}{n}\sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{30}$$

$$R^2 = 1 - \frac{RSS}{TSS} \tag{31}$$

In these equations, RSS gives the unexplainable part of the error left after performing the model, while TSS measures the total sum of squares.

4. APPLICATION

In this study, the proposed LSTM, Prophet and SARIMA approaches are applied for hourly sales forecasting with real-life data from an e-commerce site. In this section, the application steps are detailed.

4.1. Data Set

In this study, the real-world dataset that is used to train the models consisted of both numerical features and ordinal features. A sample of the dataset is shown in Fig. 2 and Table 1 with the features of total sales, day of the week, holiday, temperature and weather conditions per timestamp. The dataset used in the study consists of dates per hour starting from December 14, 2022, at 9 a.m. and ending on January 23, 2023, at 9 a.m. with a total of 961 timestamps. In order to maintain data privacy, the values in the data set were scaled.

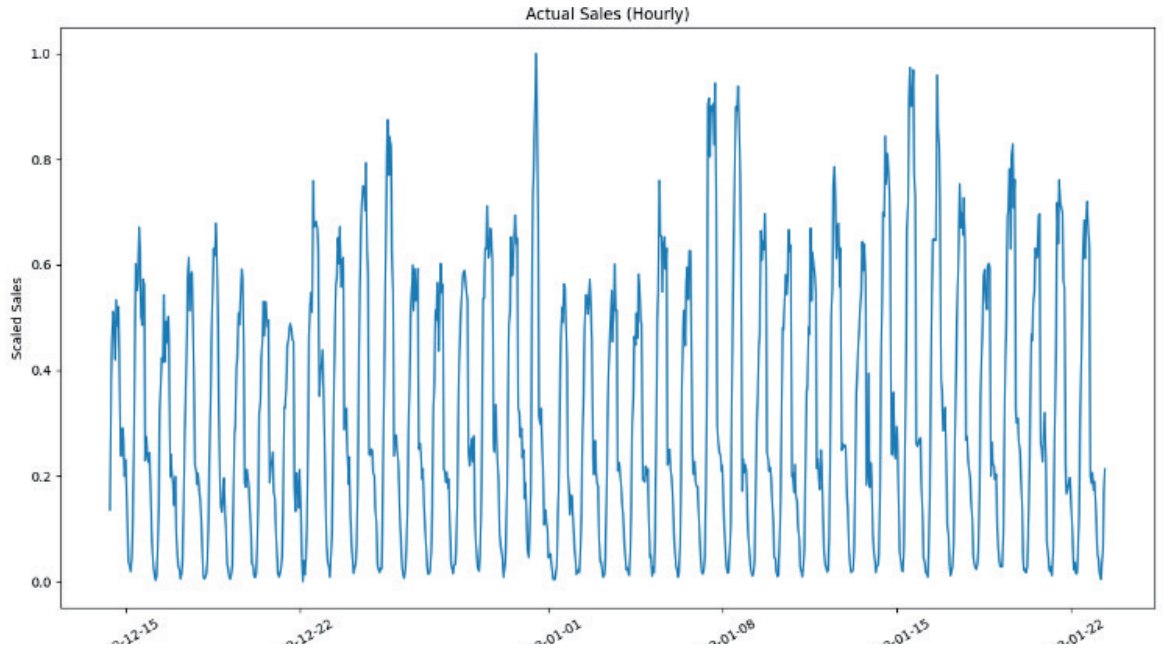

Figure 2. Time series plot of the hourly sales used in this study

Table 1
A sample of the dataset used in this study

Date	Day	Total Sales	Holiday	Temperature	Condition
2022-12-14 9:00:00	4	0.1360	0	10.5	3
2022-12-14 10:00:00	4	0.3830	0	12	4
2022-12-14 11:00:00	4	0.4602	0	12	4
2022-12-14 12:00:00	4	0.5116	0	11.7	3
2022-12-14 13:00:00	4	0.4838	0	12	3
...	...	...	...	...	...
2023-01-23 5:00:00	2	0.0043	0	9.8	2
2023-01-23 6:00:00	2	0.0279	0	10.1	2
2023-01-23 7:00:00	2	0.0581	0	10.5	2
2023-01-23 8:00:00	2	0.1682	0	11	2
2023-01-23 9:00:00	2	0.2129	0	11.5	2

4.2. Short Term Sales Forecasting Using Proposed Models

In this study, the hold-out validation method is used for training the proposed models. 80% of the dataset is used for training the model and the remaining 20% dataset is used for testing. The forecasting performance of the proposed models is compared using R-squared, RMSE, and WMAPE metrics. Since the sales data is scaled using a min-max scaling method, the R-squared calculation is implemented using the observed sales data rather than the scaled values. However, RMSE and WMAPE are calculated using the scaled values due to interpretability reasons.

This study used the Python programming language to train the models. In order to train the models, different Python libraries and/or dependencies are used. For the LSTM model, Tensorflow's Keras API is used; for the Prophet model, the Meta's Prophet package is used, and for the SARIMA model, the statsmodels package is used.

As indicated before, in this study, the SARIMA model is used as a comparison model to evaluate the prediction performance of the LSTM and PROPHET models. Due to the model's univariate structure, only the total sales feature is used for the training process, and other features are ignored. 54 combinations of different parameter values are tried in order to find the

optimal SARIMA model according to AIC values. After the training process, the best parameters (0,1,0), (1,1,1,24) are adopted, and the final model is formed using these best parameters.

After taking the SARIMA model as the baseline model, the Prophet model is trained using all of the features in the data set. Like the SARIMA model, the total sales column is also scaled; however, some additional steps are taken in the training process of the Prophet model. The holiday column of the data is wrapped into the Prophet model's holiday argument. Also, additional regressors (i.e., day of the week, temperature, and weather condition regressors) are added in order to include all of the features in our dataset. The model is trained by implementing both the linear and logistic growth trends.

The LSTM model is trained using all of the features in the data set. After scaling the total sales column, ordinal columns of weather condition and day of the week are encoded as binary features by adding new columns to the data set. Before fitting the data set into the LSTM input layer, the dataset is restructured into a 3D-array in which the timestamps are grouped by days using a shifting method. For instance, the first sample in the modified data set started from the 1st timestamp and finished with the 24th timestamp in which the 25th timestamp's sales data is used as the target variable. This process is replicated for all the timestamps, meaning that the second sample of the modified data set started from the 2nd timestamp and finished with the 25th timestamp where the 26th timestamp's sales data is used as the target variable. So, the shape of the data that included only the independent variables turned into (937, 24, 16) where 937 represented the number of samples (i.e., the grouped days) and the target array is turned into an array with the shape (937,). Representation of the reshaped data can be seen in Fig. 3 in which **t** represents the timestamps at time 1,2,3, etc. and **f** represents the feature index that is between 1 and 16 inclusively.

[[[t1f1, ..., t1f16], [t2f1, ..., t2f16], [t3f1, ..., t3f16], ..., [t24f1, ..., t24f16]] → [t25f1]

[[t2f1, ..., t2f16], [t3f1, ..., t3f16], [t4f1, ..., t4f16], ..., [t25f1, ..., t25f16]] → [t26f1]

...

[[t937f1, ..., t937f16], [t938f1, ..., t938f16], ..., [t960f1, ..., t960f16]]] → [t961f1]

Figure 3. LSTM data structure

In this study, the neural network structure included a Bidirectional LSTM layer, a global pooling layer, and 2 dense layers with relu and linear activation functions. This model is trained for 100 epochs and optimized using the Adam optimizer. In order to prevent overfitting, early stopping callback is added with a monitoring interval of 10 epochs. Different methods are implemented in order to prevent overfitting such as adding a dropout layer, adding l2 regularization, etc. However, adding the early stopping callback showed the best results. The LSTM model is also tested with different architectures, such as using a single dense layer, not using a bidirectional LSTM layer etc. Best results are achieved using the architecture described above.

5. RESULTS

The prediction performance of the proposed models is given in Table 2 for training and testing data. Numerical results showed that the proposed LSTM model outperformed the Prophet and the SARIMA in terms of R-squared, RMSE and WMAPE measures. The results of this study confirmed the argument that deep learning frameworks do perform better for sales forecasting tasks. As mentioned before, metrics for the training set is calculated using 80% of the data, and for the test set's metrics, 20% of the data is used. Amongst the trained models, Prophet performed the worst with a testing R-squared of 0.8243; however, having lower RMSE and WMAPE than the SARIMA model. The lowest error metrics for the test set are achieved with the LSTM model where R-squared, RMSE and WMAPE were 0.9113, 0.0763 and 0.1623, respectively. As this study trained the SARIMA model as the baseline model, implementation of the Prophet model showed worse results than expected. However, since the LSTM model performed the best, the study confirmed the superiority of deep learning frameworks over statistical forecasting methods. The time series plot of the predicted values of the proposed models for the test data is presented in Figure 4.

Table 2
Prediction performance of the proposed models for training and testing data

	Train Set			Test Set		
	R-squared	**RMSE**	**wMAPE**	**R-squared**	**RMSE**	**wMAPE**
LSTM	0.9643	0.0456	0.1040	0.9113	0.0763	0.1623
SARIMA	0.9388	0.0592	0.1314	0.8791	0.0952	0.2216
Prophet	0.8724	0.0805	0.1849	0.8243	0.0941	0.1939

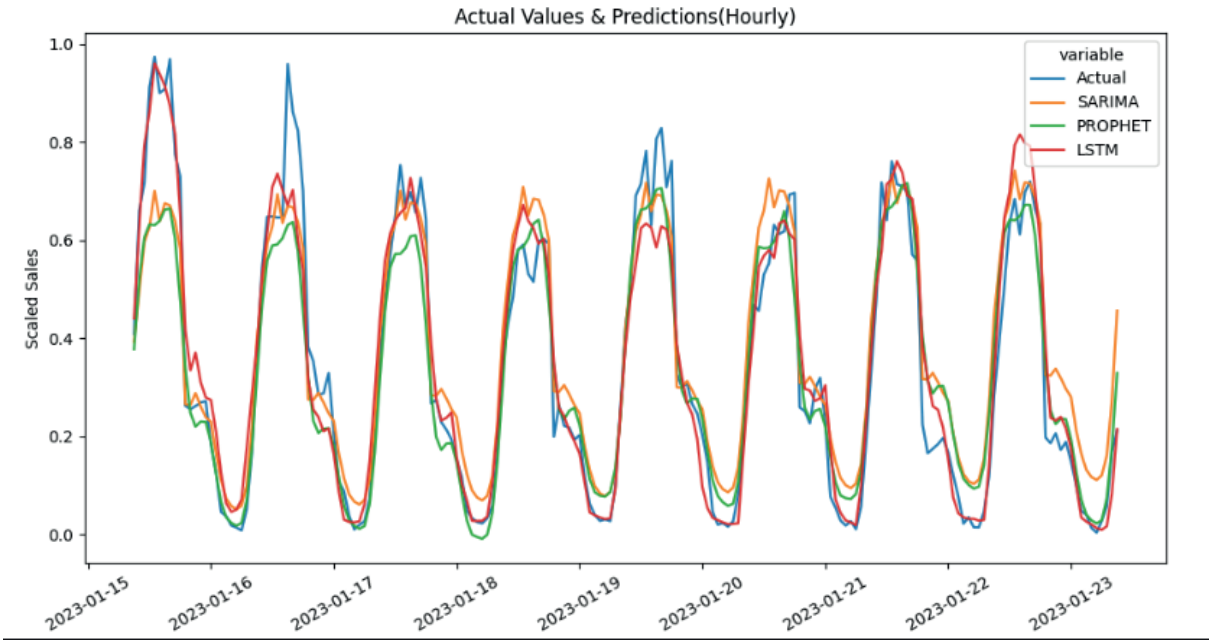

Figure 4. Time series plot of the proposed models for testing data

6. CONCLUSION AND DISCUSSION

In this study, Long-Short Term Memory (LSTM) Networks and the Prophet algorithm are compared with classic statistical approaches in terms of the aforementioned performance metrics using a real-world e-commerce data set. According to the numerical results, the LSTM model outperformed the other models in terms of prediction accuracy. These results showed the superiority of deep learning frameworks over statistical forecasting methods. Regarding the future work on this topic, despite the fact that parameter tuning is already done at the application stage, there is still some room for improvement in optimizing hyper-parameters. Future studies can also focus on using different metaheuristic algorithms and/or increasing the size of the dataset. Especially for deep learning models, a larger dataset tends to improve the model performance. The results of this study may be improved by using a bigger dataset and a more complex neural networks architecture. In this study, weekdays, holidays, weather conditions and temperature are chosen as external variables; however, new features can be added, like campaign and price information, to model extreme points in the data which may lead to an increase in the model's performance. The same method used in this study can also be used to predict the sales of particular product items. Predicting the sales of certain products might have a higher business value which might help industries to plan the supply and stock levels of their product items.

Peer-review: Externally peer-reviewed.
Author Contributions: Conception/Design of Study- A.E., İ.Ö., M.D., T.Ö.; Data Acquisition- A.E., İ.Ö., M.D.; Data Analysis/Interpretation- A.E., İ.Ö., M.D., T.Ö.; Drafting Manuscript- A.E., İ.Ö., M.D.; Critical Revision of Manuscript- T.Ö.; Final Approval and Accountability- A.E., İ.Ö., M.D., T.Ö.; Material and Technical Support- A.E., İ.Ö., M.D.; Supervision- T.Ö.

Conflict of Interest: The authors have no conflict of interest to declare.
Grant Support: The authors declared that this study has received no financial support.

REFERENCES

Arunraj, N. S., & Ahrens, D. (2015). A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics*, *170*, 321–335. https://doi.org/10.1016/j.ijpe.2015.09.039

Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., & Seaman, B. (2019). Sales Demand Forecast in E-commerce Using a Long Short-Term Memory Neural Network Methodology. In T. Gedeon, K. W. Wong, & M. Lee (Eds.), *Neural Information Processing* (pp. 462–474). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-36718-3_39

Chandriah, K. K., & Naraganahalli, R. V. (2021). RNN / LSTM with modified Adam optimizer in deep learning approach for automobile spare parts demand forecasting. *Multimedia Tools and Applications*, *80*(17), 26145–26159. https://doi.org/10.1007/s11042-021-10913-0

Chang, P.-C., Liu, C.-H., & Fan, C.-Y. (2009). Data clustering and fuzzy neural network for sales forecasting: A case study in printed circuit board industry. *Knowledge-Based Systems*, *22*(5), 344–355. https://doi.org/10.1016/j.knosys.2009.02.005

Choi, T.-M., Hui, C.-L., Liu, N., Ng, S.-F., & Yu, Y. (2014). Fast fashion sales forecasting with limited data and time. *Decision Support Systems*, *59*, 84–92. https://doi.org/10.1016/j.dss.2013.10.008

Delasalles, E., Lamprier, S., & Denoyer, L. (2019). Dynamic Neural Language Models. In T. Gedeon, K. W. Wong, & M.

Lee (Eds.), *Neural Information Processing* (pp. 282–294). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-36718-3_24

Ensafi, Y., Amin, S. H., Zhang, G., & Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning – A comparative analysis. *International Journal of Information Management Data Insights*, *2*(1), 100058. https://doi.org/10.1016/j.jjimei.2022.100058

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Sequence Modeling: Recurrent and Recursive Nets. In *Deep learning* (pp. 373–420). Cambridge, Massachusetts: The MIT Press.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). Linear Regression. In *An introduction to statistical learning: With applications in R* (pp. 59–128). New York: Springer.

Ji, S., Wang, X., Zhao, W., & Guo, D. (2019). An Application of a Three-Stage XGBoost-Based Model to Sales Forecasting of a Cross-Border E-Commerce Enterprise. *Mathematical Problems in Engineering*, *2019*, 1–15. https://doi.org/10.1155/2019/8503252

Jing, X., & Lewis, M. (2011). Stockouts in Online Retailing. *Journal of Marketing Research*, *48*(2), 342–354. https://doi.org/10.1509/jmkr.48.2.342

Loureiro, A. L. D., Miguéis, V. L., & da Silva, L. F. M. (2018). Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decision Support Systems*, *114*, 81–93. https://doi.org/10.1016/j.dss.2018.08.010

Martínez, F., Charte, F., Frías, M. P., & Martínez-Rodríguez, A. M. (2022). Strategies for time series forecasting with generalized regression neural networks. *Neurocomputing*, *491*, 509–521. https://doi.org/10.1016/j.neucom.2021.12.028

Punia, S., Nikolopoulos, K., Singh, S. P., Madaan, J. K., & Litsiou, K. (2020). Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail. *International Journal of Production Research*, *58*(16), 4964–4979. https://doi.org/10.1080/00207543.2020.1735666

Sun, Z.-L., Choi, T.-M., Au, K.-F., & Yu, Y. (2008). Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems*, *46*(1), 411–419. https://doi.org/10.1016/j.dss.2008.07.009

Taylor, S. J., & Letham, B. (2017). *Forecasting at scale* (No. e3190v2). PeerJ Inc. https://doi.org/10.7287/peerj.preprints.3190v2

Weytjens, H., Lohmann, E., & Kleinsteuber, M. (2021). Cash flow prediction: MLP and LSTM compared to ARIMA and

Prophet. *Electronic Commerce Research*, *21*(2), 371–391. https://doi.org/10.1007/s10660-019-09362-7

Yu, Q., Wang, K., Strandhagen, J. O., & Wang, Y. (2018). Application of Long Short-Term Memory Neural Network to

Sales Forecasting in Retail—A Case Study. In K. Wang, Y. Wang, J. O. Strandhagen, & T. Yu (Eds.), *Advanced Manufacturing and Automation VII* (pp. 11–17). Singapore: Springer. https://doi.org/10.1007/978-981-10-5768-7_2

Yu, Y., Choi, T.-M., & Hui, C.-L. (2011). An intelligent fast sales forecasting model for fashion products. *Expert Systems with Applications*, *38*(6), 7373–7379. https://doi.org/10.1016/j.eswa.2010.12.089

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159-175. https://doi.org/10.1016/S0925-2312(01)00702-0

Zhao, K., & Wang, C. (2017, August 26). *Sales Forecast in E-commerce using Convolutional Neural Network*. arXiv. https://doi.org/10.48550/arXiv.1708.07946

Zohdi, M., Rafiee, M., Kayvanfar, V., & Salamiraad, A. (2022). Demand forecasting based machine learning algorithms on customer information: An applied approach. *International Journal of Information Technology*, *14*(4), 1937–1947. https://doi.org/10.1007/s41870-022-00875-3