REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

PRICE FORECASTING WITH DEEP LEARNING IN BUSINESS TO CONSUMER MARKETS

Emre EĞRİBOZ

MASTER OF SCIENCE THESIS
Department of Computer Engineering
Program of Computer Engineering

Supervisor
Assoc. Prof. Dr. Mehmet S. AKTAŞ

December, 2022

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

PRICE FORECASTING WITH DEEP LEARNING IN BUSINESS TO CONSUMER MARKETS

A thesis submitted by Emre EĞRİBOZ in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 29.12.2022 in Department of Computer Engineering, Program of Computer Engineering.

Assoc. Prof. Dr. Mehmet S. AKTAŞ
Yildiz Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Mehmet S. AKTAŞ, Supervisor
Yildiz Technical University

Prof. Dr. Ahmet SAYAR, Member
Kocaeli University

Asst. Prof. Dr. Yunus Emre SELÇUK, Member
Yildiz Technical University

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Price Forecasting with Deep Learning in Business to Consumer Markets supervised by my supervisor, Assoc. Prof. Dr. Mehmet S. AKTAŞ. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Emre EĞRİBOZ

Signature

To My Precious Family

ACKNOWLEDGEMENTS

During the preparation phase of the project; I would like to thank my Honorable Professor, Associate Professor Dr. Mehmet S. AKTAŞ, for his unwavering support with his valuable ideas and guidance, the members of Yıldız Technical University Computer Engineering Department, which I am proud to be a part of, and My Family, whose support I have always felt throughout my life.

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

Emre EĞRİBOZ

TABLE OF CONTENTS

LIST OF SYMBOLS

$f(x)$	Activation function of the input x
$\Phi(x)$	Activation function of the input x for input in the long short-term memory unit
S	Approximate coefficients
D	Detailed coefficients
$\hat{y}$	Forecasting result
ψ	High-frequency discrete signal
$tanh(x)$	Hyperbolic tangent activation function of the input x
x	Input
β_0	Intersection point with the y-axis
J	Level of wavelet
$T2$	Long-term period
ϕ	Low-frequency discrete signal
max_F	Maximum value of feature F
$median(X)$	Median function of the vector X
min_F	Minimum value of feature F
α	Model weight in autoregression
new_max_F	New maximum value of feature F
new_min_F	New minimum value of feature F
x'	New value for the input
m	Number of elements tested
r	Order in autoregression
y	Output, result
h	Output vector in the long short-term memory unit

T	Period
P	Price
$T1$	Short-term period
$\sigma(x)$	Sigmoid activation function of the input x
β_1	Slope of the line
s	State vector in the recurrent neural network and long short-term memory units
t	Time
$g(t)$	Time dependent function
f	Value of forget gate
g	Value of input gate
q	Value of output gate
Z_t	Value of the random noise function
$\Delta(i, j)$	Weight function based on the distance between i and j
V	Weight vector for the gates and long short-term memory unit
W	Weight vector for the input vector
U	Weight vector for the state vector
WS	Window size

LIST OF ABBREVIATIONS

AE	Autoencoder
B2C	Business to Consumer
BIST	Borsa İstanbul
CNN	Convolutional Neural Network
coif	Coiflet
CSV	Comma Separated Values
db	Daubechies
DOM	Document Object Model
DWT	Discrete Wavelet Transform
EMA	Exponential Moving Average
GB	Gigabyte
HTML	Hypertext Markup Language
LSTM	Long Short-Term Memory
MA	Moving Average
MACD	Moving Average Convergence/Divergence
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROC	Rate of Change
SAE	Stacked Autoencoder
sym	Symlet
TL	Turkish Lira
TRIX	Triple Exponential Moving Average

XML	Extensible Markup Language
XPath	XML Path Language

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Price Forecasting with Deep Learning in Business to Consumer Markets

Emre EĞRİBOZ

Department of Computer Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Mehmet S. AKTAŞ

Price forecasting is an approximation of the financially determined value of a product or service subject for sale in markets. Depending on the problem, price forecasting, which is at different time resolutions, can be made for future times in the short, medium, and long terms. In addition, price forecasting is a challenging, important, and fundamental problem for which academic or professional studies are carried out for various markets by lots of researchers and businesses.

Previous studies in the literature for price forecasting in different markets use various traditional and deep learning methods. Researchers create forecaster models to forecast by using historical data regardless of the method in these works.

In our study, we propose a methodology that includes data collection and data processing pipelines for price index forecasting using historical data and deep learning-based methods. We implement the proposed methodology by developing data collection and advanced analytical applications. We run our pipelines using airline ticket prices marketed to consumers by businesses in the airline passenger transport industry, which is an example of a business-to-consumer (B2C) market.

In the data collection application, we collected the ticket price data of different airline companies daily for a while with a web scraper. In the data processing application, we deduplicated the collected time series data, imputed missing days, and created a price index in the preprocessing step; then we calculated technical indicators using mathematical methods; we removed noises; we extracted deep features with various autoencoders; then we generated price forecasting models with a deep learning network; in the end, we measured the effects of noise reduction, autoencoder types and technical indicators on performance by testing the produced models. Our experiments show that the use of deep features generated by autoencoders and technical indicators for price forecasting using denoised time series data in the B2C market produces promising results.

Keywords: Deep learning, feature extraction, time series, business to consumer markets, price forecasting

ÖZET

İşletmeden Tüketiciye Piyasalarında Derin Öğrenme ile Fiyat Tahmini

Emre EĞRİBOZ

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Mehmet S. AKTAŞ

Fiyat tahminleme, piyasalarda satışa tabi bir ürün veya hizmetin mali olarak belirlenen değerini yaklaşık olarak kestirmektir. Probleme bağlı olarak farklı zaman çözünürlüklerinde ele alınan fiyat tahminleme, kısa, orta ve uzun vadedeki gelecek zamanlar için yapılabilir. Ayrıca fiyat tahminleme, fazla sayıda araştırmacı ve işletme tarafından çeşitli piyasalar için akademik veya profesyonel çalışmalar yürütülen zorlayıcı, önemli ve temel bir problemdir.

Farklı piyasalarda fiyat tahmini için yapılan literatüre geçmiş çalışmalar çeşitli geleneksel yöntemler ve derin öğrenme yöntemleri kullanır. Bu çalışmalarda, araştırmacılar tahmin yapabilmek için yöntem fark etmeksizin tarihsel verilerinden faydalanarak tahminleyici modeller oluştururlar.

Çalışmamızda, tarihsel verileri kullanarak derin öğrenme temelli yöntemlerle fiyat indeksi tahminlemesi yapmak için veri toplama ve veri işleme iş akışlarını içeren bir metodoloji önermekteyiz. Önerdiğimiz metodolojiyi, veri toplama ve ileri analitik uygulamaları hazırlayarak gerçeklemekteyiz. İş akışlarımızı, işletmeden tüketiciye piyasalarının bir örneği olan havayolu yolcu taşımacılığı sektöründe işletmeler tarafından tüketicilere pazarlanan uçak bileti fiyatlarını kullanacak şekilde çalıştırmaktayız.

Veri toplama uygulamasında farklı havayolu işletmecisi firmaların bilet fiyatı verilerini günlük periyotta ağ kazıyıcı ile belirli bir süre topladık. Veri işleme uygulamasında toplanan zaman serisi verileri önişleme adımında tekilleştirdik, eksiklerini tamamladık ve fiyat indeksi oluşturduk; daha sonra matematiksel yöntemlerle teknik indikatörleri hesapladık; gürültüleri giderdik; çeşitli otokodlayıcılar ile derin öznitelikleri çıkardık; ardından bir derin öğrenme ağı ile fiyat tahminleme modelleri ürettik; sonunda, gürültü gidermenin, otokodlayıcı tiplerinin ve teknik indikatörlerin performansa olan etkilerini, üretilen modelleri test ederek ölçtük. Yaptığımız deneyler, B2C piyasasında gürültüsü giderilmiş zaman serisi verilerden otokodlayıcılarla üretilmiş derin özniteliklerin ve teknik indikatörlerin fiyat tahminleme için kullanılmasının umut verici sonuçlar ürettiğini gösterir.

Anahtar Kelimeler: Derin öğrenme, özellik çıkarma, zaman serisi, işletmeden tüketiciye piyasaları, fiyat tahminleme

1 INTRODUCTION

In the business-to-consumer (B2C) market, trade takes place between businesses and consumers. Businesses are product or service providers looking for serious competition in online platforms whose market share has been growing day by day in recent years, in addition to being able to find their customers with traditional methods. Electronic commerce, or e-commerce, is a trade model in which a product or service is sold to the consumer over the internet [1–3]. In this chapter, we explain the market share size of e-commerce, the need for price forecasting in B2C markets, motivation for the preparation of this study, hypotheses put forward within the scope of this thesis, and the general organization of this thesis' chapters.

1.1 Market Share Size of E-Commerce and Business-to-Consumer Markets

Opportunities that have become easier in recent years have expanded the use of e-commerce, increasing the trade volume in this shopping model and enabling it to reach various age groups and more consumers. Researchers estimate that the retail e-commerce sales volume, which was 4.28 trillion dollars in 2020, will reach 6.38 trillion dollars in 2024 [4]. In Turkey, the e-commerce data for the first six months of 2021 show that the volume in this area amounted to 161 billion Turkish Lira (TL) with an increase of 75.6% compared to the previous year [5]. PTTAVM is an e-commerce site, reported that the rate of users aged 65 and over increased from 6% to 10% during the COVID-19 pandemic [6]. E-Commerce Information Platform is a registration platform for e-commerce businesses in Turkey, shared statistics that the platform's user rate aged 55-64, which was 7.1% in the first half of 2020, increased to 10% for the whole of 2020 [7]. The global news agency 'We Are Social' reported that 70.5% of all age ranges varying from 16 to 64 in Turkey have purchased from the internet at least once in the last month in Digital2021 overview report shared in the first month of 2021 [8].

1.2 Reasons for the Need for Price Forecasting in B2C Markets

There are various types of scientific research on neuromarketing, effective advertising, customer satisfaction, and budget management to achieve and measure success in the field of e-commerce [9–11]. Another significant activity for success in the e-commerce domain is price forecasting studies [12]. Price forecasting studies in the B2C market mainly affect businesses and consumers. Businesses should make production or service investments within the limits of the supply-demand balance. In the case of more supply than demand, prices decrease due to many reasons, such as storage cost, shelf life, and low utilization of the service investment. Conversely, if a business does not create sufficient supply when there is an opportunity, the business is deprived of its potential profit [13]. In addition, the combination of the results of the price forecasting studies and the sales forecast data plays an important role in calculating the income to be obtained and indirectly in determining the strategic plans of the businesses [14–22]. Insufficient or no budget calculations negatively affect the finance, production, supply, and time management processes of businesses [14–22].

On the other hand, consumers want to spend with the highest profit by being minimally affected by price changes that may differ due to various reasons, such as periodically changing prices, cheap shopping demand, changes in exchange rates, or the effect of inflation [14–21].

1.3 Motivation

Forecasting prices in B2C markets and improving these forecastings have become a necessity for both businesses and consumers, as a requirement of online shopping services made widespread by ever-evolving information, payment and commerce technologies. Being able to forecast the future price of a product or service with this motivation; offers significant decision support outputs in terms of shopping at the right price, increasing the profitability of companies, stock, order, and investment management.

In addition, price forecasting studies are quite common in the stock market area, which is another strong market of commercial markets. In the studies conducted in the stock market domain, where many buyers and sellers come together, we investigated that price forecasting researches and experiments are successful and can be continuously improved. Price forecasting studies in stock markets often use historical data, calculated technical indicator features, and studies in computer science to strengthen forecasting performance. In the literature review, we observed that price forecasting studies in B2C markets are lack compared to stock markets. With this observation, examining the developmental aspects of forecasting studies for B2C markets is another source of motivation for this thesis. For this purpose, we conducted experiments within the scope of this thesis to forecast the price of a product or service in the B2C market and to improve the results by various methods.

We aim to make an acceptable price forecasting of any product or service in the B2C market to rationally contribute to our motivation. First, we collect price data of a B2C market for a certain period to be used in our price forecasting work. Then we organize it in the form of time series after the collection phase by processing this data set. Determining the technical and deep features to improve the outputs of the algorithm we use for price forecasting, supporting the price forecasting algorithm with these features, and examining the relationship between these features and the forecasting result are the outputs of this thesis.

1.4 Hypothesis

The hypotheses we put forward within the scope of this thesis in order to forecast the price of a product or service in B2C shopping markets and to improve the forecasts are as follows;

1- A deep learning model produced using historical data can be used to forecast the price of a product or service.

2- Features produced using historical data increase the performance of the model used in price forecasting.

3- Technical indicators increase the performance of the model used in price forecasting.

1.5 Organization of Thesis

The chapters of this thesis are arranged as follows. Chapter 2 provides introductory information and literature review results to understand the contents of different chapters. Chapter 3 introduces research questions. Chapter 4 provides detailed sections on the proposed solution method with all technical details. Chapter 5 specifies the software working details of the advanced analytical application prepared in the light of the proposed solution method details. Chapter 6 shares the details of the data set and configurations used in the evaluation of the proposed method. Chapter 7 presents results from the study and possible future work. Chapter A presents extended experiment results.

2
GENERAL INFORMATION AND LITERATURE REVIEW

We introduce the fundamentals of analytical data processing software we have prepared to answer our hypotheses and research questions; we present the literature review and contribution of the thesis in this chapter.

First, we explain general information and terminology regarding the field expertise of analytical data processing within the scope of our study. The information in this chapter is an introduction to understanding the detailed topics in Chapter 4 Proposed Methodology.

We present a summary of the solution methods followed in similar studies in the literature in the field of forecasting in this chapter. We understand and briefly explain under sections how the other studies in the literature handle the main steps and sub-steps of the problem studied within the scope of this thesis, which methods they prefer, and the functioning of these methods in the literature review study.

2.1 General Information

Advanced data analytics applications aim to make sense of data by analyzing it. Decision support systems, anomaly detection systems, forecasting and suggestion mechanisms, and many other fields frequently use advanced data analytics applications thanks to their ability to work with streaming data in real-time or near real-time, and with historical data, with developing technologies and infrastructure opportunities. If there is a before-after relationship between the data to be analyzed and if the data are produced over time, these data are suitable for processing using time series processing methods [23–33].

Advanced analytics applications often involve more than one step, and each step's input is the output of the previous one. An advanced analytics application usually processes the data along a processing pipeline similar to the steps below to produce results as the output of each step [34].

- Data Acquisition
- Data Preprocessing
- Feature Engineering
- Sampling The Data Set for Training and Testing
- Model Training
- Test and Optimization

2.1.1 Data Acquisition

Data is indispensable for performing experiments in analytical studies. There are different methods to gather the data to be studied. These methods vary according to the content and purpose of the experiments in the study. Observation, interview, and survey are some of them. The process is generally called data acquisition or data collection.

While collecting data, data collectors pay attention to issues such as collection systematic, relevance for purpose, frequency, correctness, or completeness, since the main purpose of data collection is to benefit from it. Data collection is a costly process in terms of both time and money. Collection of the data incorrectly or incompletely will produce dirty data. Dirty data threaten the accuracy of the results of the analytical system. In addition, since cleaning dirty data will require an effort, data collectors should not collect incorrect or incomplete data during the data acquisition step.

2.1.2 Preprocess

Algorithms do not often directly process the raw data for many structural or logical reasons. Some of these reasons are: some samples or dimensions in the data set may contain noise; the ranges of the dimensions of the data may differ from each other; there may be some duplication or missing from the data acquisition step in samples and dimensions; some samples may contain outlier values due to input or measurement errors. Generally, the preprocessing step clears the noise of the data set, standardizes the ranges between dimensions, deduplicates the repetitive data, imputes missing data within a certain logic, and regulates outlier values.

Retrieving data from multiple sources during the data acquisition step avoids the risk of dirty data but may results in more than one data representing the same sample. Although representation of the same sample by more than one data is not a problem in all cases, time series data processing needs a convenient data set. Creating a basket for samples from different sources and representing the data cumulatively with an index value is a preferred solution method. The highest 100 stocks traded in Borsa Istanbul with an index (BIST100) or the inflation basket calculation are similar examples of this method. Note that this is different from the repetitive data problem because repetitive data is the accidental acquisition of the same data from the same source more than once.

2.1.3 Feature Engineering

Feature engineering is the general name given to the processing studies carried out to get more benefit from the data. Feature engineering at any step in the analytical application also provides forecasting performance improvement possibility, as it often leads to a better understanding of the data in later steps. Feature extraction and selection are two decisive steps applied to the dimensions of the data within the scope of feature engineering. Feature extraction is the study of extracting more meaningful information from some samples or dimensions in the data set that is less significant when singular. Feature selection refers to eliminating redundancies that will not be useful to the analytical application, or in other words continuing the workflow by selecting advantageous dimensions in the data set. Both analytical computation methods and deep learning methods can perform feature extraction and selection.

The feature engineering step prepares the analytical and deep features thanks to the external feature extractor step(s) to include them in the forecasting model training. Analytical features are usually mathematical values prepared from supplementary data sources or field expertise. Technical indicators are the features calculated by analytical methods using samples in numerical time series data. Deep features, on the other hand, are the result values generated by the models trained with feature extractor deep learning networks.

The ability of deep learning methods to do feature selection and extraction by themselves is one of the main differences between classical machine learning and deep learning methods. Deep learning methods can select features through the layer weights that can be changed by the method applied during training in the model, and extract features using intermediate layers. Although deep learning training extract features, the methods explained in this section also extract features as a preliminary step.

2.1.4 Sampling The Data Set for Training and Testing

Analytical modeling applications use some of the available data for model training while reserving the rest for testing the model. There are a variety of ways to create train and test sets, such as splitting all data into portions, selecting random samples, or selecting samples guided by a distribution. Exploring the data set with a problem focus is proper to choose the separation method of the data set to create training and test sets. Solution of classification problems frequently uses a balanced sampling method from each class; solution of regression problems frequently uses the sampling method of a certain percentage of the data.

2.1.5 Machine Learning, Deep Learning and Processing Data in Time Series Form

Machine learning is a field that works on creating models based on historical data and evaluating new data with the trained models. In machine learning, the learning type is named according to the problem working on and the situation of the variables in the data set. A learning algorithm can be supervised or unsupervised. In supervised learning, the dependent variables are known within the scope of the studied data set; in unsupervised learning, the dependent variables are unknown.

Deep learning is a specialized field of machine learning that does the feature engineering of data using layered neural networks. Just like machine learning, deep learning serves supervised and unsupervised learning methods. Solutions of regression and classification which are supervised learning problems, and clustering and dimension reduction which are unsupervised learning problems use deep learning techniques frequently. One of the fundamental differences between deep learning and machine learning algorithms is that deep learning can do feature engineering without a field expert. In addition, deep learning also covers many topics, such as the connection, operation, and improvement of neural networks.

Data sets can be of many types depending on their content. The data, which are digitized by sampling from a data source and have a before-after relationship according to the timeline, are processed in the time series data set concept. Different signal processing methods have been developed to analyze data sets in this form. These methods usually process the data used after slicing it into windows that fit a certain number of points.

2.1.5.1 Artificial Neural Networks and Recurrent Neural Networks

The perceptron model is the basic building block for simulating the abilities of the brain's biological neurons in the digital environment. The perceptron is the lowest granularity unit element of artificial neural networks and produces an output using the inputs provided to it. The inputs, evaluation weights, and output behavior of this unit can be customized depending on the context of the application method. Combining multiple perceptrons in stacks creates perceptron layers, and combining these layers creates artificial neural networks. The outputs produced by the layers are used as inputs in different layers in an artificial neural network. Feedforward neural networks consist of layers that their outputs constantly connect as an input to the layers that follow them in the structure of the neural networks. Recurrent neural networks (RNN) consist of layers that connect to both the layers that follow them and themselves.

2.1.6 Test and Optimization

After the model training, the model makes predictions with a set of test data that is not used in training to determine the real-world performance of the model produced. The distance between the predictions made by the model and the actual results is expressed in numerical measurements using different metrics and shows the error (in other words, the performance) of the model.

The processes operating at each step of advanced analytical applications may evolve to become more effective to improve performance or speed up the operation. Extracting the features in a way that pays attention to the model complexity during the feature extraction stage, selecting features to make the best use of the data, choosing the most beneficial features for model performance, and running the workflow with these selections are important optimization issues of the feature engineering step.

Determining the window size taking into consideration the learning model result during the windowing of the time series data set, and choosing the fine-tuning hyperparameters such as layers, activation functions and learning rate used in the established artificial neural network according to the problem are the optimization points in the analytics of time series data sets.

2.2 Literature Review

In the literature, there are forecasting studies dealing with numerical time series data. The number of researches in this field directly or indirectly related to the economy and finance industry is very high. The models proposed in these forecasting studies, in which numerical data such as price, sales volume, and usage rate are the main actors, were created with traditional machine learning methods or deep learning algorithms.

The prominent issues in the working processes of advanced analytical applications that process numerical time series data regardless of the method are noise removal in the preprocessing step, feature extraction in the feature engineering step, and creation and optimization of the forecasting model. All steps are open to improvement in terms of the methods chosen or applied, due to the depth of the study area and the diversity of approaches.

In this section, the methods in the steps of noise removal, feature extraction, and creation of forecasting models in different studies generally containing time series data sets in the literature are scanned from general to specific.

2.2.1 Noise Canceling

Noise-canceling or noise removal is an important sub-step that is applied to the samples during the preprocessing step. Noises in the data set are tolerated by cleaning or approximating the samples to their real values. Noise is removed in the time or frequency domains. Studies in the literature use the simple or weighted average over sliding windows (namely moving average), median filter, discrete wavelet transform with signal decomposition, and various methods in different complexity to de-noise the time series data set. The moving average method is linear; the median filter method is a nonlinear noise removal method.

2.2.1.1 Moving Average

The simple moving average is the simplest, most common, and easily applicable noise removal method known. By constantly shifting a fixed size window in the same direction on the data set, averaging is performed for all individual points. The arithmetic average of the points around it determines the new value of each element as well as its window size. This operation is called convolution, and the role played by this window is called a simple mean filter [35]. It is also possible to perform the convolution operation with different filters. For example, the filter used in the weighted moving average method gives high weight to the elements close to the point itself while giving lower weight to the elements farther away. In this way, the technique ensures that close elements have more influence on the average result. The moving average method is also a linear noise removal method since the filter consists of linear variables. Equation 2.1 formulates the simple and weighted moving average. x_i' is the new value of i.th element; WS is the window size; x_j is the value of the j.th element;

Δ represents the function that determines the weight based on the distance between i and j in the formula. In the simple moving average method, the function Δ is 1 for each i and j since all neighboring elements have the same weight [35, 36].

$$
\begin{gathered}
x_i' = \frac{1}{WS} \sum_{j=i-WS/2}^{i+WS/2} \Delta(i,j) x_j \\
\sum_{j=i-WS/2}^{i+WS/2} \Delta(i,j) = WS
\end{gathered}
\tag{2.1}
$$

Jiang et al. performed noise-cleaning from plane engine vibration signals by using moving average and combined methods based on it [37]. Hu et al. performed baseline wander removal from electrocardiography signals in their proposed moving average-based method [38].

2.2.1.2 Median Filter

The median filter is a filter that removes noise as a result of representing a point with the median value around a certain window size. A fixed-size window is constantly shifted in the same direction over the data set, and a median filter is applied for the relevant point at each scroll to apply the filter to the entire data set. The median of the points around it determines the new value of each element as well as its window size [39]. Equation 2.2 formulates the median filter. x_i' is the new value of i.th element; WS is the window size; x_i is the value of the i.th element; $median$ refers to the function that finds the median in the formula [39].

$$
\begin{aligned}
x_i' = median([x_{i-WS/2}, \\
x_{i+1-WS/2}, \\
..., \\
x_i, \\
..., \\
x_{i-1+WS/2}, \\
x_{i+WS/2}])
\end{aligned}
\tag{2.2}
$$

In their study, Liu et al. showed that their proposed median filter-based methods for removing noise in seismic data effectively reduce noise [40].

2.2.1.3 Discrete Wavelet Transform

The discrete wavelet transform (DWT) method transforms a discrete time series signal into meaningful signal waves, each having separate granularity. These separate signal waves create the discrete time series signal itself [41]. This method is frequently used as a noise reduction function in signal processing works. A.J. Conejo et al. used the discrete wavelet transform method as a noise reducer in their electricity price estimation studies [42].

Section 4.2.4 Discrete Wavelet Transform explains the technical details of the DWT method.

2.2.2 Feature Extraction

Extracting features enriches the raw data since the amount of information of the raw data is often limited [43]. The studies in the literature frequently used traditional computational methods to extract analytical features and deep learning methods to extract deep features in time series data sets.

Analyzers of the financial time series data sets commonly use technical indicators to see the price change and to create projections for forecasting. Technical indicators are calculated from historical price and sales volume data [44]. K. Prachyachuwong and P. Vateekul, in their studies to forecast the trend of the stock market price, showed that usage of technical indicators in addition to news-based textual attributes increases forecasting performance [45]. Section 4.3 explains the technical indicators used in our study in detail.

Studies in the literature have used various approaches, such as convolutional neural networks (CNN) and autoencoders to extract deep features from time series data sets using deep learning. CNN is a supervised deep learning approach in which layers that perform convolution operations on inputs with various filters are brought together and operated. T. Kim and S. Cho obtained the best results with the CNN feature extractor for all time resolutions they considered in their energy consumption estimation studies in residential areas [46].

Autoencoder is an unsupervised deep learning approach that learns the representation of data belonging to a field of expertise thanks to artificial neural networks [47]. W. Bao et al., in their study to forecast stock market prices, showed that the forecasting performances increased with the stacked autoencoder [48]. L. Wang et al. used stacked autoencoder derivatives in their short-term electricity price forecasting study [49]. Section 4.5 Autoencoder explains the operation and types of autoencoders.

2.2.3 Forecasting Model

Researchers propose different models based on machine learning to forecast data in time series form in the literature. Studies dealing with regression-based problems use traditional machine learning methods, such as linear regression, moving average, and autoregression, and deep learning methods, such as long short-term memory (LSTM).

2.2.3.1 Linear Regression

Linear regression is a very simple regression method that can be used in problems where there is a linear relationship between input variables and output [43]. Establishing the line equation that will produce the output with the lowest possible error in response to all input variables in the data set creates the forecasting model [50, 51]. Equation 2.3 formulates the line equation for simple linear regression [50, 51]. $\hat{y}$ represents the forecasting result; x is the input variable; β_0 is the intersection point with the y-axis; β_1 represents the slope of the line in the equation [50, 51].

$$\hat{y} = \beta_0 + \beta_1 x \tag{2.3}$$

2.2.3.2 Moving Average

The moving average method tries to predict one step ahead from the observation obtained from the average of previous outputs [43]. The simple or weighted average calculates the forecasting result of the values within the window size. Equation 2.4 formulates the regression by simple and weighted moving average. The $\hat{y}_i$ is the i.th forecasting value; WS is the window size; x_j is the value of the j.th element; Δ represents the function that determines the weight based on the distance between i and j in the formula. In the simple moving average method, Δ is 1 for each i and j since the previous elements in the window have the same weight [35, 36].

$$\begin{gathered} \hat{y}_i = \frac{1}{WS} \sum_{j=i-WS}^{i} \Delta(i,j) x_j \\ \sum_{j=i-WS}^{i} \Delta(i,j) = WS \end{gathered} \tag{2.4}$$

2.2.3.3 Autoregression

Autoregression tries to make an estimation based on the values of the time series in the previous period. Calculation over the values in the backward-related periods produces the forecasting result. The number of periods taken into consideration is expressed in order [43]. Equation 2.5 formulates the autoregression. In the formula, the $\hat{y}_i$ is the i.th forecasting value; Z_t is the value of the random noise function; r is the order; α_j is the function that determines the model weight for the j.th element; x_{i-j} represents value of the $i-j$.th element [50, 52, 53].

$$\hat{y}_i = Z_t + \sum_{j=1}^{r} \alpha_j x_{i-j} \tag{2.5}$$

2.2.3.4 Long Short-Term Memory

Long short-term memory (LSTM) is a special deep learning architecture that includes augmented recurrent neural networks (RNN) with various technical enlargements. J. Cao et al. obtained successful forecasting results by using LSTM in their financial time series estimation studies [54]. S. Bouktif et al. obtained successful estimation forecasting by using LSTM in their electricity consumption estimation studies [55]. Section 4.4.4 RNN and LSTM Layers explains how LSTM networks work, and Section 4.7 Forecasting Model explains usage of LSTM networks within the scope of our study.

2.3 Contribution of the Thesis

In this research, we aim to make a price index forecasting with a deep learning model created using historical time series price data in B2C markets. We develop different and effective uses of preprocessing and feature extraction steps within the scope of advanced analytics application, where we propose a data collection and processing architecture. We analyze the effects of noise removal in the preprocessing step, the usage of deep features extracted by autoencoders, and the usage of technical indicators on forecasting performance, through experiments performed in the proposed application. The outputs of these analysis results, showing the effects of the methods, recipes, and materials used, contribute to improving the performance of a product's price forecasting model in the B2C market.

3
DEFINITION OF THE RESEARCH PROBLEM

As a result of and need for increasing transaction capacities, developed technologies, and trade volumes in all markets, price forecasting studies have become academically and professionally valuable. Price forecasting is needed in B2C markets, as in other markets, with the increase in payment opportunities and the spread of e-commerce in recent years. The literature review shows that studies in this area are lacking for the B2C market.

In price forecasting studies for other markets, preprocessing, feature extraction, and predictive model building methods are among the prominent topics. These methods are always open to improvement. In our study, we investigate the necessary approach, place, and importance of these prominent steps to make price forecasting in the B2C market.

3.1 Research Questions

In this section, we present the research questions related to the hypotheses put forward within the scope of the thesis and introduce their contents.

3.1.1 Could Time Series Price Data and Deep Learning Be Used to Forecast Prices in B2C Markets?

The literature review shows that deep learning methods produce successful outputs in time series price data. Although the data sets used in previous studies applying deep learning methods are in time series format, studies representing the B2C market are very limited. The first step of the research is to analyze the suitability of deep learning for price forecasting in the B2C market within the scope of this thesis.

3.1.2 Which Preprocessing Methods Need to Be Used in Time Series Price Data?

In the preparation phase, there are many preprocessing methods defined on the data. Some of these preprocessing methods work with samples in the data, and some work with dimensions. Which preprocessing methods will be used depends on the problem the study is trying to solve and the character of the data. In this part of our research, we discuss the data preprocessing methods that should be applied to make time series data ready for processing.

3.1.3 Which Deep Learning Methods Can Be Used to Extract Features from Price Data?

Deep learning methods can be used as good feature extractors with their artificial neural networks, as we mention in Section 2.1.3 Feature Engineering. We know from previous works in literature that a deep learning algorithm cannot be the best feature engineer for all types of data. In this section, we discuss different deep learning algorithms in the role of feature engineer to investigate which deep learning methods produce features that will yield good results in forecasting when used with time series financial price data.

3.1.4 Are Technical Indicators to Be Derived from Price Data Useful for Price Forecasting in the B2C Market?

While creating the price forecasting model in the B2C market, the effect of using technical indicators used in stock market studies, apart from deep attributes, on model performance can be demonstrated with several experiments to be carried out. The findings as a result of the experiments will produce outputs about the usability and benefits of technical indicators in price forecasting studies in B2C markets. In this part of our research, we also use technical indicators to train price forecasting models then we test forecasting models.

4
PROPOSED METHODOLOGY

We describe the general processes of analytical applications for data in Chapter 2. Analysis of the processes indicates that the preparation of the advanced analytics application needs a structured historical data set and a well-designed data processing pipeline to forecast the future price of a product in B2C markets. Figure 4.1 illustrates the customized workflow outline we propose based on these processes to forecast price index with deep learning using time series data set in B2C markets. In this section, we explain the technical details of each process in the workflow.

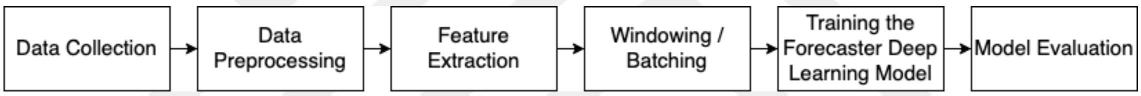

Figure 4.1 The proposed methodology outline

4.1 Data Acquisition

Collecting data from web pages is in the class of data acquisition by observation method, and web scrapers automate this work. A web scraper application makes it possible to collect structured data contained in web pages. Automatic web scrapers first make predefined searches on the target website and then collect useful data from the search results in a structured way.

Technically, a web scraper visits web pages using its browser driver, parses the page elements, navigates on the page, and collects data from the page elements. The scraper often uses Document Object Model (DOM) elements and the hierarchical layout of page elements to do parsing on the page.

4.1.1 Automatization of Browser Usage

A browser driver is software required for computer control of a web browser. Drivers automate browser usage by applying computer-executed commands on the web page.

Companies with browser software often provide their drivers for developers. Browser drivers are also often used for test automation aside from web scraping.

4.1.2 Document Object Model and XML Path Language

Web pages consist of page elements that are responsible for content such as text, images, links, and buttons; and consist of page elements responsible for layouts, such as rows, columns, tables, and separators. DOM shows the hierarchical arrangement of page elements and provides access to the elements. DOM access is used in Hypertext Markup Language (HTML) and Extensible Markup Language (XML) documents. XML Path Language (XPath) defines the placement location of an element on a page in the page tree, and browser drivers access DOM elements using XPath location.

4.1.3 Visiting a Page

The web scraper program configures the documented options of the browser driver before visiting the page. Then the program starts by visiting the target website page to be scraped.

4.1.4 Parsing and Using Page Elements

The scraper program finds the searched page element on the web page by extracting it from the XPath location with DOM element access. The parsed element can point to a search box, a data holder that contains the target data to be scraped, a navigation button, or any required element on the page.

The scraper program uses the browser driver to parse and then interact with the page elements. The browser driver performs simple and complex interactions on the page, such as typing in text boxes, clicking buttons, scrolling, navigating, or scripting.

4.1.5 Extracting Elements

The web scraper program extracts the contents of the elements found and parsed on the web page with the element selectors. The parsed element on the page can contain the target data or a list of elements that contain the target data. The program directly processes elements that contain the target data to be scraped themselves; processes elements in list form after accessing the target data from within the list. After extracting, the program serializes the data and saves it in the appropriate format.

4.1.6 Sleeping

On the visited pages, waiting for a short time is necessary to load the entire or part of the page due to network traffic delay, server density, or slow access speed while the scraper program running the commands. Not waiting enough time can result in missing some or all of the elements to be extracted from the page. Although a fixed sleep time is not defined, in practice, the web scraper application waits for a random duration that fit a uniform or normal distribution in the processes where the page sends requests to the server.

4.2 Preprocess

Preprocessing is the bunch of processing steps in which prepare the collected data is ready for analysis in the pipeline. Preprocessing tidies up the data set in the steps as following: It imputes the missing parts, applies the necessary transformations, and cleans the noise to prepare the data set for the remaining analytical processing steps.

4.2.1 Deduplication and Singularization

Exactly the same records in the data set are called repeat records. Repeat records are identical in all their features. The presence of repeated samples in the data set may create an imbalance; it can delay the analysis work as it causes an increase in the amount of data or can create a synthetic bias in the analysis model. For these reasons, repeat records are deduplicated during preprocessing.

In addition, in some cases, the records in the data set are used after they are grouped to create semantic integrity. There is no commonly known name given to these grouped records, but in our study, we can name these records as multiple records. Multiple records are similar to each other in at least one dimension and are singularized after they are grouped and aggregated. The data set collected may contain repeat or multiple records. This is not an error, but the repeat and multiple records in the data set should be handled.

4.2.2 Imputation

In contrast to the repeated samples, there is no data at all for some points on the timeline. If this situation, which causes a lack of data, is not resolved, it affects the performance of the analysis work to be done, since the time series data set is corrupted at random points.

A pre-specified procedure is followed to impute missing data of the entire data set.

One of the methods used when completing the missing parts of digital time series data is to fill in the missing data with the arithmetic average of before and after a specified period. The seasonality characteristic of the data sets having seasonality is preserved with this method.

4.2.3 Indexing

The price of the same good or service offered for sale in the market may vary due to many reasons, such as free price policy, time-dependent dynamics, or the quality of the good or service. Creating a consumption basket minimizes the effects of these changes on the analyzes to be made in statistical calculations. Thereby price indexes of goods or services are created at the working time resolution.

4.2.4 Discrete Wavelet Transform

The discrete wavelet transform removes noise by smoothing the extreme values contained in the time series data. The method proposed by S. Mallat decomposes a discrete time-series signal at a specific multilevel resolution [56]. The method exposes the discrete signals of the main signal with the low-pass and high-pass filters. The low-pass filter outputs the low-frequency discrete signal (ϕ); the high-pass filter outputs the high-frequency discrete signal (ψ). The low-frequency discrete signal refers to the father wavelet or approximate coefficients; the high-frequency discrete signal refers to the mother wavelet or the detailed coefficients. Equation 4.1 and 4.2 formulates discrete J-level low and high frequency wavelets [57].

$$\phi_{j,k} = 2^{-j/2}\phi(\frac{t-2^j k}{2^j}), \quad \psi_{j,k} = 2^{-j/2}\psi(\frac{t-2^j k}{2^j}) \tag{4.1}$$

$$\int \phi(t)dt = 1, \quad \int \psi(t)dt = 0 \tag{4.2}$$

Equation 4.3 formulates the projection of low and high frequency wavelets with a time dependent function $g(t)$.

$$\begin{gathered} a_{J,k} = \int \phi_{J,k}(t)g(t)dt, \quad d_{j,k} = \int \psi_{j,k}(t)g(t)dt \\ j = \{1,2,...,J\}, \quad a = 2^j, \quad k = \{1,2,...\} \end{gathered} \tag{4.3}$$

Equation 4.4 formulates the sequence $\{S_j, D_j, D_{j-1}, ..., D_1\}$, which is the sum of the approximate coefficients S_j and the detailed coefficients D_j as a function $g(t)$.

$$g(t) = \sum_k a_{J,k}\phi_{J,k}(t) + \sum_k d_{J,k}\psi_{J,k}(t) + \sum_k d_{J-1,k}\psi_{J-1,k}(t) + ... + \sum_k d_{1,k}\psi_{1,k}(t) \quad (4.4)$$

4.2.5 Data Scaling

Data scaling transforms the numeric features in different ranges in the data set into a common range [58]. Equation 4.5 formulates to scale any feature F in the data set to a new minimum-maximum range. x_i is the i.th value; x_i' is the scaled new i.th value; min_F is the minimum value of feature F; max_F is the maximum value of feature F; new_min_F is the new minimum value of feature F; new_max_F represents the new maximum value of feature F in the formula [58].

$$x_i' = \frac{x_i - min_F}{max_F - min_F}(new\_max_F - new\_min_F) + new\_min_F \quad (4.5)$$

4.3 Technical Indicators

Technical indicators are numerical signals used in the price analysis. Using the technical indicators in the technical analysis makes understanding the change and trend in the price and creating inferences and projections for the future for forecasting studies possible [44]. Technical indicators are calculated by using one or more of the open, close, high, or low values of the price in a certain period. The terms open, close, high, and low values refer to the first, last, highest and lowest market sales value of the analyzed price in a certain period, respectively [44].

Technical indicators are important and efficient signals that are often used in financial decision support systems. There are various indicators used for different purposes. Table 4.1 shows the abbreviations and meanings of some of these technical indicators.

4.3.1 Moving Average

The moving average (MA) indicator shows the price trend in the ascending, descending, or stable direction for a given period [59]. The equation 4.6 formulates the MA indicator at t.th day [59]. T is the period in the number of days resolution; P stands for price in the formula.

Table 4.1 Abbreviations and descriptions some of technical indicators

Indicator Name	Description
MA	Moving Average
EMA	Exponential Moving Average
TRIX	Triple Exponential Moving Average
MACD	Moving Average Convergence/Divergence
ROC	Rate of Change

$$MA_t = \frac{1}{T}\sum_{i=t-T}^{t} P_i \tag{4.6}$$

4.3.2 Exponential Moving Average

The exponential moving average (EMA) indicator shows the price trend in the ascending, descending, or stable direction for a given period. Equation 4.7 formulates EMA indicator at t.th day [59, 60]. T is the period in the number of days resolution; P stands for price in the formula.

$$\begin{gathered} c = 2/T + 1, \\ EMA_t = cP_t + (1-c)EMA_{t-1} \end{gathered} \tag{4.7}$$

4.3.3 Triple Exponential Moving Average

The triple exponential moving average (TRIX) indicator is obtained by calculating the EMA indicator three times in a row. TRIX shows the extreme demand or supply of the product in the market for a certain period [61, 62]. Equation 4.8 formulates TRIX indicator at t.th day [59]. P stands for the series of close prices in the formula. Where $EMA1$, $EMA2$ and $EMA3$ are series.

$$\begin{gathered} EMA1 = EMA(P) \\ EMA2 = EMA(EMA1) \\ EMA3 = EMA(EMA2) \\ TRIX_t = (EMA3_t - EMA3_{t-1})/EMA3_{t-1} \end{gathered} \tag{4.8}$$

4.3.4 Moving Average Convergence/Divergence

The moving average convergence/divergence (MACD) indicator shows the price trend from the relationship between the exponential moving average in the increasing, decreasing, or stable direction. MACD is obtained by subtracting the long-term exponential moving average from the short-term exponential moving average [59]. The equation 4.9 formulates the MACD indicator at t.th day [59]. $T1$ is the short-term period in the number of days resolution; $T2$ refers to the long-term period in the number of days resolution in the formula.

$$MACD_t = EMA_{(T1)t} - EMA_{(T2)t} \tag{4.9}$$

4.3.5 Rate of Change

The rate of change (ROC) indicator shows proportionally how much the price has changed for a given period compared to a previous price period [59]. The equation 4.10 formulates the ROC indicator at t.th day [59]. T is the period in the number of days resolution; P stands for price in the formula.

$$ROC_t = \frac{P_t}{P_{t-T}} * 100 \tag{4.10}$$

4.4 Deep Learning

Deep learning, which is a specialized study area of machine learning, creates models by extracting deep features from historical data with layered artificial neural networks and allows new data to be evaluated with these models [47]. Deep learning provides solutions for non-linear computer-based learning problems with models produced by various algorithms. Figure 4.2 presents the taxonomy of problems commonly studied in deep learning [47].

In deep learning terminology, training is the learning phase; testing is the evaluation phase; layer is the data structure that can receive and process inputs and provide output; model is the neural networks established with layers [63]. Except for the input and output layers in the deep network architecture, each of the other layers used is called the hidden layer [47].

The independent variables refer to the features in the data set; the dependent variable

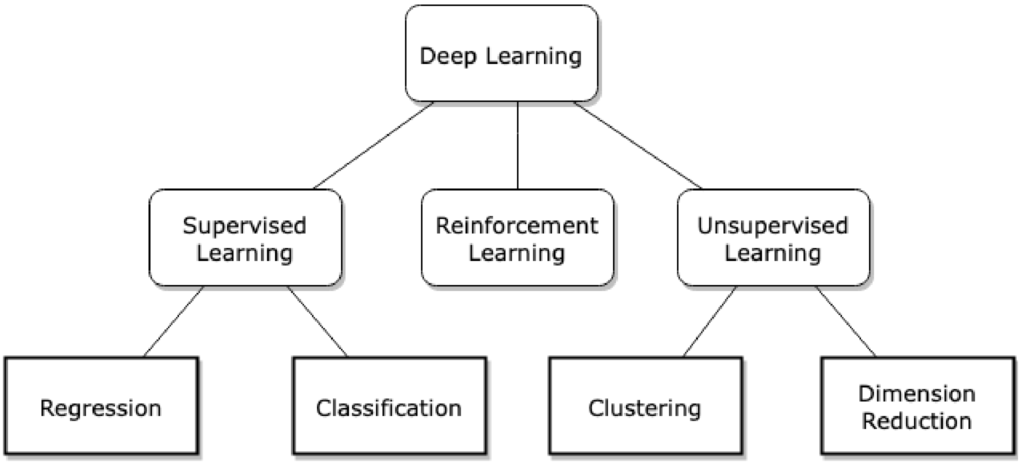

Figure 4.2 Deep learning taxonomy

refers to the output. In supervised learning problems, the dependent variable in the data set is known during the training, and the dependent variable supervises the training algorithm. The model makes predictions to find the dependent variable using the independent variables of the new data after the training. Forecasting problems whose dependent variable is determined over numerical values are regression; prediction problems whose dependent variable is determined over categorical values are called classification in supervised learning [47].

In unsupervised learning problems, the dependent variables in the data set are not known, and after the training, the model makes predictions for the new data in line with what it learned from the training data. Since the dependent variables are not known in unsupervised learning, algorithms are developed for dimensionality reduction or clustering problems that can be done on independent variables [47]. The types of problems studied in deep learning and the relationship between problems and data are not limited to those described in this section.

Combining the different layers creates deep learning architectures. Layers are built specifically according to the purpose of use and the processing of the input data. We explain the architectures and algorithms of the input and output layers, the perceptron, and the fully connected and recurrent layers in this section.

4.4.1 Input Layer

The input layer is the entry point where each sample of the data set joins the neural network. The input dimensions of this layer and the shape of the dimensions are the same as the size and shape of the samples in the data set since no processing has been done to the data yet.

4.4.2 The Perceptron

The perceptron, which is the smallest unit element used in artificial neural networks, models the brain neurons mathematically on the computer [47, 63]. Figure 4.3 illustrates the architecture of a perceptron. Equation 4.11 formulates the perceptron's result value y. x_i, is i.th input value of sample; W_i, represents the weight of the i.th input value held in the perceptron in the equation. The activation function f limits the output to a certain range can be special functions, such as tanh, sigmoid or ReLu [47, 63].

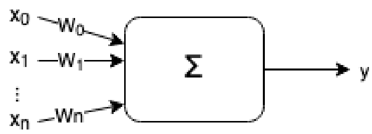

Figure 4.3 Architecture of the perceptron

$$y = f(\sum_{i=0}^{n} W_i x_i) \tag{4.11}$$

The output calculated with well-adjusted weight values achieves an accurate result. The weights are initially set under a weight initialization policy and then updated during the training phase by feedforward and backpropagation methods. Feedforwarding is the calculation of the result in the perceptron for each sample in the data set. Backpropagation reflects the error as an update to the weights. Error function calculates the error between the feedforwarding result and the expected result [47, 63].

4.4.3 Fully Connected Layer

Stacking the perceptrons creates the fully connected layer or also known as the densely connected layer. Fully connected layers powered by multiple perceptrons enable faster and more accurate decision-making for nonlinear problems [47, 63]. Figure 4.4 illustrates an example of a neural network using fully connected layers.

4.4.4 RNN and LSTM Layers

The recurrent layers used in RNN contain memory cells to hold the state of the previous iteration unlike the layers used in feedforward neural networks. These networks are called recurrent neural networks since the state held in the cell is given in a way that creates a loop in the network alongside the input data. Figure 4.5 illustrates the architecture of an RNN unit. x_t is the input vector at time t; s_t and s_{t-1} are state

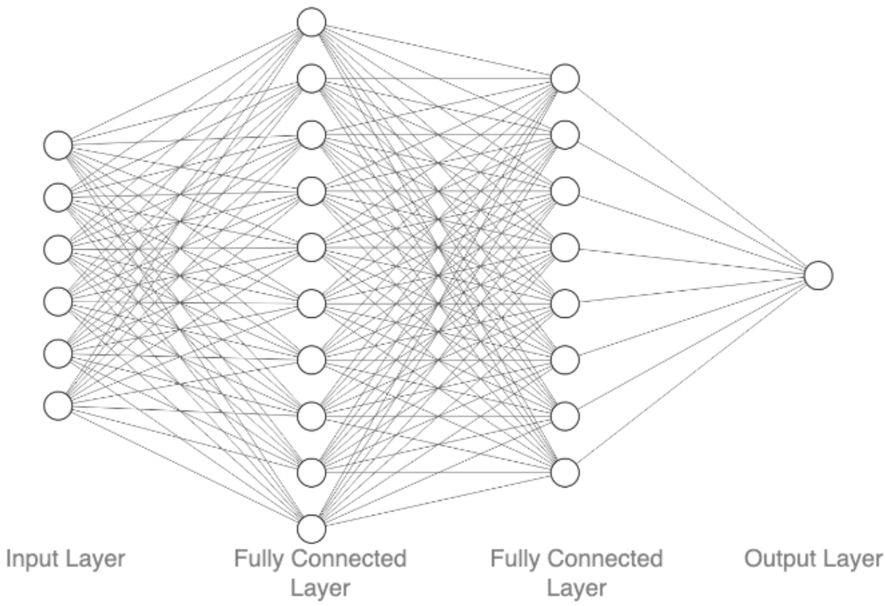

Figure 4.4 An artificial neural network consisting of input, fully connected, and output layers

vectors calculated from the input vector at time t and $t-1$; W and U represent the weights of the input and state vector in the figure. Equation 4.12 formulates the result value s_t. The activation function f can be special functions such as tanh, sigmoid or ReLu [63].

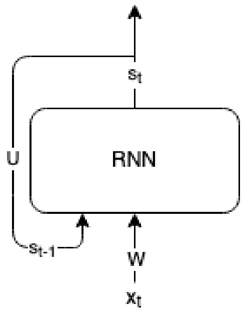

Figure 4.5 Architecture of the RNN unit

$$s_t = f(Wx_t + Us_{t-1}) \tag{4.12}$$

LSTM is a special type of recurrent neural network that has solved the disappearing gradient problem during computations in RNN. Besides the RNN unit, an LSTM unit includes the input gate that controls updating the cell state, the output gate that controls the value to be sent to the next cell, and the forget gate that controls the previous state of the unit [47, 63]. Figure 4.6 illustrates the architecture of an LSTM unit. The LSTM layer is created by setting the output size shape of the LSTM unit and parameters such as the activation function.

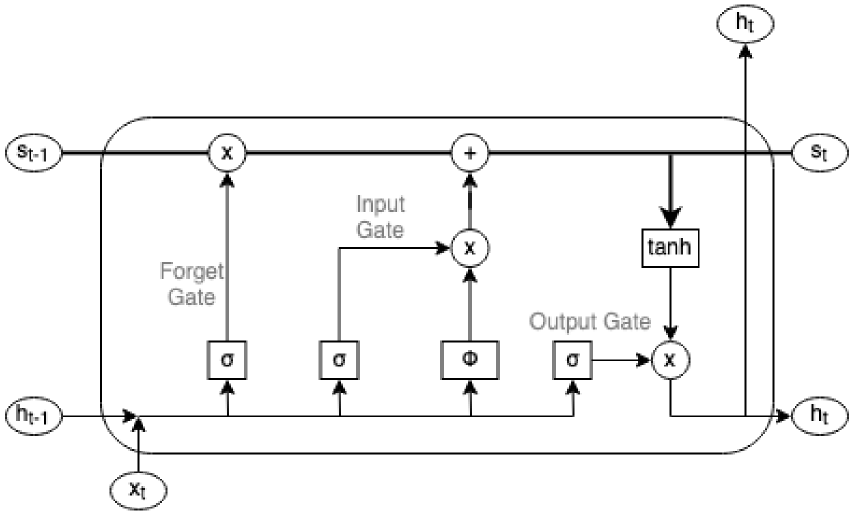

Figure 4.6 Architecture of the LSTM unit

Equation 4.13 formulates the expressions f_t, g_t, q_t specifying 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 vectors W and V represent the weight vectors for the input and gates. The activation function σ is the sigmoid function that outputs the input provided to it between 0 and 1 [47, 63].

$$\begin{aligned} f_t &= \sigma(W_f x_t + V_f h_{t-1}), \\ g_t &= \sigma(W_g x_t + V_g h_{t-1}), \\ q_t &= \sigma(W_o x_t + V_o h_{t-1}) \end{aligned} \tag{4.13}$$

Equation 4.14 formulates the state value of the LSTM unit, s_t, and the output vector, h_t, at time t. The vectors W and V are the weight vectors for the input and LSTM unit; Φ represents the input activation function, usually sigmoid or tanh in the equation [47, 63].

$$\begin{aligned} s_t &= f_t s_{t-1} + g_t \Phi(W x_t + V h_{t-1}), \\ h_t &= tanh(s_t) q_t \end{aligned} \tag{4.14}$$

4.4.5 Output Layer

The output layer is the last layer in the neural network. After the hidden layers process the samples in the data set throughout the network, the output layer produces the prediction result.

4.5 Autoencoder

Autoencoder is an unsupervised deep learning method used in feature extraction or noise-cleaning by learning domain data. The autoencoders for extracting deep features in the data set essentially consist of an encoder layer and a decoder layer [47, 48]. Figure 4.7 illustrates an autoencoder consists of a fully connected encoder layer at the input and a fully connected decoder layer at the output. We refer to this type of autoencoder as a single autoencoder throughout our study.

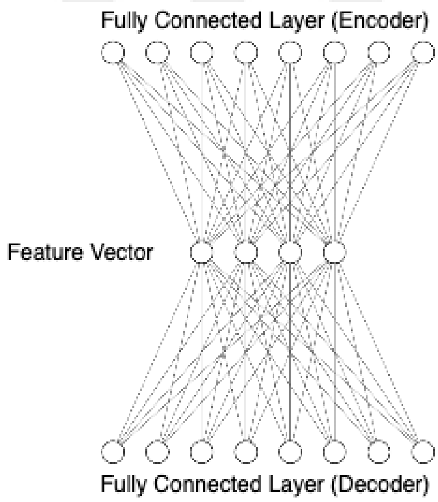

Figure 4.7 A single autoencoder consisting of fully connected encoder and decoder layers

Autoencoders differ according to the types, quantities, and connection methods of artificial neural networks they contain, and they are selected according to the problem being studied and the suitability of the data. Connecting the feature vector layer of a single autoencoder to the input layer of a new single autoencoder provides the stacked autoencoder. Figure 4.8 illustrates the stacked autoencoder, generated by sequentially connected four single autoencoders.

Combining an LSTM encoder at the input and an LSTM decoder layer at the output constructs the LSTM autoencoder. LSTM is an ideal feature extractor for data in time series form, thanks to its ability to recall historical data. Figure 4.9 illustrates the LSTM autoencoder obtained from an LSTM encoder layer at the input and an LSTM decoder layer at the output.

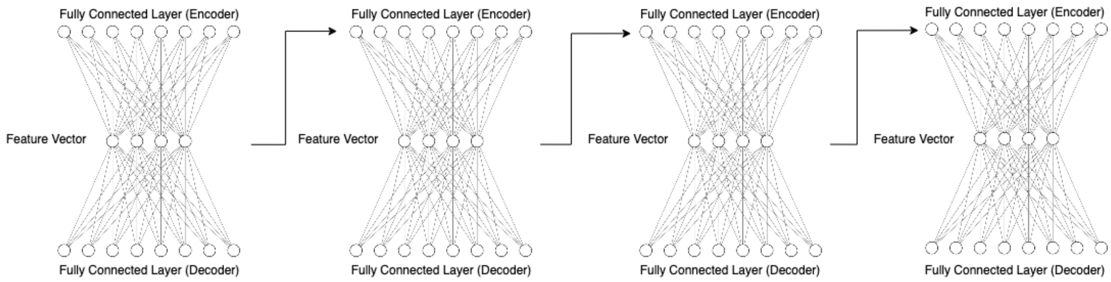

Figure 4.8 A stacked autoencoder consisting of four single autoencoders

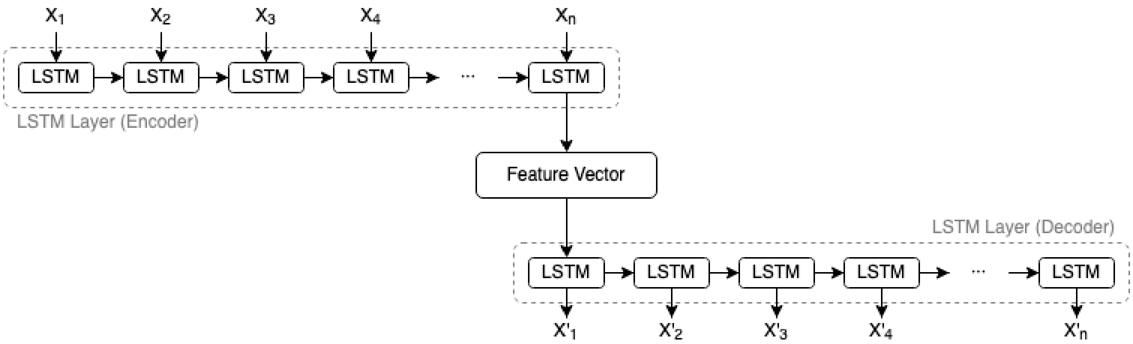

Figure 4.9 An LSTM autoencoder consisting of LSTM encoder and decoder layers

4.6 Windowing the Data Set

Data set windowing is splitting data in time series format into mini-groups to be processed at once. The data set windowing operation windows independent variables (inputs) and dependent variable to be forecasted (output) together. The window size determines how many samples each window contains.

Windowing for the entire data set proceeds in sliding windows logic. To window the whole data set, the variables of some of the samples that fit in a fixed size window are added to the input set, and the output value of the next sample is added to the output set. This process is performed sequentially as a chain for the entire data set by shifting the window each time.

4.7 Forecasting Model

The forecasting model runs the training algorithm and produces the deep learning model by using the assembled layers in the established architecture. Deep learning architectures established for numerical forecasting problems (i.e., regression) can consist of many kinds of layers in hidden layers, such as fully connected, LSTM, RNN, or CNN layers. However, the output layer is often the one-unit fully connected layer to determine the forecasting result [63].

4.8 Test

The test step evaluates the test data to measure the model performance after the forecasting model training. The analytical application applies the same operations to the test data, such as preprocessing, imputation, feature extraction, and the other steps in the pipeline, so the test data passes through the steps that the train data passed.

How close a forecasted point is to its actual value measures the performance in regression problems. Therefore, the closeness of the result to zero gives information about the goodness of the forecasting. A small error value represents a very good forecast; a large error value represents a bad forecast. There are different metrics developed for performance measurement. Root mean square error (RMSE) and mean absolute error (MAE) are two of these metrics frequently used. Equality 4.15 formulates RMSE, Equality 4.16 formulates MAE. m is the number of elements tested; y_i is the value of the i.th element; $\hat{y}_i$ represents the forecasted value for the i.th element in the formulas [64].

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{4.15}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|\hat{y}_i - y_i| \tag{4.16}$$

5
PROTOTYPE APPLICATION DETAILS

Figure 5.1 illustrates our proposed workflow, which includes data acquisition, preprocessing, feature engineering, forecasting model training, and testing for our price forecasting work in the B2C market.

Our proposed solution method consists of a data collection and processing pipeline within the scope of advanced analytics. We prepare a web scraper application that collects the flight ticket price data for a period of time in a daily interval in the tourism area, which is an example of a B2C market. We also prepare a data processing pipeline that consists of several steps, such as data preprocessing, feature extraction, model training, and testing. The data preprocessing step transforms the collected data into an appropriate and usable format. The feature extraction step extracts the deep features with autoencoders and calculates the technical indicators. The model training step creates a forecasting model with some portion of the data and deep learning technique; the testing step tests the model with the remaining part of the data. In this section, we explain the methods used in the mentioned steps and sub-steps and, in that way, how we model the financial time series data within the scope of the thesis.

5.1 Airfare Ticket Price Data Collection with Web Scraper

The automated web scraper lists the results by searching for the departure and landing cities for the determined flight route and the next day of the working time of the web scraper as the date on the visited ticket sales page. Then, we make the scraper collect the price data daily and automatically by encapsulating the found results and storing them on the hard disk in comma-separated values (CSV) file format.

Figure 5.2 and Figure 5.3 show the search window of a web scraper working on June 6, 2022, searching for a flight ticket from Istanbul to Ankara, and the results, respectively.

When encapsulating the listed results to records, the web scraper saves the product price, the departure and landing cities (route), the business company, and the flight

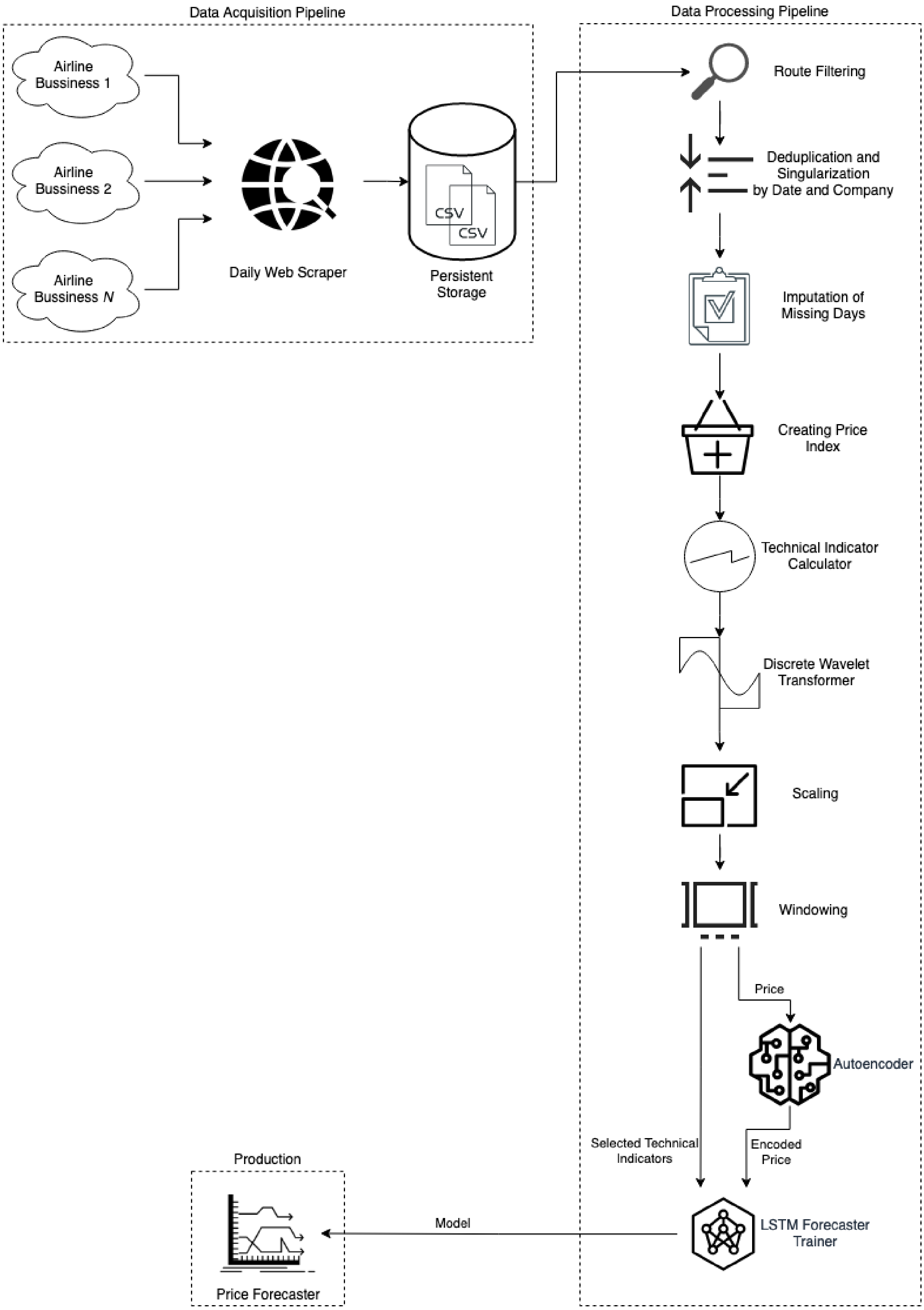

Figure 5.1 The proposed data acquisition and processing architecture

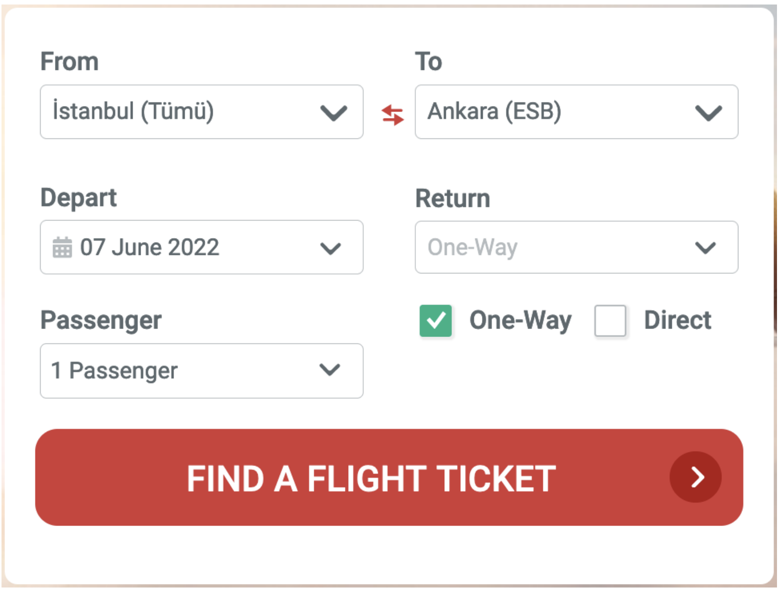

Figure 5.2 The ticket search window with departure, landing cities and date selection (Image: Obilet.com)

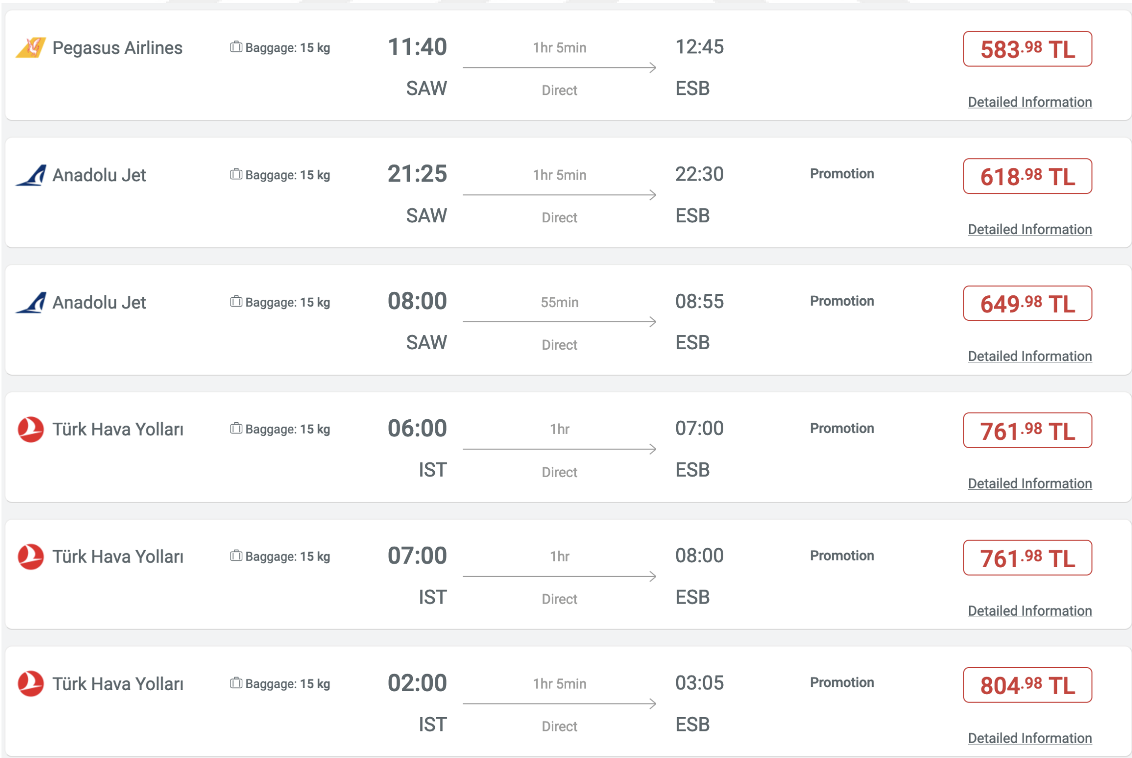

Figure 5.3 Some of the flights listed in the search result from Istanbul to Ankara for 7 June 2022 and their details (Image: Obilet.com)

information, including the date and time to a CSV file.

5.2 Preprocess

While preprocessing the time series flight ticket price data collected in our study, we first filter the collected records by the route and deduplicate repeated records. Then, we group and take the average of the multiple records by date and company fields to singularize them, impute the missing records, and create the price index with the sum of the intraday ticket prices of airline companies.

We explain the DWT method to remove the noise in the data in Section 5.4 and explain data scaling in Section 5.5 since we apply them after the technical indicator calculation.

5.2.1 Deduplication and Singularization of The Flight Records

In the data set collected in our study, there are duplicate (repeated) and multiple records as the web scraper collects the ticket prices of all airline companies and all times of day in the listed results. Running the web scraper multiple times on the same day or repeatedly collecting the same record within a run creates repeat records in the data set. The ticket price records collected by the web scraper for the same day and different hours of the same company create a multiplicity.

We deduplicate the repeat records. We singularize multiple records collected for different hours of the same day by taking their arithmetic averages after grouping day and company dimensions since our time resolution is in day granularity. Thanks to singularization, the ticket price of each airline company for a day becomes the average of the ticket prices of the company on different flights of the day.

5.2.2 Imputation of Missing Flight Records

An error that the web scraper receives at runtime or that some tickets for the searching date are not listed on the ticket sales page appear as missing in the data set. We use the arithmetic average of the values of the missing point in the previous and following weeks to impute the missing days in the timeline. If there is a lack of data in any of the weeks included in the average, we use only the previous week, and if the previous week's data is missing, we use only the next week. We cyclically repeat these three steps until there is no missing data. Consequently, we impute the missing days in the first week without the previous week and the missing days in the last week without the following week with a one-sided from the existing side.

5.2.3 Creating the Ticket Price Index

We need to create the daily price index because of the flights organized by more than one company on the same route and the variable prices between companies as a result of the free price policy. We calculate the daily price values of the time series data set in our study as a price index. The price index is the intraday sum of the preprocessed ticket prices of different companies. Companies contribute to the price index with the same weight since there is no distinction between the companies, like a simple (not weighted) consumption basket.

5.3 Calculation of Technical Indicators

We use the ticket price index as the close price to calculate the technical indicators in our study. Table 5.1 presents the abbreviations and explanations of the technical indicator names we use in our work. The short-term period for the MACD indicator is 12 days and the long-term period is 26 days. The period for the ROC indicator is two days.

Table 5.1 Abbreviations and descriptions of technical indicators used

Indicator Name	Description
MA5	5 Days Period Moving Average
MA10	10 Days Period Moving Average
MA20	20 Days Period Moving Average
TRIX20	20 Days Period Triple Exponential Moving Average
MACD	Moving Average Convergence/Divergence
ROC	Rate of Change

5.4 Discrete Wavelet Transform

We apply the discrete wavelet transform to the price and each technical indicators data in time series form to eliminate noise along the timeline.

We explain the convolution process and the filter used during the convolution in Section 2.2.1.1. Wavelet is the filter used in the discrete wavelet transform method. There are various wavelet families. Haar, Daubechies (db), Coiflet (coif), and Symlet (sym) wavelets are some known wavelets. Figure 5.4 illustrates some wavelets provided by the PyWavelets package using the Plotly library. The number after the abbreviation of the wavelet name represents the order of the wavelet [58, 65].

The denoised signal which is the result of the DWT is similar to the wavelet used. Therefore, the choice of wavelet in DWT is important [65]. We use the Daubechies

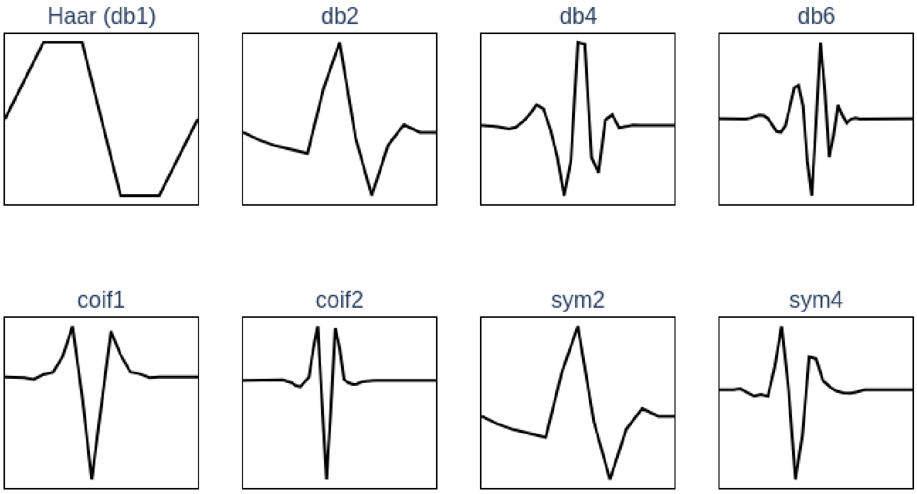

Figure 5.4 Some of the Haar, Daubechies, Coiflet, and Symlet wavelets

wavelet in the DWT method since the characters of the Daubechies wavelet and our data set in numerical time series form are similar. Using Haar (db1) wavelet, which produces cornered results in DWT, can produce such an output where at least two consecutive points will have the same value. For this reason, please note using the Haar (db1) wavelet in DWT is misleading for the forecaster model in our study where we forecast the next day's price in the numerical time series data set.

5.5 Scaling the Data Set

We scale the price data and technical indicators in our data set to values between 0 and 1 using the technique described in Section 4.2.5 before using them in deep learning models. Scaling transforms the smallest value within the feature to 0, the largest value to 1, and intermediate values to decimal numbers between 0 and 1.

5.6 Windowing the Data Set

The price data in time series form becomes useable in the regression model training after windowing. We window the data set by accepting the price index data and technical indicators within a window size as inputs and the price index value on the next day as the output.

5.7 Extracting Deep Features with Autoencoder

In our work, we use single, stacked, and LSTM autoencoders to extract deep features of time series price index data, as explained in Section 4.5. We get the stacked autoencoder by assembling four single autoencoders. In our study, we also propose to use the LSTM autoencoder as a deep feature extractor for the time series price data. Please note that this is a different use for the LSTM autoencoder than the use covered in previous studies in the literature.

5.8 Forecaster Deep Learning Model

The problem we want to forecast daily flight ticket price index is a regression problem as we are trying to make numerical forecasting. We forecast the daily flight ticket price index with an LSTM network using price index data, technical indicators, and deep features in our study. Figure 5.5 illustrates the LSTM forecaster, created by assembling the input layer, the LSTM layer, and the fully connected output layer.

The LSTM forecaster takes the encoded price data extracted by an autoencoder and technical indicators from the input layer. In the experiments without autoencoder, we use unencoded price index data; in experiments without technical indicators, we use price index data. The LSTM layer transfers the results to the one-unit fully connected layer to produce the forecasting result.

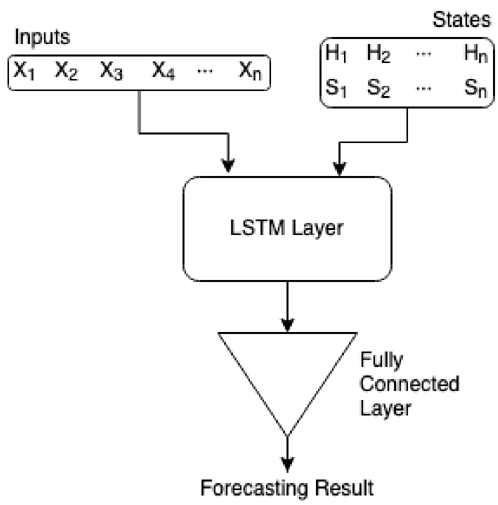

Figure 5.5 The forecaster LSTM network architecture whose input is price data divided into windows and external states

5.9 Software and Hardware Details of Application

The web scraper application prepared for the data collection workflow works with the Scrapy 2.3 package on Python 3.8.5 version. The web scraper collects the ticket prices of Turkish Airlines, AnadoluJet, Pegasus Airlines, and SunExpress companies from the online ticket search and sales portal Obilet.com by searching one day after the runtime date of the application.

Analytical application prepared for data processing workflow runs on Python 3.6.9 with Keras 2.4.3, TensorFlow 2.2, PyWavelets 1.1.1, Stockstats 0.3.2, and scikit-learn 0.24.1 packages.

Two different systems run the analytical application. The first system used has "Intel (R) Xeon (R) Gold 6138 CPU @ 2.00GHz" processor, 504 gigabytes (GB) memory, and 4 "Tesla V100-SXM2-16GB" graphics processor hardware features and Ubuntu 16.04 operating system. The second system used has "Intel Core i7-6700 CPU @ 3.40GHz x 8" processor, 16 GB memory hardware features, and Debian 11.4.0 operating system.

6
EVALUATION OF THE METHOD

We explain the details of the data set used during the experiments, the setup of the experiments, and the test results in this section.

6.1 Data Set

The web scraper application described in the Section 4.1 and Section 5.1 collects the data set used in our study. We started the web scraper first on 27.07.2020, and the web scraper application collected the domestic flight ticket prices of different companies in Turkish Lira for 158 days until 31.12.2020.

We use the records of the top 10 routes with the most records among all collected routes in the experiments in our study. Table 6.1 presents the quantities of raw records according to the companies of the top 10 routes with the most records collected with the web scraper. Table 6.2 presents the quantities of the unique records left after the row-based record deduplication process applied during preprocessing. We do not use the records collected for the 'SunExpress' company since it does not have flights for all routes, and the number of records is relatively low compared to other companies. Table 6.3 presents the final record quantities we use in the experiments, assigning an identifying route number to each route to simplify following up in the experiments. After preprocessing step, we express each route with 158 records which is the total number of days.

6.2 Experiments

In our study, we prepare two different experimental setups by using the proposed solution method to answer our research questions. The first experiment measures the performance of the discrete wavelet transform and autoencoders on forecasting. The second experiment measures the effect of technical indicators on forecasting.

Table 6.1 Total raw flight ticket record quantities by routes and companies collected with the web scraper

	Turkish Airlines	AnadoluJet	Pegasus Airlines	SunExpress	Total
Antalya to İzmir	2164	1776	5009	541	9490
Antalya to Ankara	2455	1759	2758	5	6977
İstanbul to Ankara	1710	3441	1486	0	6637
İstanbul to İzmir	1456	1830	2813	0	6099
Trabzon to İzmir	1689	1369	2443	152	5653
Trabzon to Ankara	1725	1332	2209	0	5266
Ankara to İzmir	2613	2018	620	0	5251
İzmir to Ankara	2555	1775	624	0	4954
Adana to İzmir	2107	1433	398	527	4465
Gaziantep to İzmir	1609	1019	1487	347	4462
Total	20083	17752	19847	1572	59254

Table 6.2 Unique record quantities after row based deduplication

	Turkish Airlines	AnadoluJet	Pegasus Airlines	SunExpress	Total
Antalya to İzmir	1434	1127	3258	368	6187
İstanbul to Ankara	1192	2346	1006	0	4544
Antalya to Ankara	1585	1126	1768	5	4484
İstanbul to İzmir	1025	1227	1882	0	4134
Trabzon to İzmir	1134	864	1707	103	3808
Ankara to İzmir	1811	1358	425	0	3594
Trabzon to Ankara	1131	864	1461	0	3456
İzmir to Ankara	1752	1178	437	0	3367
Gaziantep to İzmir	1080	662	1018	232	2992
Adana to İzmir	1408	902	261	345	2916
Total	13552	11654	13223	1053	39482

Table 6.3 Final record quantities used in experiments

	Turkish Airlines	AnadoluJet	Pegasus Airlines	Total
Route 1: Antalya to İzmir	1434	1127	3258	5819
Route 2: İstanbul to Ankara	1192	2346	1006	4544
Route 3: Antalya to Ankara	1585	1126	1768	4479
Route 4: İstanbul to İzmir	1025	1227	1882	4134
Route 5: Trabzon to İzmir	1134	864	1707	3705
Route 6: Ankara to İzmir	1811	1358	425	3594
Route 7: Trabzon to Ankara	1131	864	1461	3456
Route 8: İzmir to Ankara	1752	1178	437	3367
Route 9: Gaziantep to İzmir	1080	662	1018	2760
Route 10: Adana to İzmir	1408	902	261	2571
Total	13552	11654	13223	38429

6.2.1 Common Configurations of Experiments

In the preprocessing step, we apply Daubechies (db4) DWT function to the price and technical indicators in the data set with one-level decomposition. During the windowing of the samples of the time series data set into sequential parts, the window size is five. We use the Adam, which is a special stochastic gradient descent function, as the optimizer; and the Sigmoid as the activation function in the neural network layers. The feature vector size is 16 for all of the single, stacked, and LSTM autoencoder types. We determine the window and feature vector sizes by trial and error. We create the stacked autoencoder by connecting four single autoencoders sequentially. The LSTM forecaster, which has 100 hidden units in the LSTM layer and then connects to a single unit fully connected output layer to produce the forecasting result, is in the common forecaster model architecture for all experiments. We use the first 67% (105 days) of the data set for training and the second and last part of the data set 33% (53 days) for testing.

6.2.2 Configurations of the First Experiment

In the first experiment, we aim to measure the performance of DWT and autoencoders. Using the DWT on the data has two different alternatives: applying DWT or not. There are three different autoencoder types: single, stacked, and LSTM autoencoders, introduced in Section 4.5. Thereby, using the autoencoder in experiments has four different alternatives in total, also considering the training of the forecasting model without an autoencoder. Thus, the first experiment produces results in a total of eight different combinations per route.

6.2.3 Configurations of the Second Experiment

In the second experiment, we experiment on which technical indicators are advantageous to use with price data in the B2C market. We train and test models with all subsets of technical indicators introduced in Section 4.3 and Section 5.3 to determine which technical indicators provide the best benefit when used in conjunction with price data. Price data (included in each subset) and subsets of six different technical indicators produce $2^6 = 64$ different results in total per autoencoder. It is important to list the group of technical indicators with the lowest error of each subset containing n elements. Therefore, to find the best technical indicator group at the end of the tests, we compare the technical indicator groups containing n elements of the subsets.

6.3 Evaluations

We run experiments with test data we introduce in the Section 6.1 Data Set and with the experimental setups we describe in Section 6.2 Experiments. We measure the test errors of the models through the performance metrics we introduce in Section 4.8 Test.

In the first experiment, the test results of different routes show that using the DWT regardless of the autoencoder used is beneficial, and using an autoencoder most generally works better than not using an autoencoder. Table 6.4 and Table 6.5 present the test results of the first experiment.

In the second experiment, the test results of different routes show that using a selected set of technical indicators according to the route in the price forecasting model is most generally beneficial. Table 6.6 and Table 6.7 present the test results of the second experiment for LSTM AE encoded evaluations. We present the detailed results of the second experiment using other autoencoders in an expanded form in Tables A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8, A.9 and A.10 under Section A.

Table 6.4 Evaluation results of first experiment (for Route 1, 2, 3, 4, 5)

	RMSE		MAE	
	With DWT	Without DWT	With DWT	Without DWT
Route 1: Antalya to İzmir				
LSTM AE	0.11592	0.15787	0.09444	0.13987
SAE	0.11040	0.17052	0.08477	0.15119
Single AE	**0.10509**	0.16907	**0.07699**	0.14844
No AE	0.10775	0.13600	0.08034	0.11168
Route 2: İstanbul to Ankara				
LSTM AE	**0.07694**	0.12850	**0.05963**	0.10320
SAE	0.08803	0.11919	0.06311	0.09163
Single AE	0.08152	0.12040	0.05863	0.08391
No AE	0.07821	0.12423	0.05663	0.08531
Route 3: Antalya to Ankara				
LSTM AE	0.06487	0.11934	**0.05110**	0.09828
SAE	**0.06408**	0.12895	0.05112	0.10582
Single AE	0.06596	0.09567	0.05320	0.07704
No AE	0.06914	0.09616	0.05769	0.07710
Route 4: İstanbul to İzmir				
LSTM AE	0.07383	0.10489	0.05856	0.08159
SAE	0.07227	0.10868	0.05875	0.08408
Single AE	**0.07008**	0.10745	0.05396	0.08369
No AE	0.07086	0.10925	**0.05315**	0.08590
Route 5: Trabzon to İzmir				
LSTM AE	**0.07723**	0.10739	**0.06341**	0.08762
SAE	0.08945	0.10812	0.07304	0.08877
Single AE	0.08551	0.10586	0.06937	0.08761
No AE	0.08386	0.10651	0.06660	0.08844

Table 6.5 Evaluation results of first experiment (for Route 6, 7, 8, 9, 10)

	RMSE		MAE	
	With DWT	Without DWT	With DWT	Without DWT
Route 6: Ankara to İzmir				
LSTM AE	0.09881	0.12957	0.08009	0.10322
SAE	0.10485	0.11773	0.08577	0.09435
Single AE	0.09732	0.12217	0.07883	0.09856
No AE	**0.09060**	0.11333	**0.07097**	0.09239
Route 7: Trabzon to Ankara				
LSTM AE	**0.08516**	0.15638	**0.06775**	0.11941
SAE	0.09537	0.16332	0.07551	0.13394
Single AE	0.08686	0.13361	0.06937	0.10378
No AE	0.08827	0.12424	0.07060	0.10062
Route 8: İzmir to Ankara				
LSTM AE	**0.08993**	0.13247	**0.06828**	0.10057
SAE	0.09411	0.13440	0.07334	0.10319
Single AE	0.09444	0.13824	0.07351	0.10256
No AE	0.09633	0.13882	0.07499	0.10275
Route 9: Gaziantep to İzmir				
LSTM AE	0.07933	0.11998	0.06036	0.09506
SAE	0.12448	0.11449	0.10002	0.09122
Single AE	0.07976	0.12014	0.05880	0.10253
No AE	**0.07078**	0.10952	**0.05503**	0.09102
Route 10: Adana to İzmir				
LSTM AE	0.08827	0.10404	0.06929	0.08114
SAE	0.08903	0.10237	0.06972	0.07859
Single AE	0.08581	0.10168	0.06741	0.08043
No AE	**0.08412**	0.09850	**0.06533**	0.07565

Table 6.6 LSTM AE encoded evaluation results of second experiment (for Route 1, 2, 3, 4, 5)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
	Route 1: Antalya to İzmir			
1	['close']	0.11592	['close']	0.09444
2	['close','ma5']	0.10829	['close','ma5']	0.08737
3	['close','ma5','roc']	0.10586	['close','macd','roc']	0.08404
4	['close','ma5','ma10','roc']	0.10522	['close','trix20','ma5','roc']	0.08214
5	['close','trix20','ma5','ma10','roc']	0.10311	**['close','trix20','ma5','ma10','roc']**	**0.08054**
6	**['close','ma20','trix20','ma5','ma10','roc']**	**0.10236**	['close','ma20','trix20','ma5','ma10','roc']	0.08254
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.11279	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08836
	Route 2: İstanbul to Ankara			
1	['close']	0.07694	['close']	0.05963
2	['close','roc']	0.07346	['close','ma10']	0.05621
3	['close','trix20','roc']	0.06980	**['close','trix20','roc']**	**0.05215**
4	['close','macd','ma20','trix20']	0.07168	['close','trix20','ma10','roc']	0.05394
5	**['close','macd','ma20','ma5','ma10']**	**0.06849**	['close','macd','ma20','ma5','ma10']	0.05224
6	['close','macd','ma20','trix20','ma5','ma10']	0.07560	['close','macd','ma20','trix20','ma5','ma10']	0.05539
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07975	['close','macd','ma20','trix20','ma5','ma10','roc']	0.05532
	Route 3: Antalya to Ankara			
1	['close']	0.06487	['close']	0.05110
2	**['close','ma10']**	**0.06214**	**['close','ma10']**	**0.04889**
3	['close','trix20','ma10']	0.06576	['close','ma20','ma10']	0.05163
4	['close','trix20','ma5','roc']	0.06652	['close','ma20','ma10','roc']	0.05280
5	['close','ma20','trix20','ma5','roc']	0.06788	['close','ma20','ma5','ma10','roc']	0.05420
6	['close','ma20','trix20','ma5','ma10','roc']	0.06817	['close','ma20','trix20','ma5','ma10','roc']	0.05476
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10176	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08107
	Route 4: İstanbul to İzmir			
1	**['close']**	**0.07383**	**['close']**	**0.05856**
2	['close','ma5']	0.07588	['close','roc']	0.06030
3	['close','macd','trix20']	0.07601	['close','macd','trix20']	0.05936
4	['close','macd','ma20','ma10']	0.07547	['close','macd','ma5','roc']	0.05967
5	['close','macd','ma20','ma10','roc']	0.07941	['close','macd','ma20','ma10','roc']	0.06204
6	['close','macd','ma20','ma5','ma10','roc']	0.08364	['close','macd','ma20','ma5','ma10','roc']	0.06552
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08972	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07113
	Route 5: Trabzon to İzmir			
1	['close']	0.07723	['close']	0.06341
2	['close','ma5']	0.07622	['close','ma5']	0.06320
3	**['close','macd','ma20']**	**0.07541**	**['close','macd','ma20']**	**0.06193**
4	['close','macd','ma5','ma10']	0.07578	['close','macd','ma5','ma10']	0.06225
5	['close','macd','ma20','ma5','ma10']	0.07751	['close','macd','ma20','ma5','ma10']	0.06321
6	['close','macd','ma20','ma5','ma10','roc']	0.07727	['close','macd','ma20','ma5','ma10','roc']	0.06340
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08995	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07344

Table 6.7 LSTM AE encoded evaluation results of second experiment (for Route 6, 7, 8, 9, 10)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
Route 6: Ankara to İzmir				
1	**['close']**	**0.09881**	['close']	0.08009
2	['close','roc']	0.10010	['close','roc']	0.07984
3	['close','ma10','roc']	0.10014	['close','macd','ma5']	0.08040
4	['close','macd','ma20','ma10']	0.10039	**['close','macd','ma5','ma10']**	**0.07900**
5	['close','macd','ma20','ma5','ma10']	0.10135	['close','macd','ma20','ma5','ma10']	0.08053
6	['close','macd','ma20','ma5','ma10','roc']	0.10980	['close','macd','ma20','ma5','ma10','roc']	0.08474
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.16819	['close','macd','ma20','trix20','ma5','ma10','roc']	0.12634
Route 7: Trabzon to Ankara				
1	['close']	0.08516	['close']	0.06775
2	**['close','ma5']**	**0.07983**	**['close','ma5']**	**0.06411**
3	['close','ma5','ma10']	0.08238	['close','ma5','ma10']	0.06642
4	['close','ma5','ma10','roc']	0.08255	['close','ma5','ma10','roc']	0.06661
5	['close','ma20','ma5','ma10','roc']	0.08734	['close','ma20','ma5','ma10','roc']	0.06968
6	['close','macd','ma20','ma5','ma10','roc']	0.10393	['close','macd','ma20','ma5','ma10','roc']	0.08233
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.12915	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10553
Route 8: İzmir to Ankara				
1	['close']	0.08993	['close']	0.06828
2	['close','roc']	0.08804	['close','ma5']	0.06627
3	['close','macd','ma5']	0.08604	['close','macd','ma5']	0.06019
4	**['close','macd','ma20','roc']**	**0.08389**	**['close','macd','ma20','roc']**	**0.05980**
5	['close','macd','ma5','ma10','roc']	0.08547	['close','macd','ma5','ma10','roc']	0.06167
6	['close','macd','ma20','ma5','ma10','roc']	0.08950	['close','macd','ma20','ma5','ma10','roc']	0.06708
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10688	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08278
Route 9: Gaziantep to İzmir				
1	['close']	0.07933	['close']	0.06036
2	['close','macd']	0.07909	['close','roc']	0.06151
3	['close','trix20','roc']	0.07849	['close','trix20','roc']	0.06082
4	['close','ma20','trix20','roc']	0.07803	['close','trix20','ma5','roc']	0.05962
5	**['close','macd','trix20','ma5','roc']**	**0.07729**	['close','macd','trix20','ma5','roc']	0.05956
6	['close','macd','ma20','trix20','ma5','roc']	0.07791	**['close','macd','ma20','trix20','ma5','roc']**	**0.05908**
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08425	['close','macd','ma20','trix20','ma5','ma10','roc']	0.06244
Route 10: Adana to İzmir				
1	['close']	0.08827	['close']	0.06929
2	['close','roc']	0.08728	['close','ma10']	0.06825
3	['close','ma5','roc']	0.08651	['close','ma5','roc']	0.06809
4	**['close','ma5','ma10','roc']**	**0.08636**	**['close','ma5','ma10','roc']**	**0.06794**
5	['close','ma20','ma5','ma10','roc']	0.08650	['close','ma20','ma5','ma10','roc']	0.06830
6	['close','macd','ma20','trix20','ma5','ma10']	0.09102	['close','macd','ma20','trix20','ma5','ma10']	0.07083
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10430	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07905

7
RESULTS AND DISCUSSION

In this thesis, we establish and propose a deep learning-based forecasting pipeline to forecast the price of a product or service in B2C markets and to improve the forecasting results. The workflow of this forecasting pipeline consists of a web scraper to collect price data, various preprocessing methods to ready the collected data for processing, analytical and deep feature extractors for feature engineering studies, and a deep learning forecaster network to create a forecasting model. With the proposed price forecasting workflow methodology, we forecast the price indexes of airfare tickets in B2C markets within the scope of the study.

We develop an advanced analytics application that implements our proposed price forecasting workflow. Results of the experiments we evaluate using our advanced analytics application show that price index forecasting at an acceptable level and improving the forecasting results in B2C markets is possible using different techniques with the proposed method in our study.

7.1 Answers to Research Questions

This section presents the answers to the research questions introduced in Chapter 3 Research Questions in the light of the findings obtained as a result of the experiments performed in Chapter 6 Evaluation of the Method.

7.1.1 Could Time Series Price Data and Deep Learning Be Used to Forecast Price in B2C Markets?

During the literature review, we discovered that the forecasting studies carried out in stock markets are similar to the mainstream subject of our work in terms of processing time series data sets, being in the financial field, and being a widely traded market. Based on the positive feedback observed from these studies, we continue our studies and seek answers to other questions. As a result, we create a price index forecasting model with deep learning using time series price data.

7.1.2 Which Preprocessing Methods Need to Be Used in Time Series Price Data?

The preprocessing need of each data set varies according to the content of the data set and the data set's role in the study. Section 4.2 Preprocess introduces the preprocessing methods that the financial time series data set needs in our work. We share the positive contribution of the DWT method described in Section 4.2.4, which we use as a noise remover and data smoother in our study, in the Section 6.3 Evaluations.

7.1.3 Which Deep Learning Methods Can Be Used to Extract Features from Price Data?

We use single, stacked, and LSTM autoencoders to extract deep features from time series price data. We share the performance of autoencoders comparatively in the Section 6.3 Evaluations. We observe that the LSTM autoencoder, which we propose to use as a deep feature extractor for the time series price data in our study, positively contributes to the success of forecasting models.

7.1.4 Are Technical Indicators to Be Derived from Price Data Useful for Price Forecasting in the B2C Market?

To answer the last research question of our study, we examine the effect of technical indicators on price forecasting work in the B2C market. Findings from the results of experiments for all subsets of technical indicators in the Section 6.3 Evaluations show that using technical indicators is usually beneficial for creating forecasting models to forecast prices in the B2C market.

7.2 Result

Businesses and consumers are actively involved in the B2C market. Forecasting the next price of a product or service becomes more and more important in the B2C market, as in all other financial markets. Works in financial markets use various preprocessing, feature engineering, and deep learning methods in different combinations to make forecastings and reduce forecasting errors; these methods are open to improvement for new studies.

In our thesis, we study the use of historical price data and deep learning methods, preprocessing methods used in time series price data, extracting deep features with autoencoders, and analytically calculated technical indicators to forecast prices and optimize price forecasting to more accurate results in the B2C market. Results of experiments in our work, in which we study forecasting airfare price indexes and optimizing the forecasting results in the B2C market, conclude that price forecasting with deep learning is possible; optimizing forecasting results is open to improvements with new techniques. We report that applying the DWT method to the time series data set in the preprocessing step for noise removal and data smoothing reduces the forecasting model error. Also, using an autoencoder to extract deep features of price data and using the technical indicators in the forecasting model most generally contribute to creating successful forecasting models. The contributions of our thesis match the main motivations of our work.

7.3 Validity of Results

The proposed solution methodology and results are dependent on the data we use. Different data or implementations may produce different results.

7.4 Future Works

Future studies can include different data processing methods or the usage of techniques. For example, using not only the close price but also different price types, such as open, close, high, or low, allows the calculation of other technical indicators. On the other hand, different deep feature extractors, such as CNN, can take place in the deep feature extraction step. As a combination, new feature extractors can perform feature extraction by using new price types in feature engineering.

REFERENCES

[1] M. Shareef, V. Kumar, and U. Kumar, "Business-to-consumer (b2c) electronic-commerce (ec) operation: Globalization strategy," in *ASAC Conference Proceedings*, 2008, pp. 145–158.

[2] P. He, S. Zhang, and C. He, "Impacts of logistics resource sharing on b2c e-commerce companies and customers," *Electronic Commerce Research and Applications*, vol. 34, p. 100 820, 2019.

[3] *Electronic commerce (ecommerce)*. [Online]. Available: https://www.investopedia.com/terms/e/ecommerce.asp (visited on 12/06/2021).

[4] *Retail e-commerce sales worldwide from 2014 to 2024*. [Online]. Available: https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/ (visited on 11/11/2021).

[5] *2021 yılı İlk 6 ayı e-ticaret verileri açıklandı*. [Online]. Available: https://www.eticaret.gov.tr/haberler/10080/detay (visited on 12/06/2021).

[6] *E-ticaret platformlarında 65 yaş üstü kullanıcı oranı yüzde 6'dan yüzde 10'a çıktı*, 2021. [Online]. Available: https://www.aa.com.tr/tr/ekonomi/e-ticaret-platformlarinda-65-yas-ustu-kullanici-orani-yuzde-6dan-yuzde-10a-cikti/2182333 (visited on 12/12/2022).

[7] *İstatistikler*. [Online]. Available: https://www.eticaret.gov.tr/istatistikler (visited on 12/11/2021).

[8] *Digital 2021: Turkey*. [Online]. Available: https://datareportal.com/reports/digital-2021-turkey (visited on 12/11/2021).

[9] F. S. Rawnaque, K. M. Rahman, S. F. Anwar, R. Vaidyanathan, T. Chau, F. Sarker, and K. A. A. Mamun, "Technological advancements and opportunities in neuromarketing: A systematic review," *Brain Informatics*, vol. 7, no. 1, pp. 1–19, 2020.

[10] Y. Choi, and D. Q. Mai, "The sustainable role of the e-trust in the b2c e-commerce of vietnam," *Sustainability*, vol. 10, no. 1, p. 291, 2018.

[11] A. A. N. Alkenani, "Factors influencing social e-commerce success in saudi arabia-a review," in *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, 2019, pp. 1331–1335.

[12] S. Carta, A. Medda, A. Pili, D. Reforgiato Recupero, and R. Saia, "Forecasting e-commerce products prices by combining an autoregressive integrated moving average (arima) model and google trends data," *Future Internet*, vol. 11, no. 1, 2019.

[13] J. M. Perloff, *Microeconomics*. Pearson Education, 2009.

[14] K. Zhao, and C. Wang, “Sales forecast in e-commerce using convolutional neural network,” *arXiv preprint arXiv:1708.07946*, 2017.

[15] P.-H. Hsieh, “A study of models for forecasting e-commerce sales during a price war in the medical product industry,” in *HCI in Business, Government and Organizations. eCommerce and Consumer Behavior*, F. F.-H. Nah, and K. Siau, Eds., Cham: Springer International Publishing, 2019, pp. 3–21.

[16] R. Yan, and S. Ghose, “Forecast information and traditional retailer performance in a dual-channel competitive market,” *Journal of Business Research*, vol. 63, no. 1, pp. 77–83, 2010.

[17] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran, and B. Seaman, “Sales demand forecast in e-commerce using a long short-term memory neural network methodology,” in *Neural Information Processing*, T. Gedeon, K. W. Wong, and M. Lee, Eds., Cham: Springer International Publishing, 2019, pp. 462–474.

[18] A. Gürbüz, and M. S. Aktaş, “Prediction of purchase intention on the e-commerce clickstream data,” in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, 2019, pp. 1–4.

[19] A. Fronzetti Colladon, B. Guardabascio, and R. Innarella, “Using social network and semantic analysis to analyze online travel forums and forecast tourism demand,” *Decision Support Systems*, vol. 123, p. 113 075, 2019.

[20] R. Law, G. Li, D. K. C. Fong, and X. Han, “Tourism demand forecasting: A deep learning approach,” *Annals of Tourism Research*, vol. 75, pp. 410–423, 2019.

[21] Y. Li, and H. Cao, “Prediction for tourism flow based on lstm neural network,” *Procedia Computer Science*, vol. 129, pp. 277–283, 2018.

[22] *Beginner’s guide to sales forecasting methodology*. [Online]. Available: `https://www.anaplan.com/blog/sales-forecasting-guide/` (visited on 02/15/2022).

[23] M. Aktas, and M. Astekin, “Provenance aware run-time verification of things for self-healing internet of things applications,” *Concurrency and Computation: Practice and Experience*, 2019.

[24] A. Tufek, and M. S. Aktas, “On the provenance extraction techniques from large scale log files,” in *Euro-Par 2020: Parallel Processing Workshops*, M. Malawski, and K. Rzadca, Eds., Euro-Par 2020, Springer, 2020.

[25] Y. Tas, M. Baeth, and M. Aktas, “An approach to standalone provenance systems for big social provenance data,” *2016 12th International Conference on Semantics, Knowledge and Grids (SKG), 9-16*, 2016.

[26] M. Riveni, T. Nguyen, M. Aktas, and S. Dustdar, “Application of provenance in social computing: A case study,” *Concurrency and Computation: Practice and Experience 31 (3)*, 2019.

[27] M. Baeth, and M. Aktas, “An approach to custom privacy policy violation detection problems using big social provenance data,” *Concurrency and Computation: Practice and Experience 30 (21)*, 2018.

[28] M. Baeth, and M. Aktaş, "Detecting misinformation in social networks using provenance data," *Concurrency and Computation: Practice and Experience 31 (3)*, 2019.

[29] S. Jensen, B. Plale, M. Aktas, Y. Luo, P. Chen, and H. Conover, "Provenance capture and use in a satellite data processing pipeline," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 11, 2013.

[30] A. Tufek, A. Gurbuz, O. F. Ekuklu, and M. S. Aktas, "Provenance collection platform for the weather research and forecasting model," in *2018 14th International Conference on Semantics, Knowledge and Grids (SKG)*, ser. SKG '18, 14th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, Guangzhou, China: IEEE, 2018, pp. 17–24.

[31] I. Yazıcı, E. Karabulut, and M. Aktas, "A data provenance visualization approach," *The 14th International Conference on Semantics, Knowledge and Grids*, 2018.

[32] Y. Uygun, R. Oguz, E. Olmezogullari, and M. Aktas, "On the large-scale graph data processing for user interface testing in big data science projects," IEEE BigData 2020, IEEE, 2020, pp. 2049–2056.

[33] E. Olmezogullari, and M. Aktas, "Representation of click-stream datasequences for learning user navigational behavior by using embeddings," IEEE BigData 2020, IEEE, 2020, pp. 3173–3179.

[34] H. T. Kesgin, S. Shakeri, N. Bulut, S. Yüzük, and M. S. Aktaş, "Bankrupcy risk forecast based on company balance sheet data using machine learning," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, IEEE, 2019, pp. 195–200.

[35] S. W. Smith, *The scientist and engineer's guide to digital signal processing*. California Technical Pub. San Diego, California, 1997.

[36] M. B. Perry, "The weighted moving average technique," *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[37] S.-D. Jiang, Z.-M. Lu, and S.-H. Sun, "Morphological filter based noise removal from vibration signals of fighter plane," in *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No. 01EX489)*, IEEE, 2001, pp. 251–254.

[38] X. Hu, Z. Xiao, and N. Zhang, "Removal of baseline wander from ecg signal based on a statistical weighted moving average filter," *Journal of Zhejiang University SCIENCE C*, vol. 12, no. 5, pp. 397–403, 2011.

[39] S. V. Vaseghi, *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.

[40] Y. Liu, C. Liu, and D. Wang, "A 1d time-varying median filter for seismic random, spike-like noise elimination," *Geophysics*, vol. 74, no. 1, pp. V17–V24, 2009.

[41] P. Chaovalit, A. Gangopadhyay, G. Karabatis, and Z. Chen, "Discrete wavelet transform-based time series analysis and mining," *ACM Comput. Surv.*, vol. 43, no. 2, Feb. 2011.

[42] A. J. Conejo, M. A. Plazas, R. Espinola, and A. B. Molina, "Day-ahead electricity price forecasting using the wavelet transform and arima models," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 1035–1042, 2005.

[43] F. Lazzeri, *Machine learning for time series forecasting with Python*. John Wiley & Sons, 2020.

[44] S. B. Achelis, *Technical Analysis from A to Z*. McGraw Hill New York, 2001.

[45] K. Prachyachuwong, and P. Vateekul, "Stock trend prediction using deep learning approach on technical indicator and industrial specific information," *Information*, vol. 12, no. 6, p. 250, 2021.

[46] T.-Y. Kim, and S.-B. Cho, "Predicting residential energy consumption using cnn-lstm neural networks," *Energy*, vol. 182, pp. 72–81, 2019.

[47] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[48] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PloS one*, vol. 12, no. 7, e0180944, 2017.

[49] L. Wang, Z. Zhang, and J. Chen, "Short-term electricity price forecasting with stacked denoising autoencoders," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2673–2681, 2017.

[50] K.-K. Tseng, R. F.-Y. Lin, H. Zhou, K. J. Kurniajaya, and Q. Li, "Price prediction of e-commerce products through internet sentiment analysis," *Electronic Commerce Research*, vol. 18, no. 1, pp. 65–88, 2018.

[51] *6.s085 statistics for research projects: Iap 2015 - linear regression*. [Online]. Available: `https://www.mit.edu/~6.s085/notes/lecture3.pdf` (visited on 05/11/2022).

[52] *B.bogacka - time series - chapter 4. stationary ts models*. [Online]. Available: `https://webspace.maths.qmul.ac.uk/b.bogacka/TimeSeries/TS_Chapter4_5.pdf` (visited on 05/12/2022).

[53] *Dr. gabriel huerta - time series - ar, ma and arma models*. [Online]. Available: `https://math.unm.edu/~ghuerta/tseries/week4_1.pdf` (visited on 05/12/2022).

[54] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on ceemdan and lstm," *Physica A: Statistical Mechanics and its Applications*, vol. 519, pp. 127–139, 2019.

[55] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, vol. 11, no. 7, p. 1636, 2018.

[56] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[57] J. B. Ramsey, and C. Lampart, "The decomposition of economic relationships by time scale using wavelets: Expenditure and income," *Studies in Nonlinear Dynamics & Econometrics*, vol. 3, no. 1, 1998.

[58] J. Han, J. Pei, and H. Tong, *Data mining: concepts and techniques*. Morgan Kaufmann, 2022.

[59] *Instructions module tehnical analysis*. [Online]. Available: https://www.mse.mk/Repository/UserFiles/Image/Baneri/Instructions%20-%20Tehnicka.pdf (visited on 05/24/2022).

[60] *Exponential moving average*. [Online]. Available: https://origin2.cdn.componentsource.com/sites/default/files/resources/dundas/538201/WebChart2005/ExponentialMovingAverage.html (visited on 05/24/2022).

[61] *How is the exponential moving average (ema) formula calculated?* [Online]. Available: https://www.investopedia.com/ask/answers/122314/what-exponential-moving-average-ema-formula-and-how-ema-calculated.asp (visited on 05/24/2022).

[62] *Advantages of triple exponential average (trix)*. [Online]. Available: https://www.investopedia.com/articles/technical/02/092402.asp (visited on 05/24/2022).

[63] F. Chollet, *Deep learning with Python*. Manning New York, 2018.

[64] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2019.

[65] Z. Zhang, Q. K. Telesford, C. Giusti, K. O. Lim, and D. S. Bassett, "Choosing wavelet methods, filters, and lengths for functional brain network construction," *PloS one*, vol. 11, no. 6, e0157243, 2016.

A
APPENDIX A - EXTENDED RESULTS

We present the detailed results of the second experiment in this chapter in Tables A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8, A.9 and A.10.

The detailed results consist of the forecasting test results and features which have the minimum error in the group for all autoencoder types and each route. Therefore, we get and present the best features for each group containing *n* elements to find the best forecasting model.

Table A.1 The best results of second experiment for all autoencoders based on the feature count (for Route 1: Antalya to İzmir)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
LSTM AE Encoded				
1	['close']	0.11592	['close']	0.09444
2	['close','ma5']	0.10829	['close','ma5']	0.08737
3	['close','ma5','roc']	0.10586	['close','macd','roc']	0.08404
4	['close','ma5','ma10','roc']	0.10522	['close','trix20','ma5','roc']	0.08214
5	['close','trix20','ma5','ma10','roc']	0.10311	['close','trix20','ma5','ma10','roc']	0.08054
6	**['close','ma20','trix20','ma5','ma10','roc']**	**0.10236**	['close','ma20','trix20','ma5','ma10','roc']	0.08254
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.11279	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08836
SAE Encoded				
1	['close']	0.11040	['close']	0.08477
2	['close','trix20']	0.10941	['close','trix20']	0.08327
3	['close','ma5','roc']	0.10440	['close','ma5','roc']	0.07846
4	['close','ma5','ma10','roc']	0.10740	['close','ma5','ma10','roc']	0.07826
5	['close','ma20','trix20','ma5','roc']	0.10683	**['close','ma20','trix20','ma5','roc']**	**0.07671**
6	['close','ma20','trix20','ma5','ma10','roc']	0.10959	['close','ma20','trix20','ma5','ma10','roc']	0.08401
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.11641	['close','macd','ma20','trix20','ma5','ma10','roc']	0.09066
Single AE Encoded				
1	['close']	0.10509	['close']	0.07699
2	['close','trix20']	0.10292	['close','macd']	0.07707
3	['close','macd','ma20']	0.10370	['close','macd','ma10']	0.07770
4	['close','macd','ma20','ma5']	0.10461	['close','macd','ma20','ma5']	0.08037
5	['close','ma20','ma5','ma10','roc']	0.10258	['close','trix20','ma5','ma10','roc']	0.07882
6	['close','macd','ma20','ma5','ma10','roc']	0.10443	['close','macd','ma20','ma5','ma10','roc']	0.08181
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.11213	['close','macd','ma20','trix20','ma5','ma10','roc']	0.09127
No AE				
1	['close']	0.10775	['close']	0.08034
2	['close','trix20']	0.10378	['close','ma5']	0.07752
3	['close','trix20','ma5']	0.10321	['close','ma5','roc']	0.07805
4	['close','macd','trix20','ma5']	0.10447	['close','ma5','ma10','roc']	0.07855
5	['close','macd','trix20','ma5','roc']	0.10386	['close','ma20','ma5','ma10','roc']	0.07800
6	['close','macd','trix20','ma5','ma10','roc']	0.10394	['close','macd','trix20','ma5','ma10','roc']	0.08099
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.11035	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08448

Table A.2 The best results of second experiment for all autoencoders based on the feature count (for Route 2: İstanbul to Ankara)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
		LSTM AE Encoded		
1	['close']	0.07694	['close']	0.05963
2	['close','roc']	0.07346	['close','ma10']	0.05621
3	['close','trix20','roc']	0.06980	**['close','trix20','roc']**	**0.05215**
4	['close','macd','ma20','trix20']	0.07168	['close','trix20','ma10','roc']	0.05394
5	**['close','macd','ma20','ma5','ma10']**	**0.06849**	['close','macd','ma20','ma5','ma10']	0.05224
6	['close','macd','ma20','trix20','ma5','ma10']	0.07560	['close','macd','ma20','trix20','ma5','ma10']	0.05539
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07975	['close','macd','ma20','trix20','ma5','ma10','roc']	0.05532
		SAE Encoded		
1	['close']	0.08803	['close']	0.06311
2	['close','roc']	0.08684	['close','roc']	0.06396
3	['close','trix20','roc']	0.08108	['close','trix20','roc']	0.06091
4	['close','macd','trix20','roc']	0.07768	['close','macd','trix20','roc']	0.05886
5	['close','macd','ma20','trix20','roc']	0.07294	['close','macd','ma20','trix20','roc']	0.05622
6	['close','macd','ma20','trix20','ma10','roc']	0.07382	['close','macd','ma20','trix20','ma10','roc']	0.05685
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07495	['close','macd','ma20','trix20','ma5','ma10','roc']	0.05729
		Single AE Encoded		
1	['close']	0.08152	['close']	0.05863
2	['close','roc']	0.07783	['close','ma5']	0.05541
3	['close','trix20','roc']	0.07422	['close','trix20','roc']	0.05391
4	['close','ma20','ma5','roc']	0.07571	['close','macd','ma20','roc']	0.05488
5	['close','macd','ma20','trix20','roc']	0.07372	['close','macd','ma20','trix20','roc']	0.05347
6	['close','macd','ma20','trix20','ma10','roc']	0.07380	['close','macd','ma20','trix20','ma10','roc']	0.05318
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07384	['close','macd','ma20','trix20','ma5','ma10','roc']	0.05424
		No AE		
1	['close']	0.07821	['close']	0.05663
2	['close','ma5']	0.07202	['close','ma5']	0.05246
3	['close','ma5','ma10']	0.07571	['close','ma5','ma10']	0.05422
4	['close','ma20','trix20','ma5']	0.07827	['close','ma20','trix20','ma5']	0.05510
5	['close','trix20','ma5','ma10','roc']	0.07712	['close','ma20','trix20','ma5','ma10']	0.05590
6	['close','ma20','trix20','ma5','ma10','roc']	0.07855	['close','ma20','trix20','ma5','ma10','roc']	0.05599
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08060	['close','macd','ma20','trix20','ma5','ma10','roc']	0.05670

Table A.3 The best results of second experiment for all autoencoders based on the feature count (for Route 3: Antalya to Ankara)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
LSTM AE Encoded				
1	[’close’]	0.06487	[’close’]	0.05110
2	[’close’,’ma10’]	0.06214	[’close’,’ma10’]	0.04889
3	[’close’,’trix20’,’ma10’]	0.06576	[’close’,’ma20’,’ma10’]	0.05163
4	[’close’,’trix20’,’ma5’,’roc’]	0.06652	[’close’,’ma20’,’ma10’,’roc’]	0.05280
5	[’close’,’ma20’,’trix20’,’ma5’,’roc’]	0.06788	[’close’,’ma20’,’ma5’,’ma10’,’roc’]	0.05420
6	[’close’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.06817	[’close’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.05476
7	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.10176	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.08107
SAE Encoded				
1	[’close’]	0.06408	[’close’]	0.05112
2	[’close’,’trix20’]	0.06290	[’close’,’trix20’]	0.05008
3	[’close’,’macd’,’ma5’]	0.06025	**[’close’,’macd’,’ma5’]**	**0.04746**
4	**[’close’,’macd’,’trix20’,’ma5’]**	**0.05999**	[’close’,’macd’,’trix20’,’ma5’]	0.04842
5	[’close’,’macd’,’trix20’,’ma5’,’ma10’]	0.06027	[’close’,’macd’,’trix20’,’ma5’,’ma10’]	0.04912
6	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’]	0.06341	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’]	0.05135
7	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.07998	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.06079
Single AE Encoded				
1	[’close’]	0.06596	[’close’]	0.05320
2	[’close’,’trix20’]	0.06286	[’close’,’macd’]	0.04874
3	[’close’,’ma20’,’trix20’]	0.06262	[’close’,’macd’,’trix20’]	0.04848
4	[’close’,’macd’,’trix20’,’ma10’]	0.06254	[’close’,’macd’,’trix20’,’ma5’]	0.04866
5	[’close’,’macd’,’ma20’,’trix20’,’ma10’]	0.06549	[’close’,’macd’,’ma20’,’trix20’,’ma10’]	0.05057
6	[’close’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.07121	[’close’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.05446
7	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.08788	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.06703
No AE				
1	[’close’]	0.06914	[’close’]	0.05769
2	[’close’,’ma5’]	0.06413	[’close’,’ma5’]	0.05117
3	[’close’,’ma20’,’trix20’]	0.06253	[’close’,’ma5’,’ma10’]	0.04997
4	[’close’,’ma20’,’trix20’,’roc’]	0.06371	[’close’,’ma20’,’trix20’,’ma5’]	0.05058
5	[’close’,’ma20’,’trix20’,’ma10’,’roc’]	0.06630	[’close’,’ma20’,’trix20’,’ma10’,’roc’]	0.05330
6	[’close’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.07844	[’close’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.05995
7	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.08299	[’close’,’macd’,’ma20’,’trix20’,’ma5’,’ma10’,’roc’]	0.06344

Table A.4 The best results of second experiment for all autoencoders based on the feature count (for Route 4: İstanbul to İzmir)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
	LSTM AE Encoded			
1	['close']	0.07383	['close']	0.05856
2	['close','ma5']	0.07588	['close','roc']	0.06030
3	['close','macd','trix20']	0.07601	['close','macd','trix20']	0.05936
4	['close','macd','ma20','ma10']	0.07547	['close','macd','ma5','roc']	0.05967
5	['close','macd','ma20','ma10','roc']	0.07941	['close','macd','ma20','ma10','roc']	0.06204
6	['close','macd','ma20','ma5','ma10','roc']	0.08364	['close','macd','ma20','ma5','ma10','roc']	0.06552
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08972	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07113
	SAE Encoded			
1	['close']	0.07227	['close']	0.05875
2	['close','roc']	0.07315	['close','roc']	0.05852
3	['close','ma5','roc']	0.07593	['close','ma5','roc']	0.05937
4	['close','ma5','ma10','roc']	0.07708	['close','macd','ma20','roc']	0.06080
5	['close','macd','ma20','ma5','roc']	0.07918	['close','macd','ma20','ma5','roc']	0.06103
6	['close','macd','ma20','ma5','ma10','roc']	0.07926	['close','macd','ma20','ma5','ma10','roc']	0.06142
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.09363	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07255
	Single AE Encoded			
1	**['close']**	**0.07008**	['close']	0.05396
2	['close','ma5']	0.07447	['close','roc']	0.05758
3	['close','ma5','ma10']	0.07793	['close','ma5','roc']	0.05970
4	['close','ma20','ma5','ma10']	0.08107	['close','ma5','ma10','roc']	0.06204
5	['close','macd','ma20','ma5','ma10']	0.08240	['close','macd','ma20','ma5','ma10']	0.06279
6	['close','macd','ma20','ma5','ma10','roc']	0.08350	['close','macd','ma20','ma5','ma10','roc']	0.06229
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.09674	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07215
	No AE			
1	['close']	0.07086	**['close']**	**0.05315**
2	['close','ma5']	0.07564	['close','ma5']	0.05706
3	['close','ma5','ma10']	0.08155	['close','ma5','ma10']	0.06104
4	['close','ma5','ma10','roc']	0.08695	['close','ma5','ma10','roc']	0.06460
5	['close','ma20','ma5','ma10','roc']	0.09432	['close','ma20','ma5','ma10','roc']	0.06944
6	['close','macd','ma20','ma5','ma10','roc']	0.10195	['close','macd','ma20','ma5','ma10','roc']	0.07428
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.11301	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08557

Table A.5 The best results of second experiment for all autoencoders based on the feature count (for Route 5: Trabzon to İzmir)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
	LSTM AE Encoded			
1	['close']	0.07723	['close']	0.06341
2	['close','ma5']	0.07622	['close','ma5']	0.06320
3	['close','macd','ma20']	0.07541	['close','macd','ma20']	0.06193
4	['close','macd','ma5','ma10']	0.07578	['close','macd','ma5','ma10']	0.06225
5	['close','macd','ma20','ma5','ma10']	0.07751	['close','macd','ma20','ma5','ma10']	0.06321
6	['close','macd','ma20','ma5','ma10','roc']	0.07727	['close','macd','ma20','ma5','ma10','roc']	0.06340
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08995	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07344
	SAE Encoded			
1	['close']	0.08945	['close']	0.07304
2	['close','macd']	0.08586	['close','macd']	0.07049
3	['close','macd','ma5']	0.08265	['close','macd','ma5']	0.06745
4	['close','macd','ma20','ma5']	0.08155	['close','macd','ma5','ma10']	0.06633
5	['close','macd','ma20','trix20','ma5']	0.08002	['close','macd','ma20','trix20','ma5']	0.06467
6	['close','macd','ma20','trix20','ma5','ma10']	0.08403	['close','macd','ma20','trix20','ma5','ma10']	0.06933
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08537	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07109
	Single AE Encoded			
1	['close']	0.08551	['close']	0.06937
2	['close','macd']	0.07917	['close','macd']	0.06441
3	['close','macd','ma20']	0.07742	['close','macd','ma20']	0.06244
4	['close','macd','ma20','ma5']	0.07728	['close','macd','ma20','ma5']	0.06229
5	['close','ma20','trix20','ma5','ma10']	0.07920	['close','ma20','trix20','ma5','ma10']	0.06525
6	['close','ma20','trix20','ma5','ma10','roc']	0.08265	['close','macd','ma20','ma5','ma10','roc']	0.06813
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08851	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07234
	No AE			
1	['close']	0.08386	['close']	0.06660
2	['close','macd']	0.07698	['close','macd']	0.06158
3	**['close','macd','ma5']**	**0.07374**	**['close','macd','ma5']**	**0.05988**
4	['close','macd','ma5','ma10']	0.07578	['close','ma20','trix20','ma10']	0.06157
5	['close','ma20','trix20','ma5','ma10']	0.07603	['close','ma20','trix20','ma5','ma10']	0.06145
6	['close','ma20','trix20','ma5','ma10','roc']	0.08364	['close','ma20','trix20','ma5','ma10','roc']	0.06723
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10010	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08202

Table A.6 The best results of second experiment for all autoencoders based on the feature count (for Route 6: Ankara to İzmir)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
	LSTM AE Encoded			
1	['close']	0.09881	['close']	0.08009
2	['close','roc']	0.10010	['close','roc']	0.07984
3	['close','ma10','roc']	0.10014	['close','macd','ma5']	0.08040
4	['close','macd','ma20','ma10']	0.10039	['close','macd','ma5','ma10']	0.07900
5	['close','macd','ma20','ma5','ma10']	0.10135	['close','macd','ma20','ma5','ma10']	0.08053
6	['close','macd','ma20','ma5','ma10','roc']	0.10980	['close','macd','ma20','ma5','ma10','roc']	0.08474
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.16819	['close','macd','ma20','trix20','ma5','ma10','roc']	0.12634
	SAE Encoded			
1	['close']	0.10485	['close']	0.08577
2	['close','roc']	0.09551	['close','roc']	0.07620
3	['close','ma5','roc']	0.09714	['close','ma5','roc']	0.07705
4	['close','ma5','ma10','roc']	0.09993	['close','ma5','ma10','roc']	0.07992
5	['close','ma20','ma5','ma10','roc']	0.10353	['close','ma20','ma5','ma10','roc']	0.08179
6	['close','macd','ma20','ma5','ma10','roc']	0.11100	['close','macd','ma20','ma5','ma10','roc']	0.08621
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.16671	['close','macd','ma20','trix20','ma5','ma10','roc']	0.12684
	Single AE Encoded			
1	['close']	0.09732	['close']	0.07883
2	['close','roc']	0.09153	['close','roc']	0.07174
3	['close','ma5','roc']	0.09440	['close','ma5','roc']	0.07388
4	['close','ma5','ma10','roc']	0.09536	['close','ma5','ma10','roc']	0.07512
5	['close','macd','ma20','ma5','ma10']	0.10054	['close','macd','ma20','ma10','roc']	0.07870
6	['close','macd','ma20','ma5','ma10','roc']	0.10515	['close','macd','ma20','ma5','ma10','roc']	0.08075
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.16606	['close','macd','ma20','trix20','ma5','ma10','roc']	0.12696
	No AE			
1	**['close']**	**0.09060**	**['close']**	**0.07097**
2	['close','roc']	0.09211	['close','roc']	0.07273
3	['close','ma5','roc']	0.09428	['close','ma5','roc']	0.07447
4	['close','ma5','ma10','roc']	0.09529	['close','ma5','ma10','roc']	0.07501
5	['close','ma20','ma5','ma10','roc']	0.09921	['close','ma20','ma5','ma10','roc']	0.07785
6	['close','macd','ma20','ma5','ma10','roc']	0.11332	['close','macd','ma20','ma5','ma10','roc']	0.08724
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.17666	['close','macd','ma20','trix20','ma5','ma10','roc']	0.13505

Table A.7 The best results of second experiment for all autoencoders based on the feature count (for Route 7: Trabzon to Ankara)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
	LSTM AE Encoded			
1	['close']	0.08516	['close']	0.06775
2	**['close','ma5']**	**0.07983**	**['close','ma5']**	**0.06411**
3	['close','ma5','ma10']	0.08238	['close','ma5','ma10']	0.06642
4	['close','ma5','ma10','roc']	0.08255	['close','ma5','ma10','roc']	0.06661
5	['close','ma20','ma5','ma10','roc']	0.08734	['close','ma20','ma5','ma10','roc']	0.06968
6	['close','macd','ma20','ma5','ma10','roc']	0.10393	['close','macd','ma20','ma5','ma10','roc']	0.08233
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.12915	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10553
	SAE Encoded			
1	['close']	0.09537	['close']	0.07551
2	['close','roc']	0.09305	['close','ma20']	0.07280
3	['close','trix20','ma5']	0.09111	['close','trix20','ma5']	0.06919
4	['close','ma20','trix20','roc']	0.09189	['close','ma20','trix20','roc']	0.07136
5	['close','trix20','ma5','ma10','roc']	0.09374	['close','macd','ma20','trix20','roc']	0.07229
6	['close','ma20','trix20','ma5','ma10','roc']	0.09529	['close','macd','ma20','trix20','ma10','roc']	0.07212
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10463	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08299
	Single AE Encoded			
1	['close']	0.08686	['close']	0.06937
2	['close','trix20']	0.08445	['close','ma10']	0.06741
3	['close','trix20','ma5']	0.08380	['close','trix20','ma5']	0.06735
4	['close','ma5','ma10','roc']	0.08376	['close','trix20','ma5','roc']	0.06866
5	['close','trix20','ma5','ma10','roc']	0.09052	['close','trix20','ma5','ma10','roc']	0.07274
6	['close','macd','ma20','ma5','ma10','roc']	0.10432	['close','macd','ma20','ma5','ma10','roc']	0.08594
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.12180	['close','macd','ma20','trix20','ma5','ma10','roc']	0.09946
	No AE			
1	['close']	0.08827	['close']	0.07060
2	['close','trix20']	0.08523	['close','trix20']	0.06837
3	['close','trix20','roc']	0.08447	['close','trix20','roc']	0.06800
4	['close','trix20','ma5','roc']	0.08575	['close','ma20','trix20','roc']	0.06925
5	['close','ma20','trix20','ma10','roc']	0.09047	['close','ma20','trix20','ma10','roc']	0.07299
6	['close','macd','ma20','ma5','ma10','roc']	0.10317	['close','macd','ma20','ma5','ma10','roc']	0.08202
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.11680	['close','macd','ma20','trix20','ma5','ma10','roc']	0.09517

Table A.8 The best results of second experiment for all autoencoders based on the feature count (for Route 8: İzmir to Ankara)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
LSTM AE Encoded				
1	['close']	0.08993	['close']	0.06828
2	['close','roc']	0.08804	['close','ma5']	0.06627
3	['close','macd','ma5']	0.08604	['close','macd','ma5']	0.06019
4	**['close','macd','ma20','roc']**	**0.08389**	**['close','macd','ma20','roc']**	**0.05980**
5	['close','macd','ma5','ma10','roc']	0.08547	['close','macd','ma5','ma10','roc']	0.06167
6	['close','macd','ma20','ma5','ma10','roc']	0.08950	['close','macd','ma20','ma5','ma10','roc']	0.06708
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10688	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08278
SAE Encoded				
1	['close']	0.09411	['close']	0.07334
2	['close','roc']	0.09544	['close','roc']	0.07416
3	['close','ma5','roc']	0.10059	['close','ma5','roc']	0.07922
4	['close','macd','ma5','roc']	0.11021	['close','macd','ma5','roc']	0.08222
5	['close','macd','trix20','ma5','roc']	0.10987	['close','macd','trix20','ma5','roc']	0.08258
6	['close','macd','ma20','trix20','ma5','roc']	0.10503	['close','macd','ma20','trix20','ma5','roc']	0.07913
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.14405	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10935
Single AE Encoded				
1	['close']	0.09444	['close']	0.07351
2	['close','roc']	0.09611	['close','roc']	0.07641
3	['close','macd','ma5']	0.08756	['close','macd','ma5']	0.06388
4	['close','macd','ma20','ma5']	0.08633	['close','macd','ma5','roc']	0.06266
5	['close','macd','ma20','ma5','ma10']	0.08963	['close','macd','ma20','ma5','ma10']	0.06584
6	['close','macd','ma20','trix20','ma5','ma10']	0.09503	['close','macd','ma20','trix20','ma5','ma10']	0.07109
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10115	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07574
No AE				
1	['close']	0.09633	['close']	0.07499
2	['close','ma5']	0.09439	['close','ma5']	0.07317
3	['close','macd','ma5']	0.09368	['close','macd','ma5']	0.07063
4	['close','macd','ma5','roc']	0.10300	['close','macd','ma5','roc']	0.07617
5	['close','macd','ma5','ma10','roc']	0.10741	['close','macd','ma5','ma10','roc']	0.08227
6	['close','macd','trix20','ma5','ma10','roc']	0.12006	['close','macd','trix20','ma5','ma10','roc']	0.09181
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.14320	['close','macd','ma20','trix20','ma5','ma10','roc']	0.11125

Table A.9 The best results of second experiment for all autoencoders based on the feature count (for Route 9: Gaziantep to İzmir)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
	LSTM AE Encoded			
1	['close']	0.07933	['close']	0.06036
2	['close','macd']	0.07909	['close','roc']	0.06151
3	['close','trix20','roc']	0.07849	['close','trix20','roc']	0.06082
4	['close','ma20','trix20','roc']	0.07803	['close','trix20','ma5','roc']	0.05962
5	['close','macd','trix20','ma5','roc']	0.07729	['close','macd','trix20','ma5','roc']	0.05956
6	['close','macd','ma20','trix20','ma5','roc']	0.07791	['close','macd','ma20','trix20','ma5','roc']	0.05908
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08425	['close','macd','ma20','trix20','ma5','ma10','roc']	0.06244
	SAE Encoded			
1	['close']	0.12448	['close']	0.10002
2	['close','roc']	0.07848	['close','roc']	0.05803
3	['close','macd','roc']	0.08318	['close','macd','roc']	0.06058
4	['close','macd','trix20','roc']	0.07752	['close','macd','trix20','roc']	0.05810
5	['close','macd','trix20','ma5','roc']	0.09554	['close','macd','trix20','ma5','roc']	0.07050
6	['close','macd','ma20','trix20','ma5','roc']	0.09179	['close','macd','ma20','trix20','ma5','roc']	0.06758
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.09553	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07001
	Single AE Encoded			
1	['close']	0.07976	['close']	0.05880
2	['close','trix20']	0.07366	['close','trix20']	0.05629
3	['close','trix20','roc']	0.07316	['close','macd','roc']	0.05569
4	['close','macd','trix20','roc']	0.07237	['close','macd','trix20','roc']	0.05557
5	['close','macd','trix20','ma5','roc']	0.07050	**['close','macd','trix20','ma5','roc']**	**0.05427**
6	['close','macd','trix20','ma5','ma10','roc']	0.07132	['close','macd','trix20','ma5','ma10','roc']	0.05473
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.08776	['close','macd','ma20','trix20','ma5','ma10','roc']	0.06584
	No AE			
1	['close']	0.07078	['close']	0.05503
2	**['close','macd']**	**0.06945**	['close','macd']	0.05484
3	['close','macd','ma5']	0.06973	['close','trix20','ma5']	0.05506
4	['close','macd','trix20','ma5']	0.07022	['close','macd','ma5','roc']	0.05503
5	['close','macd','trix20','ma5','roc']	0.07135	['close','macd','ma20','ma5','roc']	0.05602
6	['close','macd','trix20','ma5','ma10','roc']	0.07126	['close','macd','trix20','ma5','ma10','roc']	0.05531
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07413	['close','macd','ma20','trix20','ma5','ma10','roc']	0.05917

Table A.10 The best results of second experiment for all autoencoders based on the feature count (for Route 10: Adana to İzmir)

Feature Count	RMSE		MAE	
	Features	Loss Value	Features	Loss Value
	LSTM AE Encoded			
1	['close']	0.08827	['close']	0.06929
2	['close','roc']	0.08728	['close','ma10']	0.06825
3	['close','ma5','roc']	0.08651	['close','ma5','roc']	0.06809
4	['close','ma5','ma10','roc']	0.08636	['close','ma5','ma10','roc']	0.06794
5	['close','ma20','ma5','ma10','roc']	0.08650	['close','ma20','ma5','ma10','roc']	0.06830
6	['close','macd','ma20','trix20','ma5','ma10']	0.09102	['close','macd','ma20','trix20','ma5','ma10']	0.07083
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10430	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07905
	SAE Encoded			
1	['close']	0.08903	['close']	0.06972
2	['close','ma10']	0.08883	['close','ma10']	0.06960
3	['close','ma5','ma10']	0.08873	['close','macd','ma5']	0.06874
4	['close','ma20','ma5','ma10']	0.08820	['close','macd','ma5','ma10']	0.06851
5	['close','ma20','ma5','ma10','roc']	0.08749	['close','ma20','ma5','ma10','roc']	0.06962
6	['close','ma20','trix20','ma5','ma10','roc']	0.09405	['close','macd','ma20','trix20','ma5','ma10']	0.07067
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10399	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07843
	Single AE Encoded			
1	['close']	0.08581	['close']	0.06741
2	['close','macd']	0.08589	['close','macd']	0.06778
3	['close','macd','ma10']	0.08603	['close','ma20','ma5']	0.06833
4	['close','macd','ma5','ma10']	0.08574	['close','macd','ma5','ma10']	0.06762
5	['close','macd','ma5','ma10','roc']	0.08718	['close','macd','ma20','ma5','ma10']	0.06815
6	['close','ma20','trix20','ma5','ma10','roc']	0.08768	['close','macd','ma20','trix20','ma5','ma10']	0.06800
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.09485	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07207
	No AE			
1	['close']	0.08412	['close']	0.06533
2	['close','roc']	0.08280	['close','macd']	0.06474
3	['close','ma5','ma10']	0.08296	['close','ma20','ma5']	0.06476
4	**['close','macd','ma5','roc']**	**0.08206**	**['close','macd','trix20','ma5']**	**0.06435**
5	['close','macd','trix20','ma5','ma10']	0.08372	['close','macd','trix20','ma5','ma10']	0.06540
6	['close','macd','ma20','trix20','ma5','ma10']	0.08488	['close','macd','ma20','trix20','ma5','ma10']	0.06695
7	['close','macd','ma20','trix20','ma5','ma10','roc']	0.10057	['close','macd','ma20','trix20','ma5','ma10','roc']	0.07585

PUBLICATIONS FROM THE THESIS

Conference Papers

1. E. Eğriboz, and M. S. Aktaş, "Price forecasting with deep learning in business to consumer markets," in International Conference on Computational Science and Its Applications, Springer, 2021, pp. 565–580.