Price Forecasting with Deep Learning in Business to Consumer Markets

Emre Eğriboz[1](✉) and Mehmet S. Aktaş[2]

[1] Information Technologies Institute, Informatics and Information Security Research Center, TÜBİTAK, Kocaeli, Turkey
emre.egriboz@tubitak.gov.tr

[2] Computer Engineering Department, Yildiz Technical University, Istanbul, Turkey
aktas@yildiz.edu.tr

Abstract. Price forecasting is a challenging and essential problem studied in different markets. Many researchers and institutions, academically and professionally, develop future price forecasting techniques. This study proposes a data collection and processing pipeline to forecast the next day's price of a product in business to consumer (B2C) markets using the price data obtained from web crawlers, preprocessing steps, the deep features produced by the autoencoder, and the technical indicators. For this purpose, we use web crawlers to collect different airline companies' ticket prices daily and create a price index. We apply the discrete wavelet transform (DWT) preprocessing method to denoise the price index data, calculate some technical indicators analytically, and extract the deep features of the price data via three different autoencoders, linear, stacked linear, and long short term memory (LSTM). An LSTM forecaster generates forecasts using deep and calculated features. Finally, we measure the effects of autoencoder types, and mentioned features on the forecasting performance. Our study shows that using LSTM autoencoder on denoised time series price data with technical indicators in B2C markets yields promising results.

Keywords: Deep learning · Feature extraction · Time series · Business to consumer market

1 Introduction

In the business-to-consumer (B2C) trade model, businesses market their services or products to many buyers. This commercial model can be carried out either face-to-face or online with the dramatic increase in e-commerce sales opportunities recently. Positive developments in e-commerce show that this shopping model has become widespread in recent years and will reach more consumers in the coming years [1]. The volume of e-commerce in Turkey increased by 64% in the first half of 2020 compared to the same period of the previous year and reached 91 billion 700 million Turkish Liras [2]. E-commerce is becoming more

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12954, pp. 565–580, 2021.
https://doi.org/10.1007/978-3-030-86979-3_40

and more common among individuals of different ages. The proportion of e-commerce users over the age of 65 increased from 6% to 10% during the COVID-19 pandemic [3]. In such an environment, forecasting the price of a product in the next period has direct or indirect effects. It is an important research subject in terms of direct reasons such as purchasing decision, choice of the seller, procurement, price determination, and indirect reasons such as strategic planning from sales volume information [4–8].

In the field of tourism, which is an example of B2C with many service buyers and few sellers, price depends on many variables' effects such as demand, exchange rate, inflation, season, or supply [9,10].

Machine learning algorithms are often used to forecast the value of the following time period on price data [11]. The methods used in the deep learning approach, an expert in feature extraction and a sub-branch of machine learning, have achieved more successful results than classical machine learning methods in recent years [10]. Price forecasting studies with deep learning are generally used in consumer to consumer (C2C) or markets where stock movements are determinant [12]. We observe that there is a lack of studies in the literature on price forecasting in B2C markets. Furthermore, there exists studies that analyze user-system interaction data [13–23]. These studies mainly focus on data lineage of the user' s interactions for various reasons such as debugging data and transformations, auditing, evaluating the quality and trust in data. We also observe that there is lack of studies on analyzing the user-system interactions in tourism sector for the purpose of price forecasting.

In the tourism sector, where high capital is required in its establishment and management, it is necessary for businesses to forecast the future price of a product or service with deep learning supported solutions [10,24].

We perform data collection, data preprocessing, data processing, data analysis, and analysis result evaluation within the scope of this research to meet the forecasting requirement. A prototype application developed runs to demonstrate the performance of the proposed approach with different settings. This application includes experiments based on 158 days with price data from three different airline companies in Turkey. The results obtained from the methodology proposed within the scope of this study show that deep feature extraction with the LSTM autoencoder and technical indicators effectively reduces the forecasting loss in B2C markets. These results obtained confirm the proposed methodology.

Other parts of the article have been arranged as follows. Firstly, in Sect. 2, the requirements to start the study and the research questions arising from these requirements are introduced. Section 3 includes a literature review where related works are discussed. The methodology we propose for answering research questions is introduced in Sect. 4. The details of the data set used, the details of the experimental design, and the experimental results are included in Sect. 5, while the technical information about the prototype application is presented in Sect. 6. In the 7th Section, the answers given in the article to the research questions of the experimental study are presented collectively. Finally, in Sect. 8, some results are reported, and directions for future works are given.

2 Research Questions

Forecasting the future price of a product or service in B2C markets is very important, especially in sectors such as tourism, where demand forecasting is strategically vital for organizations. After the price data obtained, their noise should be removed by preprocessing studies. Technical indicators calculated used to forecast prices in stock markets of similar interest are useful auxiliary features for forecasting the price of a product in the future period. In addition to these features, the use of deep features that can be obtained from price data with deep learning methods for price forecasting in B2C markets can be beneficial for getting more accurate results. Therefore, the data should be prepared for processing by taking analytical methods and feature extraction methods as examples in similar studies. Finally, a deep learning model should be trained to create a price forecasting model, and then the trained model should be tested.

In line with these requirements, answers to the following research questions are sought in this study.

1. In B2C markets, can the historical price data and deep learning algorithms be used for forecasting the price of the next period of a product or service?
2. Which preprocessing methods can be used on price data?
3. Which deep learning methods can be used to extract features using price data?
4. Are technical indicators that can be calculated from price data useful in forecasting the next step ahead of price in B2C markets?

3 Literature Review

Discrete wavelet transform (DWT) method is used to transform a discrete time series signal into discrete meaningful components called wavelets that make up it [25]. It is frequently used as a preprocessing function to remove noise in the field of signal processing. A.J. Conejo et al. used the DWT method as a noise suppressor on electricity price data in their studies [26].

Autoencoder is an unsupervised deep learning approach used to learn domain data patterns and represent them with an artificial neural network model. The stacked autoencoder (SAE) is a derivative of autoencoders created by sequentially linking the encoded outputs. W. Bao et al. used stacked autoencoders on stock price data to reduce forecast loss [12]. L. Wang et al. used stacked autoencoders and their variants in short-term electricity price forecasting to reduce forecast loss [27]. The autoencoder and its derivative techniques improve the experimental performance in the studies by processing the domain data to make it more qualified.

Deep learning, a sub-branch of machine learning study area, learns by extracting high-level features using layers of artificial neural networks one or more times. Long short-term memory (LSTM) is a deep learning architecture with specialized recurrent neural networks (RNNs) for processing sequential data, thanks to feedback links [28]. J. Cao et al. got successful forecasting results using LSTM in their financial time series price data forecasting studies [29]. S. Bouktif et al. achieved successful forecasting results using LSTM in their study of electricity load forecasting using France metropolitan's electricity consumption data [30].

The common feature of these studies and our study, time series price data, shows that the DWT and autoencoder methods are useful in our research. Inspired by these studies, we use different autoencoder kinds in our study. In addition, the successful results obtained in the forecast of time series data in previous studies using LSTM show that the forecasting infrastructure in our study can be performed using LSTM.

4 Proposed Methodology

In tourism markets, since the service's sale is possible over the internet, it is possible to access price data from the sellers' reservation sites easily. Automatic web crawler scripts collect the data set used in this research on a daily basis by querying the price of the next day in the ticket sales system of three different airline companies. The collected data are included in the training of the LSTM forecaster model after passing through preprocessing, technical indicator calculation, DWT, and autoencoder, respectively, in the data processing workflow. The proposed data collection and processing workflow architecture is illustrated in Fig. 1.

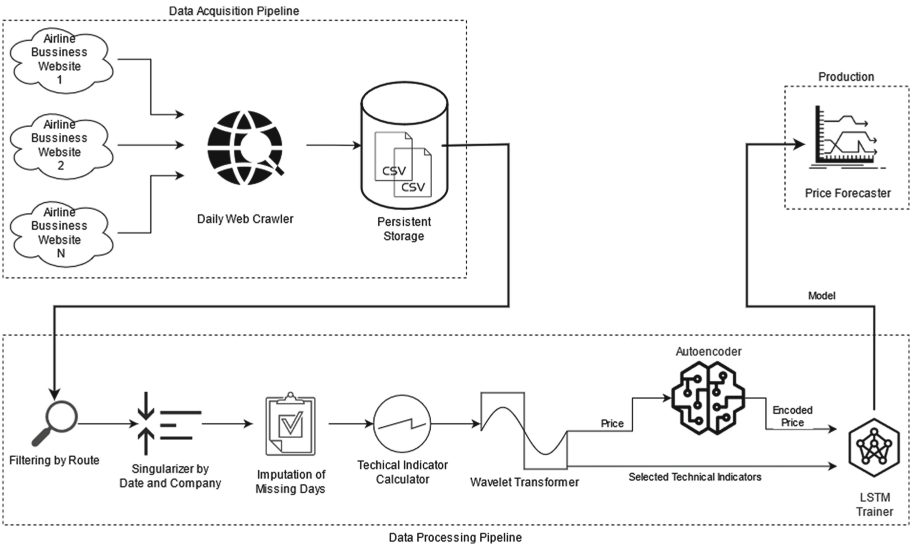

Fig. 1. Proposed data collection and processing architecture.

4.1 Daily Web Crawling

The company website where the reservation can be made online supplies the final price determined for the customer's purchase. However, during the data collection phase, this job should be done automatically on a daily basis, specific to the designated departure and landing locations and for the next day. It is very practical to use web crawlers to automate this job. The data collected, stored daily in comma-separated files (CSV), includes the cities of departure and landing airports, ticket prices, and date information.

4.2 Preprocessing

Collected data is read batch-wise from files and pass through some preprocessing steps. The first step in preprocessing is to filter the batch data based on their departure and landing location. Then, the collected data are averaged according to the date and company criteria to ensure that they are singularized from the date and the airline company criteria. In this way, only one price record per day remains from each company. Preprocessing continues with the imputation of the missing days, if any, by an average of the same day of the previous and next week and creating the daily price index. The daily price index creation process is the sum of the average ticket price of all companies on that day. Since our study is similar to the various previous studies about using price data, we apply the discrete wavelet transform to the tourism price data at the last step of the preprocessing.

4.3 Technical Indicator Calculation

It is possible to calculate some well-known technical indicators using the price index data introduced in the preprocessing section. We accept the index price we calculated as the close price for that day and using this close price, the technical indicators whose abbreviations are given in Table 1 are calculated. Please note that only the close price is used to calculate these technical indicators.

Table 1. Abbreviations of technical indicators used.

Indicator name	Description
MACD	Moving Average Convergence Divergence
BOLL	Bollinger Bands index
EMA20	Exponential Moving Average for 20 days
MA5	Moving Average for 5 days
MA10	Moving Average for 10 days
ROC	Rate of Change

4.4 Discrete Wavelet Transform

Discrete wavelet transform smooths out the outliers in the time-dependent price data and reduces the data's noise. S. Mallat proposed the wavelet representation method to decompose a multilevel signal of content at a specific resolution [31]. In this method, discrete approximations considered to belong to the main signal are calculated by passing through low pass and high pass filters. The low pass filter gives the low-frequency component (ϕ), the high pass filter gives the high-frequency component (ψ). Equation (1) and (2) formulate decomposed mother and father wavelets at the J level [32]. Generally, the father wavelet refers to a low-frequency component or approximate coefficients, and also the main wavelet refers to high-frequency components or detailed coefficients.

$$\phi_{j,k} = 2^{-j/2}\phi(\frac{t-2^j k}{2^j}), \quad \psi_{j,k} = 2^{-j/2}\psi(\frac{t-2^j k}{2^j}) \tag{1}$$

$$\int \phi(t)dt = 1, \quad \int \psi(t)dt = 0 \tag{2}$$

A time dependent $f(t)$ function is defined and projected onto low-frequency and high-frequency wavelets, as in Eq. (3).

$$\begin{aligned} s_{J,k} &= \int \phi_{J,k}(t)f(t)dt, \quad d_{j,k} = \int \psi_{j,k}(t)f(t)dt \\ where\ j &= \{1,2,...,J\}, \quad s = 2^j, \quad k = \{1,2,...\} \end{aligned} \tag{3}$$

The sequence $\{S_j, D_j, D_{j-1}, ..., D_1\}$, which stands for $f(t)$, which is the sum of approximate coefficients (S_j) and detailed coefficients (D_j), is expressed as in Eq. (4).

$$f(t) = \sum_k s_{J,k}\phi_{J,k}(t) + \sum_k d_{J,k}\psi_{J,k}(t) + \sum_k d_{J-1,k}\psi_{J-1,k}(t) + ... + \sum_k d_{1,k}\psi_{1,k}(t) \tag{4}$$

Finally, with the resulting equation $f(t)$, wavelet transform is applied on time series data.

4.5 Autoencoder

Machine learning algorithms are classified as supervised and unsupervised. While unsupervised algorithms do not need labels while working, supervised algorithms work with ground truth and increase model performance in this way. Deep learning architectures are learning structures obtained by connecting different types of layers one after another in various numbers and shapes. Autoencoders are unsupervised deep learning architectures typically used to extract features or remove noise by learning domain data. An example autoencoder constructed with an encoder and a decoder is illustrated in Fig. 2.

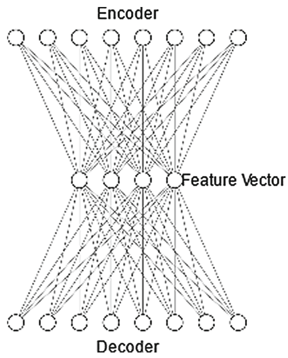

Fig. 2. An example autoencoder consists of an encoder and a decoder layer.

Autoencoder architectures vary depending on the problem and the data they are used. Autoencoder in different architectures is used in the feature extraction of price data in different markets. W. Bao et al. used the stacked autoencoder architecture to reduce the noise in the data in their study of estimating financial time series [12]. In the study, the stacked autoencoder removed the noise of the time series data and extracted the deep features. This study shows that the data in time series format is feature engineered by a stacked autoencoder built with feed-forward linear layers. The stacked autoencoder for the study is illustrated in Fig. 3.

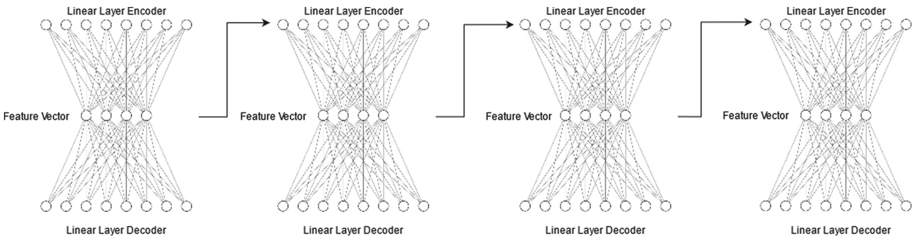

Fig. 3. A stacked autoencoder consists of 4 feed-forward linear autoencoders.

LSTM is an excellent feature extractor for time series data. In the literature, we see that this type of autoencoder is often used for the machine translation problem; however, there is a lack of its use with time series price data. We propose the sequence-to-sequence method formed by LSTM units as a feature extractor and noise suppressor. Note, LSTM is frequently used with time series data as a forecaster. The autoencoder we recommend is illustrated in Fig. 4.

From RNN to LSTM. Recurrent neural networks, unlike feed-forward networks, contain memory cells that hold the state information corresponding to

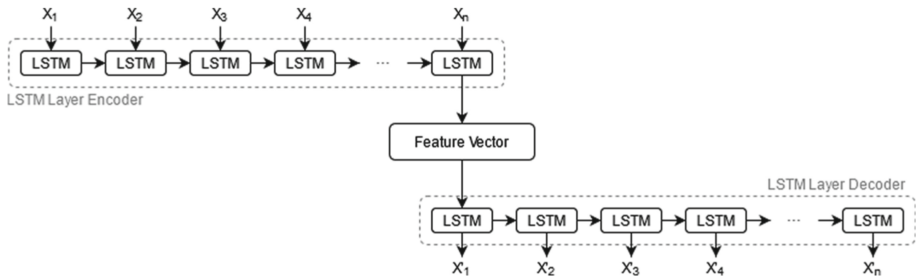

Fig. 4. An LSTM autoencoder with unrolled units.

the previous iteration. In addition to input data, state information is provided as input to the cell, providing a loop within the network. The architecture of an RNN cell is illustrated in Fig. 5. x_t in the figure represents the input vector at time t; s_t and s_{t-1} represents the state vector calculated from the input vector at t and $t-1$; W and U expressions express the weights of the input vector. The formula s_t is given in Eq. (5). The activation function f, which is used to limit the output to a certain range; there may be special functions such as tanh, sigmoid, ReLu [33].

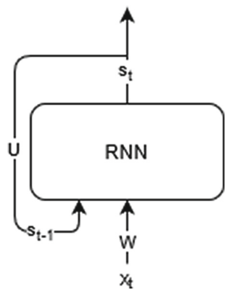

Fig. 5. Architecture of RNN unit.

$$s_t = f(Wx_t + Us_{t-1}) \tag{5}$$

LSTM is a special variant of RNN that solved the vanishing gradient problem. An LSTM unit contains the RNN memory as well as the input gate that controls the update of the cell state, the output gate that controls the value to be sent to the next cell, and the forget gate that checks the previous state of the cell in the cell [28,33]. The architecture of an LSTM cell is shown in Fig. 6.

The f_t, g_t, q_t expressions given in Eq. (6) indicate the values for the forget gate, input gate, and output gate, respectively. X_t is the input vector at time t; h_{t-1} is the output vector at time $t-1$; The U and W vectors represent the

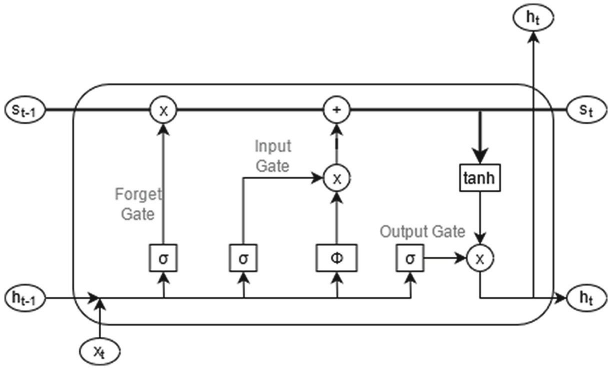

Fig. 6. Architecture of an LSTM unit.

input and recurring weights held for the respective gate. The activation function σ is the sigmoid function that returns the output between 0 and 1.

$$\begin{aligned} f_t &= \sigma(U_f X_t + W_f h_{t-1}), \\ g_t &= \sigma(U_g X_t + W_g h_{t-1}), \\ q_t &= \sigma(U_o X_t + W_o h_{t-1}) \end{aligned} \tag{6}$$

The expression s_t given in Eq. (7) gives the state value for the LSTM cell; The expression h_t refers to the output vector at the time t. U and W vectors are input and recurring weights retained for the LSTM cell; ϕ represents the input activation function (usually sigmoid or tanh) [28,33].

$$\begin{aligned} s_t = f_t s_{t-1} + g_t \phi(U X_t + W h_{t-1}), \\ h_t = tanh(s_t) q_t \end{aligned} \tag{7}$$

4.6 LSTM Forecaster

In general, our problem is regression-based as we try to forecast the next day's ticket sale price. The autoencoder section explains how the autoencoder extracts deep features of the price data in a period. We use the LSTM forecaster to forecast the next day's price using deep encoded features and calculated technical indicators.

An LSTM forecaster usually consists of a layer containing a certain number of LSTM units and a dense layer that will produce the result attached to it; this architecture is shown in Fig. 7. The matrix used as inputs to the LSTM forecaster is the windowed data, which data can be raw or obtained from an autoencoder's encoder layer. If the LSTM autoencoder is used, the hidden states vector of the units of the autoencoder is also provided as the input vector to the forecaster (See Fig. 7: Hidden States). The output of the dense layer is calculated as in Eq. (8). The expression x in Eq. (8) represents the output matrix of the LSTM layer; W indicates the weight matrix of the dense layer; f is the activation function.

$$output = f(Wx) \tag{8}$$

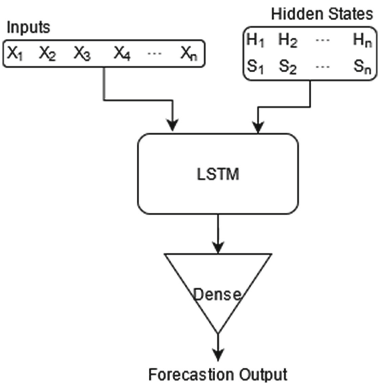

Fig. 7. An LSTM forecaster takes the windowed input matrix and the extra hidden states matrix as inputs.

5 Evaluation

This section shares the details of the data set used, the design of the experimental setup, and the experimental results.

5.1 Data Set

Web crawlers collect the training and test data set daily for domestic flight ticket prices in TRY currency between 27.07.2020 and 31.12.2020 (158 days). We filter the route as flights from the most populous city of Turkey, Istanbul, to Ankara, capital of Turkey and the second-most populous city. The route has a total of 4544 flight records from Istanbul to Ankara. Since the daily price index is created for the route after preprocessing, the data set is reduced to a total of 158 records.

5.2 Experimental Design

We set up two different experimental setups to answer research questions. The first experiment compares the performance of wavelet transform and autoencoders. The second experiment measures the contribution of the calculated technical indicators. Daubechies (db4) wavelet transform function is applied with 1 level decomposition to the data in preprocessing phase. Window size for all experiments is determined as five days by trial and error. In all experiments, the sigmoid function is used as the activation function, and Adam, a special stochastic gradient descent (SGD) function, is used as the optimizer. The LSTM forecaster layer is the same for all experiments. The LSTM layer contains 100 hidden units and then connects to the dense layer to produce the forecasting result. The first 67% portion (105 days) of the 158 days data is used for training and the last 33% portion (53 days) for the test.

In the first experiment, using and not using the wavelet transform run as separate configurations. There are four different configurations for the autoencoder; single layer linear autoencoder (Single AE), stacked linear autoencoder (SAE), LSTM autoencoder (LSTM AE), and forecasting without autoencoder. Thus, the first experiment results in eight different combinations. The SAE is obtained by connecting four consecutive single layer linear autoencoders.

The second experiment investigates which of the calculated technical indicators are appropriate to use together with the price data. All subsets of technical indicators run as separate configurations to determine the technical indicators that will provide the best benefit when used with price data. The second experiment produces a possible result of $2^6 = 64$, considering six different technical indicators and one fixed feature. Since we search the technical indicators that will provide the most benefit in our study, we share the best result of the group with n and the technical indicators belonging to the group.

The encoded feature vector size of single layer, stacked, and LSTM autoencoders is determined as 16 by trial and error.

Performance Metric. The performances of the implemented forecaster models are measured in terms of root mean square error (RMSE) and mean absolute error (MAE) metric, which are frequently used in regression problems.

5.3 Experimental Results

In the configurations introduced in the experimental design section, measurements are made with test data according to the metrics specified in the performance metric subsection.

In the first experiment, the LSTM autoencoder performed better than other autoencoders in the RMSE metric, and it seems beneficial to apply DWT regardless of the autoencoder used. Evaluation results of the Experiment 1 are presented in Table 2.

The second experiment shows that the most successful model is generated according to the RMSE metric when the technical indicators MACD, BOLL, MA5, MA10 are used in the group of 5 combinations. It is possible to make a similar observation in the MAE metric. The striking point here is observed when the second most successful model is examined. The second most successful model is created using the EMA20 and ROC technical indicators, which are the complete exclusions of the indicators used in the most successful model.

Table 2. Evaluation results of Experiment 1.

	RMSE		MAE	
	With DWT	Without DWT	With DWT	Without DWT
LSTM AE	0.0769	0.1285	0.0596	0.1032
SAE	0.0880	0.1191	0.0631	0.0916
Single AE	0.0815	0.1204	0.0586	0.0839
No AE	0.0782	0.1242	0.0566	0.0853

Table 3. LSTM AE encoded evaluation results of Experiment 2.

Feature count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
1	[close]	0.0769	[close]	0.0596
2	[close,roc]	0.0734	[close,ma10]	0.0562
3	[close,ema20,roc]	0.0698	[close,ema20,roc]	0.0521
4	[close,macd,boll,ema20]	0.0716	[close,ema20,ma10,roc]	0.0539
5	[close,macd,boll,ma5,ma10]	0.0684	[close,macd,boll,ma5,ma10]	0.0522
6	[close,macd,boll,ema20,ma5,ma10]	0.0756	[close,macd,boll,ema20,ma5,ma10]	0.0553
7	[close,macd,boll,ema20,ma5,ma10,roc]	0.0797	[close,macd,boll,ema20,ma5,ma10,roc]	0.0553

Hence, it is possible to say that the use of technical indicators achieves the highest benefit when divided into two groups. The first group consists of MACD, BOLL, MA5, and MA10 indicators, while the second group consists of EMA20 and ROC indicators; of course, the close price is included in both groups. Evaluation results of the Experiment 2 are presented in Table 3.

6 Prototype Application

The web crawler application used to collect data runs with Python 3.8.5 using the Scrapy 2.3 package. Web crawler application collects ticket prices, together with search keys and date information, from the online ticket search portal for Pegasus, Turkish Airlines, and Atlas Jet companies. The departure and landing cities are used as the search key, and the day after the web crawler is running, as the departure date.

The data processing application is implemented in Ubuntu 16.04 operating system and Python 3.6.9 using the Keras 2.4.3 application infrastructure with TensorFlow 2.2.

The system used works with "Intel (R) Xeon (R) Gold 6138 CPU @ 2.00 GHz" processor, 504 GB memory, and 4 * "Tesla V100-SXM2-16GB" graphics processors.

7 Experimental Study

This section gives the answers to the questions in the research questions section with the literature review results and the proposed methodology implementation's results.

Similar studies in the literature review section confirm that deep learning can make price forecasting to next step ahead on time series data of a product or service. Since the answer to the first research question is positive, studies are continued, and answers to other questions are sought.

The second research question, the selection of the preprocessing method, is introduced in the proposed methodology section, and its effect on the result is presented in the experimental results section. Accordingly, the positive contribution of the DWT, which is the preprocessing method for removing the noise and smoothing in the data, is shown in the experimental results section.

The third research question, in which deep learning methods can be used for feature extraction of price data, is answered as linear, stacked linear, and LSTM autoencoders. We show their performances comparatively in the experimental results section.

The answer to the last question, which is investigating the benefit of technical indicators in B2C markets, is presented in the experimental results section with the comparison table made with the subsets of all technical indicators. It is beneficial to use subsets of technical indicators for price forecasting in B2C markets.

Please note that the validity of the proposed techniques and all results obtained depends on the data. Different implementations or different data may produce different results.

8 Conclusion and Future Work

Forecasting the next step ahead of price in different kinds of markets is being more critical. It is strategically important to forecast the next price in B2C markets dominated by competitive service providers, as well as in C2C markets where consumer transactions are determinant. Eventually, various preprocessing steps should be applied, and the domain data features should be extracted to reduce the forecasting error. The problem of extracting deep and calculated features and using them with different combinations at different stages to minimize the forecasting error is always open to improvement.

We investigate the benefits of the DWT preprocessing method and the use of calculated technical indicators in B2C markets. We also test the effect of the LSTM autoencoder in the feature extraction phase on the time series price data. Our benchmarking results for forecasting airfare prices in the B2C market show that applying DWT in preprocessing, and using a subset of technical indicators, and using LSTM autoencoder for feature extraction succeed the forecaster to better results than not applying DWT, or not using technical indicators, or using other autoencoder types. The best model is obtained using moving average convergence divergence, Bollinger bands, moving average in 5 and 10 days technical indicators.

In future studies, by extracting price types such as open, high, and low from the raw data, various technical indicators to be calculated can be added to the process. In addition, different feature extractors such as convolutional neural network (CNN) can be used. Finally, even the feature extraction phase can be done using candlestick charts created by new price types extracted.

Acknowledgments. This study is carried out using the data center and web crawler facilities of Cloud Computing and Big Data Research Laboratory (B3LAB) of TÜBİTAK BİLGEM.

References

1. He, P., Zhang, S., He, C.: Impacts of logistics resource sharing on b2c e-commerce companies and customers. Electron. Commer. Res. Appl. **34**, 100820 (2019). https://doi.org/10.1016/j.elerap.2018.100820. https://www.sciencedirect.com/science/article/pii/S1567422318300863
2. 2020 yılı İlk 6 ay e-ticaret verileri açklandı (2020). https://www.eticaret.gov.tr/haberler/10040/detay
3. e-ticaret platformlarında 65 yaş üstü kullanıcı oranı yüzde 6'dan yüzde 10'a çıktı (2021). https://www.aa.com.tr/tr/ekonomi/e-ticaret-platformlarinda-65-yas-ustu-kullanici-orani-yuzde-6dan-yuzde-10a-cikti/2182333
4. Zhao, K., Wang, C.: Sales forecast in e-commerce using convolutional neural network (2017)
5. Hsieh, P.-H.: A study of models for forecasting e-commerce sales during a price war in the medical product industry. In: Nah, F.F.-H., Siau, K. (eds.) HCII 2019. LNCS, vol. 11588, pp. 3–21. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22335-9_1
6. Yan, R., Ghose, S.: Forecast information and traditional retailer performance in a dual-channel competitive market. J. Bus. Res. **63**(1), 77–83 (2010). https://doi.org/10.1016/j.jbusres.2009.02.017. https://www.sciencedirect.com/science/article/pii/S0148296309000460
7. Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., Seaman, B.: Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In: Gedeon, T., Wong, K.W., Lee, M. (eds.) ICONIP 2019. LNCS, vol. 11955, pp. 462–474. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36718-3_39
8. Gürbüz, A., Aktaş, M.S.: Prediction of purchase intention on the e-commerce click-stream data. In: 2019 27th Signal Processing and Communications Applications Conference (SIU), pp. 1–4 (2019). https://doi.org/10.1109/SIU.2019.8806311
9. Fronzetti Colladon, A., Guardabascio, B., Innarella, R.: Using social network and semantic analysis to analyze online travel forums and forecast tourism demand. Decis. Supp. Syst. **123**, 113075 (2019). https://doi.org/10.1016/j.dss.2019.113075. https://www.sciencedirect.com/science/article/pii/S0167923619301046

10. Law, R., Li, G., Fong, D.K.C., Han, X.: Tourism demand forecasting: a deep learning approach. Ann. Tour. Res. **75**, 410–423 (2019). https://doi.org/10.1016/j.annals.2019.01.014, https://www.sciencedirect.com/science/article/pii/S0160738319300143
11. Tseng, K.K., Lin, R.F.Y., Zhou, H., Kurniajaya, K.J., Li, Q.: Price prediction of e-commerce products through internet sentiment analysis. Electron. Commer. Res. **18**(1), 65–88 (2018). https://doi.org/10.1007/s10660-017-9272-9
12. Bao, W., Yue, J., Rao, Y.: A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS ONE **12**(7), e0180944 (2017)
13. Aktas, M., Astekin, M.: Provenance aware run-time verification of things for self-healing internet of things applications. Concurr. Comput. Pract. Exp. (2019). https://doi.org/10.1002/cpe.4263
14. Tufek, A., Aktas, M.S.: On the provenance extraction techniques from large scale log files: a case study for the numerical weather prediction models. In: Balis, B., et al. (eds.) Euro-Par 2020. LNCS, vol. 12480, pp. 249–260. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-71593-9_20
15. Tas, Y., Baeth, M., Aktas, M.: An approach to standalone provenance systems for big social provenance data. In: 2016 12th International Conference on Semantics, Knowledge and Grids (SKG), pp. 9–16 (2016)
16. Riveni, M., Nguyen, T., Aktas, M., Dustdar, S.: Application of provenance in social computing: a case study. Concurr. Comput.: Pract. Exp. **31**(3), e4894 (2019)
17. Baeth, M., Aktas, M.: An approach to custom privacy policy violation detection problems using big social provenance data. Concurr. Comput.: Pract. Exp. **30**(21), e4690 (2018)
18. Baeth, M., Aktas, M.: Detecting misinformation in social networks using provenance data. Concurr. Comput.: Pract. Exp. **31**(3), e4793 (2019)
19. Jensen, S., Plale, B., Aktas, M., Luo, Y., Chen, P., Conover, H.: Provenance capture and use in a satellite data processing pipeline. IEEE Trans. Geosci. Remote Sens. **51**(11), 5090–5097 (2013). https://doi.org/10.1109/TGRS.2013.2266929
20. Tufek, A., Gurbuz, A., Ekuklu, O.F., Aktas, M.S.: Provenance collection platform for the weather research and forecasting model. In: 2018 14th International Conference on Semantics, Knowledge and Grids (SKG), SKG '18, 14th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, IEEE, pp. 17–24 (2018). https://doi.org/10.1109/SKG.2018.00009
21. Yazıcı, I., Karabulut, E., Aktas, M.: A data provenance visualization approach. In: The 14th International Conference on Semantics, Knowledge and Grids (2018)
22. Uygun, Y., Oguz, R., Olmezogullari, E., Aktas, M.: On the large-scale graph data processing for user interface testing in big data science projects. In: IEEE BigData 2020, pp. 2049–2056. IEEE (2020)
23. Olmezogullari, E., Aktas, M.: Representation of click-stream data sequences for learning user navigational behavior by using embeddings. In: In: IEEE BigData 2020, pp. 3173–3179. IEEE (2020)
24. Li, Y., Cao, H.: Prediction for tourism flow based on LSTM neural network. Procedia Comput. Sci. **129**, 277–283 (2018)
25. Chaovalit, P., Gangopadhyay, A., Karabatis, G., Chen, Z.: Discrete wavelet transform-based time series analysis and mining. ACM Comput. Surv. **43**(2), 1–37 (2011). https://doi.org/10.1145/1883612.1883613
26. Conejo, A.J., Plazas, M.A., Espinola, R., Molina, A.B.: Day-ahead electricity price forecasting using the wavelet transform and arima models. IEEE Trans. Power Syst. **20**(2), 1035–1042 (2005). https://doi.org/10.1109/TPWRS.2005.846054

27. Wang, L., Zhang, Z., Chen, J.: Short-term electricity price forecasting with stacked denoising autoencoders. IEEE Trans. Power Syst. **32**(4), 2673–2681 (2017). https://doi.org/10.1109/TPWRS.2016.2628873
28. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep Learning, vol. 1. MIT Press, Cambridge (2016)
29. Cao, J., Li, Z., Li, J.: Financial time series forecasting model based on CEEMDAN and LSTM. Physica A **519**, 127–139 (2019)
30. Bouktif, S., Fiaz, A., Ouni, A., Serhani, M.A.: Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: comparison with machine learning approaches. Energies **11**(7), 1636 (2018)
31. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. **11**(7), 674–693 (1989)
32. Ramsey, J.B., Lampart, C.: The decomposition of economic relationships by time scale using wavelets: expenditure and income. Stud. Nonlinear Dyn. Econ. **3**(1) (1998)
33. Chollet, F., et al.: Deep Learning with Python, vol. 361. Manning, New York (2018)