Deep Learning Based Prediction Model for the Next Purchase

Anıl UTKU, Muhammet Ali AKCAYOL
Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey
anilutku@gazi.edu.tr

***Abstract*—Time series represent the consecutive measurements taken at equally spaced time intervals. Time series prediction uses the information in a time series to predict future values. The future value prediction is important for many business and administrative decision makers especially in e-commerce. To promote business, sales prediction and sensing of future consumer behavior can help business decision makers in marketing campaigns, budget and resource planning. In this study, deep learning based a new prediction model has been developed for the time of next purchase in e-commerce. The proposed model has been extensively tested and compared with RF, ARIMA, CNN and MLP using a retail market dataset. The experimental results show that the developed model has been more successful than RF, ARIMA, CNN and MLP to predict the time of the next purchase.**

***Index Terms*—time series analysis, deep learning, prediction, e-commerce.**

I. Introduction

Time series represent a series of consecutive measurements taken at equally spaced time intervals [1]. Measurements in different time steps are related to each other. Since successive values are related to others, future values can be predicted by using prior values. Time series analysis is an important research area used to analyze relationships and dependencies between values [2].

The main purpose of time series analysis is to develop a prediction model using past values [3]. The model is used to predict future values based on previously observed values. The time series analysis is mainly used in business, finance, energy management, telecommunications, bioinformatics, science and engineering [4].

Autoregressive Integrated Moving Average (ARIMA) is one of the most popular probabilistic time series models [5]. The basic assumption in this model is that values have a specific statistical distribution [6]. The ARIMA model has several subclasses such as the autoregressive (AR), the moving average (MA) and the autoregressive moving average (ARMA). Box et al. proposed Seasonal Autoregressive Integrated Moving Average (SARIMA) model for seasonal time series [7]. The ARIMA is widely used because of its simplicity and flexibility to represent different types of time series [8]. However, disadvantage of the ARIMA is the assumption that the time series should be linear [9]. To overcome this disadvantage, various nonlinear models such as artificial neural networks (ANNs) have been proposed [10]. The ANNs can be used to predict the next time of event without any assumptions about the statistical distribution of the values [9].

Chujai et al. have developed a new model to predict electricity consumption in a household daily, weekly, monthly and quarterly [10]. They have used electricity consumption data between December 2006 and November 2010 and analyzed data using ARIMA and ARMA models. The proper prediction model has been selected according to the lowest Akaike Information Criterion (AIC) and Root Mean Square Error (RMSE).

Kane et al. have used ARIMA and random forest (RF) models to predict the future status of disease using the data on avian influenza disease accessed via the EMPRES-i system in Egypt [11]. The experimental results show that RF model is more successful than ARIMA.

Fu et al. have developed a deep learning-based prediction model for vehicle traffic flow [12]. In the developed model, Long-Short Term Memory (LSTM) and Gated Recurrent Units (GRU) have been used to predict the short-term traffic flow with PeMS data which consists of data created by more than 15,000 sensors in California. Experimental results show that deep learning-based LSTM and GRU methods perform better than ARIMA.

Amini et al. have developed a prediction model using the ARIMA to predict the charge demand for electric vehicles [13]. The developed model has used daily driving patterns and distances as inputs to predict charge demand. The experimental results show that the ARIMA-based developed model has successful results.

Yang et al. have developed a new ARIMA-based model to predict change of metal content in aircraft engine oil for safe flight. The developed ARIMA-based model has been compared with moving averages and exponential correction methods. The experimental results show that the ARIMA-based developed model has more successful results than moving average and exponential correction methods.

Zheng et al. have developed a LSTM based model to predict future values for the electric charge in smart grids [15]. In the developed LSTM-based repetitive model has used long-term dependencies in time series to obtain more accurate predictions. They have compared Support Vector Machine (SVM), Autoregressive Neural Network, LSTM, Non-linear Autoregressive exogenous (NARX) and ARIMA. The experimental results show that LSTM model has more successful results than others.

Sobreiro et al. have developed a prediction model for the activities of companies [16] using the exponential correction, ARIMA and ANN. The dataset used in this study has 192 months of activity data of a company. The experimental results show that the ANN based model has higher predictive accuracy than others.

McNally et al. have presented a comparative analysis of

Digital Object Identifier 10.4316/AECE.2020.02005

1582-7445 © 2020 AECE

ARIMA, Recurrent Neural Network (RNN) and LSTM models for predicting bitcoin price [17]. Bitcoin price indices have been used as dataset in the developed model. The comparison results have shown that non-linear deep learning methods, especially LSTM, has higher prediction accuracy than the ARIMA.

Wang et al. have presented an application of the ARIMA to predict the behavior of smart home users [18]. In the developed model, the data of on/off time of the water heater for 12 weeks have been used to predict the next on/off time of the heater. The experimental results show that the developed new ARIMA model has better prediction performance than the classical ARIMA model.

Fu et al. have developed LSTM and GRU based model to predict short-term traffic flow using PeMS dataset [19]. PeMS dataset containing data from more than 15,000 sensors in California has been used as dataset. A comparative experimental study was conducted using ARIMA, LSTM and GRU models. Experimental results show that deep learning LSTM and GRU models give more successful results than ARIMA.

Chen et al. have developed an LSTM-based model for predicting Chinese stock returns [20]. Historical data on the Chinese stock exchange has been used as dataset. The developed model has been trained over 900000 sequences and tested using 311361 sequences. As a result of comparative experimental studies conducted with random prediction method, it has been seen that the LSTM-based model increased the stock return estimate accuracy from 14.3% to 27.2%.

Sarma have presented a study on feature extraction, which is an important component in the neural network-based pattern recognition problem [21]. In the study, a neural feature extractor is presented to be used for character and numerical recognition in the Assamese language. The results have been compared with similar studies and the results obtained from a PCA-based feature extraction method. Experimental results show that the developed method gives more successful results.

Pozna et al. have proposed a unique heuristic modeling algorithm, expressed as Bayesian accuracy [22]. The main benefits of the proposed approach compared to conventional modeling approaches are increased transparency and reduced computation time. Two sample scenarios for a mobile robot and an unforced pen dulum system have been used to sample and test the results.

Gil et al. have developed a fuzzy model for determining travel and delay times on a road [23]. It is aimed to determine different traffic flows and traffic light cycles using a microscopic traffic simulator. Under different traffic loads, optimal traffic light times are determined. In addition, a method is provided to obtain the optimum green time of a traffic light as well as the cycle time at the intersection.

Albu et al. have developed LSTM-based artificial neural network model that can help make medical decisions [24]. With the developed model, experimental studies have been carried out for the detection and diagnosis of skin diseases, hepatitis B, stroke and hepatitis C diseases.

Time series analyses and the future value prediction have been used in many kinds of areas such as electricity consumption prediction, wind speed prediction, traffic load prediction, stock market prediction, users' behavior prediction, and prediction of the next crime time [9-24]. In the latest studies, it has been seen that machine learning and deep learning models have been used apart from statistical methods such as ARMA and ARIMA.

The prediction of next time of an event is critical in some of the application areas, such as Web-based attacks, forensic applications, users' behavior analysis for e-commerce. The risks that may occur can be reduced by predicting the time of occurrence before the events occur. On the other hand, in terms of e-commerce applications, the benefit can be maximized by providing users with timely and personalized recommendations.

In this study, LSTM-based deep learning model has been developed for predicting the next time of the events. With the event time prediction, the damage caused by the events can be minimized and the the benefit to be obtained can be maximized. Although, time series analysis has used for many kinds of areas such as users' behavior analysis, demand prediction, Web based attack prediction, however there is no study in the literature for event time prediction.

In this study, an LSTM-based deep learning model has been developed to improve predictive accuracy and effectiveness of the next purchase time in e-commerce systems. Comparative experimental studies have been conducted for the proposed LSTM based model with RF, ARIMA, Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP). The experimental results show that the developed LSTM based model has more successful than others in predicting the time of next purchase.

Time series analysis and the prediction of the next time have been presented in Section II. Machine learning methods, linear methods and deep learning-based models have been explained in detail for the prediction of the next time in the time series data.

II. Prediction of The Next Time

Time series analysis determine the corresponding new values of the time series from past values using a prediction model [25]. Time series analysis methods try to understand the nature of the data. In the time series analysis, it is aimed to develop an appropriate mathematical model by using past observations to predict the future values [5].

Time series are series of data points measured at consecutive time intervals. Mathematically, $\{y_t, t = 0,1,2,\ldots,T\}$ is defined as a set of T vectors, where t is the elapsed time. A time series is generated by y_t values representing the value at time t. Therefore, the prediction process aims to determine the $y_t+1-(\hat{y}_t+1)$. value and to minimize the error represented as $y_t+1-(\hat{y}_t+1)$. The y_t is considered a random variable. In time series, measurements taken during an event are listed in accordance with an application specific order [1].

The general strategy used to analyse time series is to differentiate time series into three main components: tendency, seasonality, and irregular components [3]. The trend refers to the tendency of the data during the observation period without considering the seasonality and irregularities. Trends can be linear, exponential or parabolic. Seasonality refers to the periodic fluctuations of the values. Seasonality can be caused by various factors, such as weather, vacation, economics, and holidays [26]. Irregular components refer to residual values, and cannot be affected by trend and seasonality. These values may mask the tendency and seasonality of the data. The time series in real world applications includes many kinds of irregular components and these are the most challenging issue for the construction of a model. Most of the techniques used in time series analysis are focused on determining tendency and seasonality [1].

Statistical models such as ARMA and ARIMA are widely used in time series prediction. However, ARIMA, which is a statistical model widely used in time series prediction, cannot define the stochastic and nonlinear structure in the time series. For this reason, machine learning and deep learning methods are used to modelling the relationships in time series. In the literature, apart from classical machine learning approaches such as RF and Support Vector Regression (SVR), deep learning models such as MLP, CNN, GRU and LSTM are used in time series prediction.

A. Machine Learning Methods for Prediction of the Next Time

Predicting future values from past values requires the use of techniques that can extract dependencies between past and future. Time series data can be transformed into a supervised learning problem using the window method as described in the Chapter III. In this way, the future values of the data can be predicted from the past values by using machine learning methods.

RF is a commonly used machine learning algorithm that uses the bagging technique. Bagging technique refers to combining the results of multiple models to achieve better performance. Predictions can be obtained by using random sampling of training data. The main predictors in RF are decision trees. RF adds randomness to the model. Instead of searching for the most important property, it looks for the best property in a random subset of properties. This provides a wide variety and often better results.

B. Linear Methods for Prediction of the Next Time

Because the time series is not inherently deterministic, it is unclear what the future will be. In general, the time series $\{y_t,\ t=\ 0,1,\ 2,\ ...,T\}$ is assumed to follow the combined probability distribution of a random y_t [3]. The mathematical expression defining the probability of a time series is called a probabilistic process. Thus, the sequence of observation of the series is in fact a sample of the probabilistic process [5]. The general assumption in the time series is that the y_t values in the series are distributed independently. However, in the long run, time series follow a regular pattern. For example, if the current temperature of a particular city is very high, it can be estimated that the temperature of tomorrow is probably high. This is related to the concept of stationarity in time series [4].

The concept of stationarity in a probable process can be seen as a statistical form of equilibrium. The statistical properties of a static process, such as the average and standard deviation are not time-dependent. These statistical features are a necessary condition to create a time series model that will be used to make predictions for the future values [1]. Selecting the appropriate model in time series directly affects the accuracy of the predictions. Generally, models to be used in time series may include different probabilistic processes [27]. AR and MA commonly use linear time series. ARMA and ARIMA are used by combining these two models [6].

The ARMA (p, q) model used for the modelling of univariate time series is a combination of AR (p) and MA (q) models [28]. In the AR (p) model, it is assumed that the future value of a variable is a linear combination of past observations and has a fixed error term. Mathematically, the AR (p) model is expressed by using Equation (1) [1].

$$y_t = c + \sum_{i=1}^{p} \varphi_t y_{t-i} + \varepsilon_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \ldots + \varphi_p y_{t-p} + \varepsilon_t \tag{1}$$

where, y_t is the real value and ε_t is the random error in time t. $\varphi_i\,(i=1,2,..,p)$ refers to parameters and c is a constant value. The integer constant p is known as the order of the model. The MA (q) model uses past errors as explanatory variables, as seen in Equation (2) [3].

$$y_t = \mu + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{2}$$

where, μ is the mean of time series, $\theta_j\,(i=1,2,..,j)$ is the model parameter. The random error is assumed to be a randomly distributed. The AR and MA models can be combined to create a general and useful time series model known as the ARMA model. Mathematically, an ARMA (p, q) model is expressed using Equation (3) [7].

$$y_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \tag{3}$$

where, p represents autoregressivity and q represents the moving average.

The ARIMA is a model that integrates autoregressive AR (p) and moving average MA (q) and creates a combined model for time series. ARIMA has three parameters as p, d and q. The autoregressive AR (p) in the ARIMA is a regression model using dependencies between observations [29]. Integrated model measures differences of observation at different times (d). The moving average MA (q) in the ARIMA is an approach that considers the dependencies between the observations [30].

The ARIMA follows a methodology introduced by Box and Jenkins in 1976 [31]. the ARIMA consists of defining the model, estimating the parameters, verifying the model. The defining of the model is to determine the changes of the mean and standard deviation of the time series [30]. Parameter estimation refers to the estimation of p and q parameters using least squares minimization or maximum probability algorithms. The least squares minimization minimizes the squared error in the estimation of the training set, while the maximum probability algorithm aims to maximize the similarity function [1]. Model validation refers to determining whether the model is appropriate or not for the problem. If the data is completely random, the model is approved by assuming that it has white noise [2]. Metrics such as AIC, MAE or RMSE can be used to evaluate the performance of models. Future predictions with ARIMA are calculated by using Equation (4).

$$\hat{y}_t = \mu + \varphi_1 y_{t-1} + ... + \varphi_p y_{t-p} - \theta_1 e_{t-1} - ... - \theta_q e_{t-q} \quad (4)$$

Where $\hat{y}_t$ is the predicted value, e_t is the error at time t and φ_i, θ_j are coefficients, μ stands for white noise. AR part of ARIMA shows that y_t is obtained from previous values and MA part shows that regression error is a linear combination of previous errors. I part indicates that values are changed by the difference between previous values [3].

C. Deep Learning Based Models for Prediction of the Next Time

Time series can be obtained by using many real-world applications, such as monitoring price changes in the stock market, spreading processes of diseases, and monitoring of changes in audio signals [32]. While physicians are interested in identifying abnormalities in the sleep patterns of patient, economists are interested in determining the future value of the interest rate. This kind of problems are examined as classification, estimation or anomaly detection problems [33].

ANN is a computational model based on the structure and function of biological neural networks [34]. ANNs are considered as non-linear statistical data modelling tools in which complex relationships between inputs and outputs are modelled. Today, ANN is used in various areas such as robotics, object recognition, speech and handwriting recognition.

An ANN consists of at least three layers, an input layer, hidden layers, and an output layer. The number of features in the dataset determines the size of the input layer. Input layer nodes are connected to hidden layer nodes by structures called synopsis [33]. Synopsis connections have some weights for each node output in the input layer. The weights play important role in deciding which signal or input can pass. The weights indicate the strength of the hidden layer.

In hidden layers, nodes implement an activation function to convert inputs to outputs. Using a function called SoftMax, the output layer tries to minimize the difference between expected and estimated values. To find the most appropriate weight values, the output error is propagated from the output layer to the input layer. The training process are repeated and the weights are adjusted until predefined success rate obtained [35].

Deep neural networks are defined as artificial neural networks with more than one hidden layer between the input and output layers [36]. Deep neural networks have a more complex structure and computational complexity. Deep neural networks have more interconnected neurons than classical feed forward neural networks [37]. The deep label in deep learning means multiple layers and huge number of neurons [38]. In this study, CNN, MLP, RNN and LSTM models have been examined comparatively.

1) CNN

CNN is a deep neural network that has gained popularity thanks to its success in application areas such as image recognition and classification. CNN is successfully used for applications such as computer vision, image classification, face recognition, object identification and classification, and image processing in autonomous vehicles [39].

In general, CNN has 3 basic steps. In a simple classification example, a feature map is created with convolution in which several pixels of the image are scanned. Secondly, pooling (sub-sampling) is performed, where the size of each feature is reduced. The pooling phase creates a summary of the most important features in the image [40]. For example, in the first stage, the image is decomposed into objects such as people, ships or trees. In the second stage, CNN defines features such as hand, face, arm, leg inside each object. In the third stage, CNN can analyse facial features with a deep analysis.

2) MLP

MLP is the result of work done to solve the XOR problem. For many inputs, one neuron may not be enough. The layer concept comes into play when multiple parallel neurons are needed. The input layer receives the incoming data and sends it to the intermediate layer [41]. The incoming information is transferred to the next layer. The number of interlayer varies depending on the problem, at least one and adjusted according to the need. The output of each layer becomes the entrance of the next layer. Thus, the exit is reached. Each processing element, the neuron, is connected to all neurons in the next layer. The number of neurons in the layer is determined by the problem. The output layer determines the output of the network by processing data from previous layers. The output number of the system is equal to the number of elements in the output layer.

3) RNN

The RNN aims to predict the next step according to the previous steps observed in the sequence. The idea behind the RNNs is to learn from the previous values to predict future values [42].

The hidden layers in the RNN serve as an internal storage unit for storing information obtained in the previous stages of sequential data. The RNNs have a repetitive structure because they perform the same task for each element of the sequence and use the old information to predict unseen data [33]. The biggest challenge of the RNN is that the recently processed data in the series is remembered and therefore it is not suitable for remembering longer data sequences. This problem is solved in LSTM using the memory unit [7].

4) LSTM

In the event that the ANN structure is too complex, the weight values are not updated and the training of the network is terminated [43]. This problem makes it difficult to learn and adjust the parameters of the previous layers in the network [44]. The LSTM is a neural network architecture that has been developed as a solution to backward dependence problems in repetitive neural network structures [43].

The LSTM eliminates the vanishing gradient problem that has been trained using backward propagation over time [45]. The LSTM have memory blocks attached to layers instead of neurons. The components of memory blocks contain doors that manage the status and output of the blocks. In LSTM, blocks are more intelligent than a conventional neuron. A block operates on an input array and each door in the block uses sigmoid activation units to control whether they are triggered. Sigmoid activation units also allow the change of state and the addition of information flowing through the block condition [33].

The LSTM are a set of cells in which data streams are obtained and stored. The cells resemble a transmission line that transmits and collects data from one module to another. Due to the use of some doors in each cell, the data in the cells can be disposed, filtered or added to the next cells. Therefore, doors based on the sigmoidal neural network layer allow data to pass or block depending on the request of the cells [46]. Each sigmoid layer outputs zero and one, indicating the amount of data portion allowed in each cell. Zero value means that nothing is allowed to pass, and one value is allowed to pass through everything [7].

There are three types of doors in the sigmoid activation units for controlling the status of each cell. The entry door conditionally decides which values in the entry update the memory status. The exit door decides conditionally on what to output depending on the input and the memory of the block. The remembering door decides which information is to be removed from the block. Here, each unit is like a state machine where the unit doors have weights learned during the training procedure [44].

D. Accuracy Metrics for Prediction of the Next Time

The purpose of accuracy and error metrics is to obtain a summary of the error distribution. After calculating the loss function, it is common practice to determine the error measurements by calculating the mean. The t_x is the elapsed time between purchase event x and previous purchase event from event x. $\hat{t}_x$ is the predicted elapsed time. The error value is calculated by $t_x - \hat{t}_x$ [3].

Scale-dependent metrics are useful when comparing different methods on the same dataset [43]. Commonly used criteria are MAE and MSE metrics. The MAE metric is the mean of the errors as seen in Equation (5).

$$MAE = \frac{1}{n}\sum_{x=1}^{n} | t_x - \hat{t}_x | \tag{5}$$

MSE metric has been calculated by mean of the squares of errors as seen in Equation (6), and the RMSE has been calculated by the square root of the mean of the squares of error as seen in Equation (7) [47].

$$MSE = \frac{1}{n}\sum_{x=1}^{n} (t_x - \hat{t}_x)^2 \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{x=1}^{n} (t_x - \hat{t}_x)^2} \tag{7}$$

MAE is widely used as it is easy to calculate, MAE does not penalize over-estimation errors, MSE and RMSE emphasize that total estimation error. MSE and RMSE are more sensitive to outliers than MAE [7].

The developed deep learning model for the prediction of the time of the next purchase has been explained in detail in Section III. The data pre-processing process, converting time series data to be able to use for supervised learning problem, splitting data into as train, test and validation datasets have been explained.

III. The Developed Deep Learning Model for Prediction of the Time of Next Purchase

Probability based ARIMA and machine learning based methods are widely used in time series analysis. Although ARIMA are widely used in modelling of economic and financial data, there are some important limitations. For example, in the ARIMA, it is difficult to model nonlinear relationships between variables. In addition, the ARIMA assumes that errors have a fixed standard deviation. In order to overcome the difficulties related to linear models, new methods have been developed based on deep learning.

Event time prediction is an interesting problem for many application areas. In this study, a model has been developed for predicting purchase time.

Anonymous data consisting of user interactions in an e-commerce system have been used in the developed model. The retail rocket dataset is publicly available in Kaggle via https://www.kaggle.com/retailrocket/ecommerce-dataset [48].

There are three different user activity in the dataset: (1) movie viewing, (2) add to basket and (3) purchase. Dataset consists of 2756101 users' behaviors. The behavior data have been collected over a period of 4.5 months. There are total of 2756101 events produced by 1407580 individual visitors including 2664126 movie viewing, 69332 add to basket and 22457 purchase. After the data pre-processing, 17939 purchase events have been obtained. Table I shows the first five rows of the dataset.

TABLE I. THE DATASET USED IN THIS STUDY

Timestamp	Visitor ID	Event	Movie ID	Purchase ID
1433221332117	257597	Viewing	355908	-
1433224214164	992329	Viewing	248676	-
1433221999827	111016	Viewing	318965	-
1433221955914	483717	Viewing	253185	-
1433221337106	951259	Viewing	367447	-

The dataset consists of time information, user and movie ID, event and purchase ID. The time information in the dataset has been recorded according to the date format in Unix systems. Unix time information is a time type that indicates the number of seconds since January 1, 1970. The time information in the dataset has been changed from Unix time information to day/month/year, hour/minute/second time format for this study. User and movie ID fields refer to the ID numbers of users and movies in the system. The event field refers to the type of interaction performed. Purchase ID field refers to the number assigned to purchase interactions. For example, “1433221332117, 257597, view, 355908, -” line indicates that the Visitor ID 257597 views the Movie ID 355908 at time 1433221332117 (02/06/2015, 08:02:12).

The input, output and structure of the developed model are shown in Fig. 1.

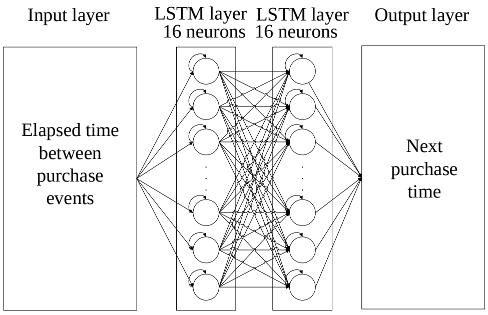

Figure 1. The network architecture of the developed model

The developed model has 2 LSTM layers as seen in Fig. 1. There are 16 LSTM neurons in each layer. The input of the developed model is the elapsed time between purchasing events. Two LSTM layers with 16 neurons are used for learning and prediction. The output of the model is the time that will pass until the next purchase time.

To predict the time of next purchase, the elapsed time between purchases has been calculated in seconds. For example, there are 460 seconds between May 3, 2015 06:27:21 and May 3, 2015 06:35:01.

Time series refers to the sequence of numbers ordered by a time index. In supervised learning problems, it is aimed to estimate the output from the input, such as $y = f(x)$, where x is input, y is output. Time series data have been converted to use supervised learning models. The time series data is shown in Fig. 2 have been configured as a supervised learning problem using the values from the previous time step to estimate the value in the next time step.

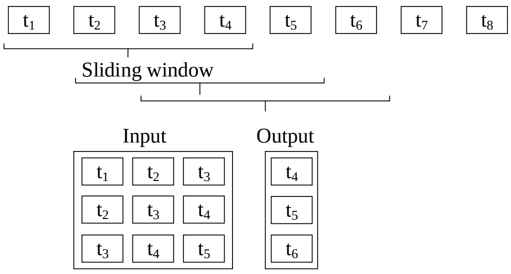

Figure 2. Converting time series data to supervised learning problem

As shown in Fig. 2, the input (x) in the previous time step is the output (y) of the next time step. Sliding window method has been used to estimate the next time step. The number of previous time steps determines the size of the sliding window. In this work, the sliding window size is defined as 3.

In developed system, data pre-processing has been carried out firstly. Missing and incorrect data in the dataset have been checked and cleaned. Due to its importance in e-commerce systems, the purchasing behavior has been chosen as an event. After the event selection, the elapsed times in seconds between the purchase events have been calculated. A sliding window has been used to convert the time series data into supervised learning data. In this way, data at time t_1, t_2 and t_3 are configured to be input, and data at time t_4 to be output.

Then, the elapsed time between events has been normalized. Normalization is a technique that often used as part of data preparation for machine learning. The purpose of normalization is to change the values of numerical columns in a dataset to a common scale without disrupting the differences in the range of values. Normalization affects the performance of the developed model and the stability of training. Data have been rescaled in the range 0-1. For normalization, the maximum and minimum values in the data have been determined. Using the equation $y = (x - \min) / (\mathrm{maks} - \min)$, y, the normalized value for x, has been obtained. After the normalization step, the data have been split into 67% training and 33% testing. 10% of the training data have been split for validation. Validation data have been used for optimization of model parameters. Walk forward validation is shown in Fig. 3. With parameter optimization, the parameters with the lowest MSE value have been determined and the model has been created.

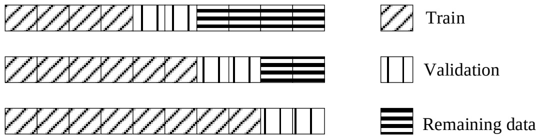

Figure 3. Walk forward validation

Walk forward validation is an adapted version of the cross-validation method for time series data. Cross validation is a technique used to set parameters and measure model performance. In cross validation, the model is trained with the training set and parameters that minimize the error are selected in the validation set. Finally, the model is trained on the full training set using the selected parameters and the error in the test set is recorded. Cross-validation technique cannot be used in time series data due to time dependency. In walk forward validation, first, the window size has been defined. After predicting the value in the window, the values in the window are added to the training data. This process has been repeated for all data in the validation set.

The elapsed time between the occurrences of the purchase events has been calculated in order to determine the time of next purchase. The top 5 lines of data generated according to the elapsed time between the purchases are shown in Table II.

TABLE II. ELAPSED TIME BETWEEN PURCHASE EVENTS

Events	Elapsed time (sec)
1	460
2	1606
3	351
4	1416
5	30

The elapsed time in seconds between purchases are shown in Fig. 4.

The dataset has been split into 67% for training and 33% for testing. 10 percent of the training dataset has been used for validation. Parameter optimization has been performed by using walk forward validation on the validation data.

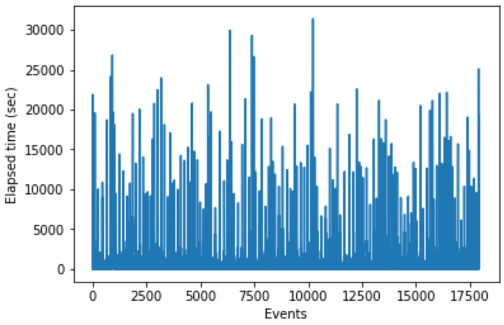

Figure 4. Elapsed time between purchases

In the developed system, first, pre-processing has been carried out on the data. Missing and outlier data in the dataset have been checked and cleaned. Due to its importance in e-commerce systems, we focused on the purchasing activity in this study. Then, the elapsed time between the purchase events have been calculated in seconds. Sliding window method has been used to convert the time series data to input data for the developed LSTM based model. After, the elapsed time between events has been normalized, the dataset has been split into 67% for training and 33% for testing.

In order to determine the parameters of the RF model, an analysis study has been performed using MSE metric as shown in Table III.

TABLE III. RF PARAMETER SELECTION

Number of trees	**Minimum number of samples required to split**	**MSE**
100	2	1654738,83
300	2	1601182,39
500	**2**	***1598352,08**
1000	2	1599545,16
2000	2	1598541,92

*Selected parameters

The results of the parameter analysis for RF are shown in Table III. The lowest MSE value has been obtained when the number of trees is 500. These parameters have been selected for experimental studies. Fig. 5 shows the results of RF for the test data.

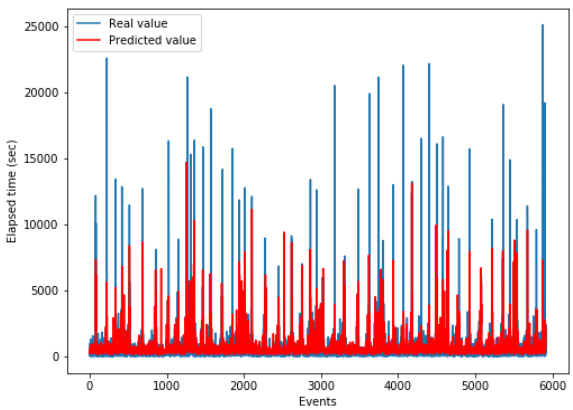

Figure 5. The results of RF

An analysis study has been performed using the MSE metric as shown in Fig. 6 for the determination of the parameters *p*, *d* and *q* for the ARIMA.

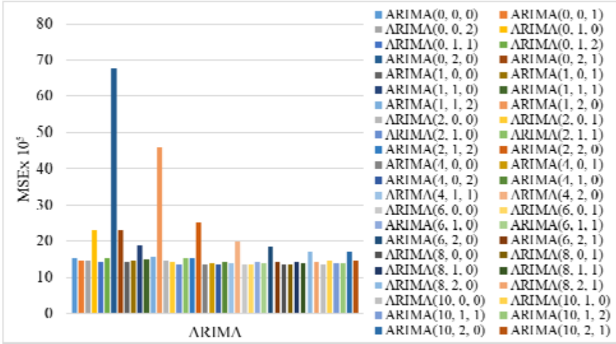

Figure 6. The parameter selection for ARIMA

ARIMA (6,0,0) model has been chosen to be used in experimental studies because it has the lowest MSE value. Fig. 7 shows the results of ARIMA for the test data.

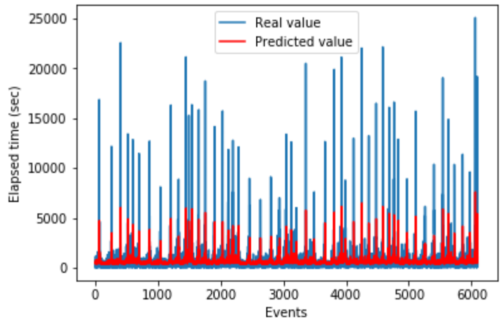

Figure 7. The results of ARIMA

In order to determine the parameters of the CNN, an analysis study has been performed using MSE metric as shown in Table IV.

TABLE IV. CNN PARAMETER SELECTION

Filter size	**Epoch**	**Kernel size**	**MSE**
32	**200**	**1**	***1410249,29**
64	100	2	1596508,06
64	200	2	1509089,40
64	300	2	1511621,07
16	200	2	1534501,56
32	200	2	1536087,57
32	200	3	1430266,59
64	200	3	1442160,81

*Selected parameters

The results of the parameter analysis for CNN are shown in Table IV.

Analysis results showed that CNN has the lowest MSE value with 32 filter number, 200 epochs and 1 kernel size. These values are selected for use in experimental studies. Fig. 8 shows the results of CNN for the test data.

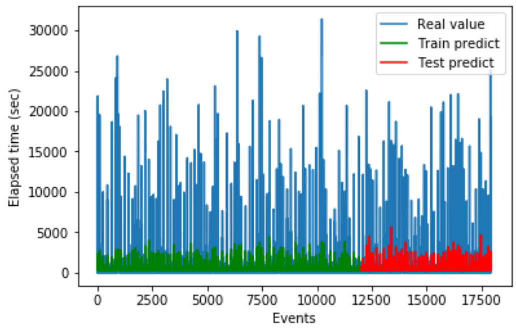

Figure 8. The results of CNN

In order to determine the parameters of the MLP, an analysis study has been performed using MSE metric as shown in Table V.

TABLE V. MLP PARAMETER SELECTION

Neurons	Epoch	MSE
4	50	1556962.81
8	50	1561413,90
12	50	1558471.27
16	50	1561096.64
12	**100**	***1556961.85**
12	200	1576933.04

*Selected parameters

The results of the parameter analysis for MLP are shown in Table V.

Analysis results showed that MLP has the lowest MSE value with 12 neurons and 100 epochs. These values are selected for use in experimental studies. Fig. 9 shows the results of MLP for the test data.

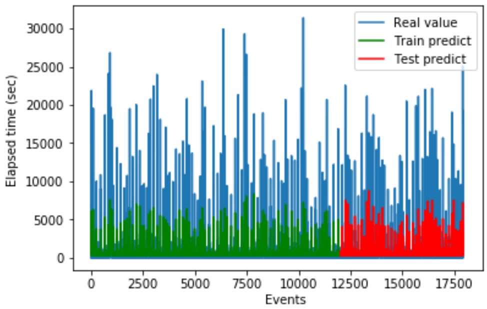

Figure 9. The results of MLP

In order to determine the parameters of the developed model, such as the batch size, number of epochs and the number of neurons in the hidden layer, an analysis study has been performed using the MSE metric.

The results of the optimization of parameters for the developed model have been shown in Table VI.

TABLE VI. FIRST PHASE FOR THE PARAMETER SELECTION

Batch size	Epoch	Number of neurons	MSE
1	100	4	1426055.16
16	100	4	1442976.84
32	100	4	1437500.39
64	100	4	1414555.30
256	100	4	1392441.75
512	**100**	**4**	***1398286.37**
1024	100	4	1418868.37

* Selected parameters

At the second phase, the studies for the selection of parameters have been focused on batch size 512. The different number of neurons and epochs have been tried and the best values have been determined. The second phase of the selection of parameters has been shown in Table VII.

TABLE VII. THE SECOND PHASE FOR PARAMETER SELECTION

Batch size	Epoch	Number of neurons	MSE
512	100	8	1392404.08
512	**100**	**16**	***1387064.60**
512	100	32	1394094.64
512	200	32	1388986.57

* Selected parameters

The experimental results show that the developed model has the lowest MSE, the batch size 512, the number of epochs 100, and the number of neurons in the hidden layer 16.

The experimental results of the developed model are shown in Fig. 10.

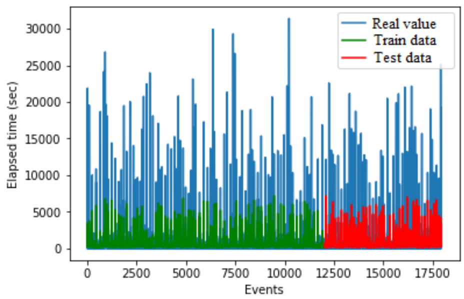

Figure 10. The results for the train and the test data

RF, ARIMA, CNN, MLP and developed model have been run 10 times and the results have been analysed comparatively. The experimental results according to MSE, RMSE and MAE are shown in Table VIII, Table IX, Table X and Fig. 11.

TABLE VIII. EXPERIMENTAL RESULTS ACCORDING TO MSE

Test	RF	ARIMA	CNN	MLP	Developed model
1	2437427.19	2383163.05	2338472.25	2317023.33	**2064187.40**
2	2437427.19	2383163.05	2342609.34	2322314.46	2064346.14
3	2437427.19	2383163.05	2355577.03	2323738.82	2064349.11
4	2437427.19	2383163.05	2355610.90	2324489.55	2064360.08
5	2437427.19	2383163.05	2360438.57	2325388.83	2064369.32
6	2437427.19	2383163.05	2363212.32	2325485.06	2064449.69
7	2437427.19	2383163.05	2363929.76	2325693.99	2064472.17
8	2437427.19	2383163.05	2368117.63	2325759.97	2064479.26
9	2437427.19	2383163.05	2381283.45	2327537.06	2064528.98
10	2437427.19	2383163.05	2400102.93	2328349.80	**2064610.41**
Average	**2437427.19**	**2383163.05**	**2362935.41**	**2324578.08**	**2064415.25**

Table VIII shows that the best MSE value of developed model is 2064187.40 and the worst MSE value is 2064610.41. As RF and ARIMA are deterministic models, the MSE values of RF is 2437427.19 and the MSE values of ARIMA is 2383163.05. The average MSE value of CNN is 2362935.41 and the average MSE value of MLP is 2324578.08. The average MSE value of developed model is 2064415.25.

The equations shown below have been used to determine the improvement rates of the developed model according to RF (Equation (8)), ARIMA (Equation (9)), CNN (Equation (10)) and MLP (Equation (11)).

$$\frac{\text{RF-Developed model}}{\text{RF}} \times 100 \tag{8}$$

$$\frac{\text{ARIMA-Developed model}}{\text{ARIMA}}\times 100 \quad (9)$$

$$\frac{\text{CNN-Developed model}}{\text{CNN}}\times 100 \quad (10)$$

$$\frac{\text{MLP-Developed model}}{\text{MLP}}\times 100 \quad (11)$$

The developed model improved the average MSE rate by 15.30% according to RF, 13.37% according to ARIMA, 12.63% according to CNN and 11.19% according to MLP.

TABLE IX. EXPERIMENTAL RESULTS ACCORDING TO RMSE

Test	RF	ARIMA	CNN	MLP	Developed model
1	1561.22	1543.74	1529.20	1522.18	**1436.72**
2	1561.22	1543.74	1530.55	1523.91	1436.78
3	1561.22	1543.74	1534.78	1524.38	1436.78
4	1561.22	1543.74	1534.79	1524.63	1436.79
5	1561.22	1543.74	1536.37	1524.92	1436.79
6	1561.22	1543.74	1537.27	1524.95	1436.82
7	1561.22	1543.74	1537.50	1525.02	1436.83
8	1561.22	1543.74	1538.86	1525.04	1436.83
9	1561.22	1543.74	1543.14	1525.63	1436.85
10	1561.22	1543.74	1549.22	1525.89	**1436.88**
Average	**1561.22**	**1543.74**	**1537.16**	**1524.65**	**1436.80**

Table IX shows that the best RMSE value of developed model is 1436.72 and the worst RMSE value is 1436.88. As RF and ARIMA are deterministic models, the RMSE value of RF is 1561.22 and the RMSE value of ARIMA is 1543.74. The average RMSE value of CNN is 1537.16 and the average RMSE value of MLP is 1524.65. The average RMSE value of developed model is 1436.80.

Equations (8-11) have been used to determine the improvement rates of the developed model according to RF, ARIMA, CNN and MLP.

The developed model improved the average RMSE rate by 7.96% according to RF, 6.92% according to ARIMA, 6.52% according to CNN and 5.76% according to MLP.

TABLE X. EXPERIMENTAL RESULTS ACCORDING TO MAE

Test	RF	ARIMA	CNN	MLP	Developed model
1	678.55	672.19	644.32	662.01	**629.05**
2	678.55	672.19	663.29	662.46	659.95
3	678.55	672.19	666.23	662.57	659.95
4	678.55	672.19	666.53	663.07	659.96
5	678.55	672.19	668.90	664.02	659.97
6	678.55	672.19	669.09	665.55	659.97
7	678.55	672.19	671.14	666.10	659.99
8	678.55	672.19	674.94	666.20	659.99
9	678.55	672.19	675.75	667.35	659.99
10	678.55	672.19	676.69	668.46	**659.99**
Average	**678.55**	**672.19**	**667.68**	**664.77**	**656.88**

Table X shows that the best MAE value of developed model is 629.05 and the worst MAE value is 659.99. As RF and ARIMA are deterministic models, the MAE value of RF is 678.55 and the MAE value of ARIMA is 672.19. The average MAE value of CNN is 667.68 and the average MAE value of MLP is 664.77. The average MAE value of developed model is 656.88.

Equations (8-11) have been used to determine the improvement rates of the developed model according to RF, ARIMA, CNN and MLP.

The developed model improved the average MAE rate by 3.19% according to RF, 2.27% according to ARIMA, 1.61% according to CNN and 1.18% according to MLP.

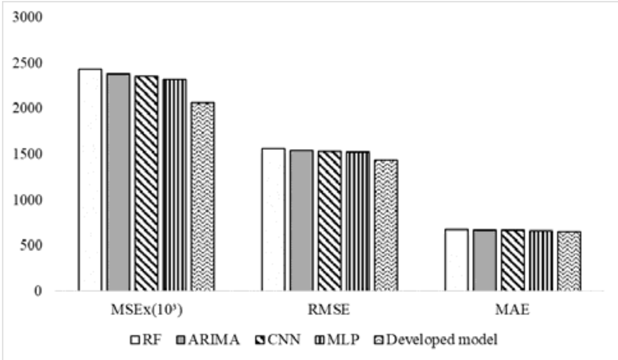

Figure 11. Experimental results according to MSE, RMSE and MAE

As shown in Table VIII, Table IX, Table X and Fig. 11, the developed model is more successful than RF, ARIMA, CNN and MLP in predicting the time of next event occurrence.

IV. CONCLUSION

In this study, deep learning based a new prediction model has been developed for the time of next purchase in e-commerce. The proposed model has been extensively tested and compared with RF, ARIMA, CNN and MLP using a retail market dataset. The dataset used consists of anonymous user interaction data, such as movie viewing, basket addition and purchase in an e-commerce system. The purchasing activity has been selected as event.

The experimental results show that the average MSE of the developed model improved by 15.30% compared to RF, 13.37% compared to ARIMA, 12.63% compared to CNN and 11.19% compared to MLP. According to RMSE metric, the average RMSE of the developed model improved by 7.96% compared to RF, 6.92% compared to ARIMA, 6.52% compared to CNN and 5.76% compared to MLP. According to MAE metric, the average MAE of the developed model improved by 3.19% compared to RF, 2.27% compared to ARIMA, 1.61% compared to CNN and 1.18% compared to MLP. The experimental results show that the developed model has been more successful than RF, ARIMA, CNN and MLP to predict the time of the next purchase.

REFERENCES

[1] F. Martínez-Álvarez, A. Troncoso, G. Asencio-Cortés, J. C. Riquelme, "A survey on data mining techniques applied to electricity-related time series forecasting," Energies, vol. 8, no. 11, pp. 13162-13193, 2015. doi:10.3390/en81112361

[2] S. Lahmiri, "A variational mode decompoisition approach for analysis and forecasting of economic and financial time series," Expert Systems with Applications, vol. 55, pp. 268-273, Aug. 2016. doi:10.1016/j.eswa.2016.02.025.

[3] R. Adhikari, R. K. Agrawal, "An introductory study on time series modeling and forecasting," LAP Lambert Academic Publishing, Germany, 2013, arXiv. doi:10.13140/2.1.2771.8084

[4] I. E. Livieris, E. Pintelas, P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," Neural Computing and Applications, pp. 1-10. doi:10.1007/s00521-020-04867-x

[5] S. Bisgaard, C. L. Jennings, M. Kulahci, "Introduction to time series analysis and forecasting by example," John Wiley & Sons, Canada, pp. 73-171, 2015. doi:10.1002/9781118056943

[6] C. Chatfield, "The analysis of time series: an introduction," CRC press, pp. 73-103, 2016. doi:10.4324/9780203491683

[7] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, "Time series analysis: forecasting and control," John Wiley & Sons, pp. 1-88, 2015. doi:10.1002/9781118619193

[8] S. Siami-Namini, N. Tavakoli, A. S. Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," In 2018 17th IEEE International Conference on Machine Learning and Applications, Florida, pp. 1394-1401, Dec. 2018. doi:10.1109/ICMLA.2018.00227

[9] O. Claveria, S. Torra, "Forecasting tourism demand to Catalonia: Neural networks vs. time series models," Economic Modelling, vol. 36, pp. 220-228, Jan. 2014. doi:10.1016/j.econmod.2013.09.024

[10] P. Chujai, N. Kerdprasop, K. Kerdprasop, "Time series analysis of household electric consumption with ARIMA and ARMA models," In: Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, pp. 295-300, Mar. 2013.

[11] M. J. Kane, N. Price, M. Scotch, P. Rabinowitz, "Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks," BMC bioinformatics, vol. 15, no. 1, Aug. 2014. doi:10.1186/1471-2105-15-276

[12] R. Fu, Z. Zhang, L. Li, 2016, "Using LSTM and GRU neural network methods for traffic flow prediction," In: Chinese Association of Automation, Youth Academic Annual Conference, Jinzhou, pp. 324-328, Nov. 2016. doi:10.1109/YAC.2016.7804912

[13] M. H. Amini, A. Kargarian, O. Karabasoglu, "ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation," Electric Power Systems Research, vol. 140, pp. 378-390, Nov. 2016. doi:10.1016/j.epsr.2016.06.003

[14] Y. Yang, W. Gao, C. Guo, "Aero-Engine Lubricating Oil Metal Content Prediction Using Non-stationary Time Series ARIMA Model," In: Computational Intelligence and Design, 2017 10th International Symposium, Hangzhou, pp. 51-54, Dec. 2017. doi:10.1109/ISCID.2017.173

[15] J. Zheng, C. Xu, Z. Zhang, X. Li, "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network," In: 2017 51st Annual Conference on Information Sciences and Systems, Baltimore, pp. 1-6, Mar. 2017. doi:10.1109/CISS.2017.7926112

[16] P. Sobreiro, D. Martinho, A. Pratas, "Sales forecast in an IT company using time series," In: 2018 13th Iberian Conference on Information Systems and Technologies, pp. 1-5, June 2018. doi:10.23919/CISTI.2018.8399191

[17] S. McNally, J. Roche, S. Caton, "Predicting the price of Bitcoin using Machine Learning," In: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing, pp. 339-343, Mar. 2018. doi:10.1109/PDP2018.2018.00060

[18] H. Wang, K. Wang, H. Zhao, Y. Yue, "Prediction of User Behavior in Smart Home Based on Improved ARIMA Model," In: 2018 IEEE International Conference on Mechatronics and Automation, pp. 298-302, Aug. 2018. doi:10.1109/ICMA.2018.8484380

[19] R. Fu, Z. Zhang, L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, pp. 324-328, Nov. 2016. doi:10.1109/YAC.2016.7804912

[20] K. Chen, Y. Zhou, F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," In: 2015 IEEE international conference on big data, pp. 2823-2824, Oct. 2015. doi: 0.1109/BigData.2015.7364089

[21] K. K. Sarma, "Neural network based feature extraction for assamese character and numeral recognition," Int. J. Artif. Intell, vol. 2, no. S09, pp. 37-56, 2009.

[22] C. Pozna, R. E. Precup, J. K. Tar, I. Škrjanc, S. Preitl, "New results in modelling derived from Bayesian filtering," Knowledge-Based Systems, vol. 23, no. 2, pp. 182-194, March 2010. doi: 10.1016/j.knosys.2009.11.015

[23] R. A. Gil, Z. C. Johanyák, T. Kovács, "Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation," Int. J. Artif. Intell, vol. 16, no. 1, pp. 20-40, 2018.

[24] A. Albu, R. E. Precup, T. A. Teban, "Results and Challenges of Artificial Neural Networks Used for Decision-Making and Control In Medical Applications," Facta Universitatis, Series: Mechanical Engineering, vol. 17, no. 3, pp. 285-308, 2019. doi:10.22190/FUME190327035A

[25] R. H. Shumway, D. S. Stoffer, "Time series analysis and its applications with R exaamples," Springer, pp. 375-376, 2000. doi:10.1007/978-1-4419-7865-3

[26] Y. Zhang, P. Wang, P. Cheng, S. Lei, "Wind speed prediction with wavelet time series based on Lorenz disturbance," Advances in Electrical and Computer Engineering, vol. 17, no.3, pp. 107-115, Aug. 2017. doi:10.4316/AECE.2017.03014

[27] A. Harvey, S. J. Koopman, "Structural time series models," Wiley StatsRef: Statistics Reference Online, New Jersey, ABD, pp. 106-166, 2014. doi:10.1002/0470011815.b2a12069

[28] M. Valipour, M. E. Banihabib, S. M. R. Behbahani, "Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir," Journal of hydrology, vol. 476, pp. 433-441, Jan. 2013. doi:10.1016/j.jhydrol.2012.11.017

[29] A. S. Weigend, "Time series prediction: forecasting the future and understanding the past," Routledge, New York, ABD, pp. 345-491 2018. doi:10.4324/9780429492648

[30] W. C. Wang, K. W. Chau, D. M. Xu, X. Y. Chen, "Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition," Water Resources Management, vol. 29, no. 8, pp. 2655-2675, June 2015. doi:10.1007/s11269-015-0962-6

[31] G. E. Box, G. M. Jenkins, "Time series analysis, control, and forecasting," San Francisco, CA: Holden Day, New Jersey, ABD, pp. 46-89, 1976. doi:10.2307/2284112

[32] M. Längkvist, L. Karlsson, A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," Pattern Recognition Letters, vol. 42, pp. 11-24, June 2014. doi:10.1016/j.patrec.2014.01.008

[33] H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P. Muller, "Deep learning for time series classification: a review," Data Mining and Knowledge Discovery, vol. 33, no. 4, pp. 917-963, 2019. doi:10.1007/s10618-019-00619-1

[34] D. Graupe, "Principles of artificial neural networks," World Scientific, World Scientific, Singapore, pp. 9-16, 2013. doi:10.1142/8868

[35] A. Namozov, Y. Cho, "An Efficient Deep Learning Algorithm for Fire and Smoke Detection with Limited Data," Advances in Electrical and Computer Engineering, vol. 18, no. 4, pp.121-129, Nov. 2018. doi:10.4316/AECE.2018.04015

[36] O. Irsoy, E. Alpaydin, "Continuously constructive deep neural networks," IEEE transactions on neural networks and learning systems, 2019. doi:10.1109/TNNLS.2019.2918225

[37] A. Ardakani, C. Condo, W. J. Gross, "Sparsely-connected neural networks: towards efficient vlsi implementation of deep neural networks," arXiv preprint arXiv:1611.01427, 2016.

[38] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, "Understanding neural networks through deep visualization," arXiv preprint arXiv:1506.06579, 2015.

[39] M. Hussain, J. J. Bird, D. R. Faria, "A study on cnn transfer learning for image classification,". In: UK Workshop on Computational Intelligence, Cham, pp. 191-202, Sept. 2018. doi: 10.1007/978-3-319-97982-3_16

[40] K. Park, D. H. Kim, "Accelerating image classification using feature map similarity in convolutional neural networks," Applied Sciences, vol. 9, no.1, 2019. doi:10.3390/app9010108

[41] P. Araujo, G. Astray, J. A. Ferrerio-Lage, J. C. Mejuto, J. A. Rodriguez-Suarez, B. Soto, B. "Multilayer perceptron neural network for flow prediction," Journal of Environmental Monitoring, vol. 13, no. 1, pp. 35-41, 2011. doi:10.1039/C0EM00478B

[42] A. Graves, A. R. Mohamed, G. Hinton, "Speech recognition with deep recurrent neural networks," In: 2013 IEEE international conference on Acoustics, speech and signal processing, Vancouver, pp. 6645-6649, May 2013. doi:10.1109/ICASSP.2013.6638947

[43] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, "Long short term memory networks for anomaly detection in time series," In: 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, pp.89-94, Apr. 2015.

[44] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," In: 2014 IEEE symposium on computational intelligence in ensemble learning (CIEL), pp. 1-6, Orlando, Dec. 2014. doi:10.1109/CIEL.2014.7015739

[45] O. Buyuk, M. L. Arslan, "Combination of Long-Term and Short-Term Features for Age Identification from Voice," Advances in Electrical and Computer Engineering, vol. 18, no. 2, pp. 101-109, May 2018. doi:10.4316/AECE.2018.02013

[46] L. Wang, Y. Zeng, T. Chen, "Back propagation neural network with adaptive differential evolution algorithm for time series forecasting," Expert Systems with Applications, vol. 42, no. 2, pp. 855-863, Feb. 2015. doi:10.1016/j.eswa.2014.08.018.

[47] B. Wang, X. Zhu, Q. He, G. Gu, "The forecast on the customers of the member point platform built on the blockchain technology by ARIMA and LSTM," In: 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis, Chengdu, pp. 589-593, Apr. 2018. doi:10.1109/ICCCBDA.2018.8386584

[48] https://www.kaggle.com/retailrocket/ecommerce-dataset, (Accessed: Aug. 2019).