KSÜ Mühendislik Bilimleri Dergisi, 27(2), 2024
Araştırma Makalesi

KSU J Eng Sci, 27(2), 2024
Research Article

Kahramanmaras Sutcu Imam University
Journal of Engineering Sciences

Geliş Tarihi : 01.12.2023
Kabul Tarihi :26.03.2024

Received Date : 01.12.2023
Accepted Date : 26.03.2024

A DEEP LEARNING-BASED DEMAND FORECASTING SYSTEM FOR PLANNING ELECTRICITY GENERATION

ELEKTRİK ÜRETİMİNİN PLANLANMASI İÇİN DERİN ÖĞRENME TABANLI TALEP TAHMİN SİSTEMİ

Muhammet Mustafa GÖKÇE[1*] (ORCID: 0000-0003-1214-1698)
Erkan DUMAN[2] (ORCID: 0000-0003-2439-7244)

[1] Turkish Electricity Transmission Corporation, Ankara, Türkiye
[2] Firat University, Department of Computer Engineering, Elazig, Türkiye

*Sorumlu Yazar / Corresponding Author: Muhammet Mustafa GÖKÇE, gokcemmustafa@gmail.com.tr

ABSTRACT

In today's world, where economic and industrial development continues, the importance of electrical energy is constantly increasing. Energy demand should be forecast as precisely as possible to reduce lost energy costs in the system, to plan generation expenditures appropriately, to ensure that market players are not economically harmed, and to deliver quality and uninterrupted energy to system consumers. Balancing the electric energy supply and demand of the system is possible with a forecasting plan. Our research aims to generate hourly electricity consumption load forecasts for the period 2018-2021 using Turkish Electricity Consumption Data and meteorological data, with the addition of time and public holiday features. The forecasting performance of the models is evaluated by training multiple machine learning models and deep neural network-based time series models with the data. When the prediction results of our load demand forecasting problem were evaluated, it was seen that deep learning methods gave higher results in prediction success compared to machine learning models. It has been observed that the prediction success of the LSTM model, one of the deep learning methods we use, is higher than the RNN and GRU models. The analysis envisages the elimination of mismatches between energy supply and demand.

Keywords: Electricity demand forecasting, deep learning, LSTM, GRU, RNN

ÖZET

Ekonomik ve endüstriyel gelişimin devam ettiği günümüz dünyasında elektrik enerjisinin önemi sürekli artmaktadır. Sistemdeki kayıp enerji maliyetlerini azaltmak, üretim harcamalarını uygun şekilde planlamak, piyasa oyuncularının ekonomik olarak zarar görmemesini sağlamak ve sistem tüketicilerine kaliteli ve kesintisiz enerji ulaştırmak için enerji talebinin mümkün olduğunca hassas bir şekilde tahmin edilmesi gerekmektedir. Sistemin elektrik enerjisi arz ve talebinin dengelenmesi bir tahmin planı ile mümkündür. Araştırmamız, Türkiye Elektrik Tüketim Verileri ve meteorolojik veriler kullanılarak, zaman ve resmi tatil özellikleri de eklenerek 2018-2021 dönemi için saatlik elektrik tüketim yük tahminleri üretmeyi amaçlamaktadır. Modellerin tahmin performansı, çoklu makine öğrenimi modelleri ve derin sinir ağı tabanlı zaman serisi modelleri verilerle eğitilerek değerlendirilmektedir. Yük talep tahmin problemimizinin tahmin sonuçları değerlendirildiğinde derin öğrenme yöntemlerinin makine öğrenmesi modellerine kıyasla tahmin başarısında daha yüksek sonuçlar verdiği görülmüştür. Kullandığımız derin öğrenme yöntemlerinden LSTM modelinin tahmin başarısının RNN ve GRU modellerinden daha yüksek olduğu gözlemlenmiştir. Analiz, enerji arzı ve talebi arasındaki uyumsuzlukların giderilmesini öngörmektedir.

Anahtar Kelimeler: Elektrik talebi tahmini, derin öğrenme, LSTM, GRU, RNN

INTRODUCTION

ToCite: GÖKÇE, M. M., & DUMAN, E., (2024). A DEEP LEARNING-BASED DEMAND FORECASTING SYSTEM FOR PLANNING ELECTRICITY GENERATION. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 27(2), 511-522.

Countries' social and economic developments are closely related to energy. Electrical energy has taken the most prominent place in human life due to its flexibility and ease of conversion (Nisanci, 2005). Today, electrical load forecasting is essential for energy exchange, transmission and distribution planning, operation and maintenance service, energy demand management, and financial planning of all industry areas (Hong, 2010). Predicting the electricity load demand correctly is necessary for planning energy infrastructure investments and for maximum economic savings (Singh et al., 2012). Technological developments in intelligent grids concern system operators but also industrial and commercial enterprises and end users regarding energy demand forecasting. Big data collected using intelligent grids are used more frequently in demand forecasting (Kim et al., 2019). Climate changes, special days and public holidays, technological developments, and changes in energy policies can affect electricity demand. Estimating this demand accurately is essential to avoid energy losses because forecasting less than the actual consumption value will create an energy supply problem and cause power outages. Making forecasting, on the contrary, causes excess electricity production and creates a disadvantage in terms of system and cost (Amjady, 2001).

This study focuses on Turkey's hourly total electricity consumption demand forecasting using deep learning algorithms. In the dataset we used in the study, the country's consumed electricity load data and the temperature values of the 16 regions where the maximum consumption is reached are available. Finally, the attributes derived from the time variable were added to the dataset. In the study, prediction models were made with several deep learning algorithms using these data. In our study, Turkey's hourly energy demand was modeled by using deep learning methods, Recursive Neural Network (RNN), Long-Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) algorithms for our time series forecasting problem with the dataset, and the forecasting performance between these models was compared. This study is critical in forecasting the amount of electrical load Turkey needs hourly and correctly planning the electricity grid operation.

RELATED WORKS

Chan et al., in their work, combined a Convolutional Neural Network (CNN) with an artificial intelligence tuned Support Vector Machine and made an electricity consumption forecasting within the scope of smart grid studies. For short-term electrical load forecasting, CNN is fed from transfer learning (TL) and combined with the SVM method. In the study, due to the difficulty of training the CNN method from scratch, it was fine-tuned with a Deep Learning or Transfer Learning method beforehand. Thus, the importance of adjustments made with Transfer Learning in CNN and SVM applications has been understood (Chan et al., 2019).

Yildiz et al. made the electrical load forecasting for commercial buildings. Their work aimed to create an economic and environmental contribution by correctly predicting electricity usage demand and reducing greenhouse gas emissions. The study used electrical load data from two critical commercial buildings for day-ahead hourly consumption forecasting. When the model results were examined, it was seen that the model trained with Bayesian Regulated Backpropagation Artificial Neural Networks showed the highest performance when evaluated with error metrics in MAPE and RMSE (Yildiz et al., 2017).

Madrid et al., have worked to increase the forecast accuracy weekly with several machine learning methods to overcome the planning problems in renewable energy sources and to minimize electricity generation costs in short-term load forecasting. In their study, they trained five machine learning methods using previous electrical load data, weather, and holiday data. Among these methods, it has been observed that the Extreme Gradient Boosting Regressor (XGBoost) algorithm outperforms the methods based on neural networks in estimation (Madrid and Antonio, 2021).

Cunkas et al. made long-term forecasting studies with models on Turkey's electricity demand. Their studies observed the effects of weather conditions and economic indicators on electrical load demand using the Recursive Neural Networks (RNN) model and Back Propagation (BP) architecture. Among the factors affecting the market, economic factors were influential in short-term forecasts. On the other hand, economic conditions impacted forecasts. When the results of the two models are compared, it has been revealed that the RNN model gives better results than BP in long-term predictions (Cunkas and Altun, 2010).

Biskin et al. worked on Turkey's prospective electricity consumption forecast using Long Short-Term Memory (LSTM) and another deep learning method, Gated Repetitive Unit (GRU). Their study used electrical data from previous years and made hourly and three-hour forecasts. Considering the forecasting performances, the GRU method performed better than LSTM (Biskin and Cifci, 2021).

Banik et al. developed a prediction model using factors such as temperature, humidity, pressure, and air density that are likely to directly affect an electric load in the Indian state of Tripura. Random Forest Algorithm and XGBoost are used together in the model. The model they used in their work has been compared with different machine learning methods such as SVR, NN, and AdaBoost. It has been shown that the model is suitable for short, medium, and long-term electrical load estimation and has achieved successful results with high accuracy values (Banik et al., 2021).

Yan et al. trained their model by combining LSTM and CNN networks with household power consumption data. They used the model to increase the power consumption estimation accuracy and shorten the estimation time. Training, the model they developed, made performance comparisons with LSTM, ARIMA, and SVR. They have proven that their proposed model outperforms most other approaches (Yan et al., 2018).

MATERIALS AND METHODS

Recurrent Neural Network (RNN)

RNN, also called Recurrent Neural Networks, was founded by Jeff Elman in 1990 as a Simple Recurrent Network (SRN). The purpose of using RNN networks is to analyze, process, and classify sequential data. Iterative Neural Networks differ from feedforward networks in that they have feedback connections from the previous layer and the current layer. In this way, the RNN architecture has a structure that processes information continuously and consists of many loops. Training of RNNs takes place with the help of the backpropagation algorithm (Elmas, 2018).

Sequential data that vary depending on time are processed with recurrent neural networks. Having a sequential flow of data expands the usage area of RNN and enables it to be used in fields such as speech and natural language processing (Aydogan, 2019). The RNN structure is given in Figure 1.

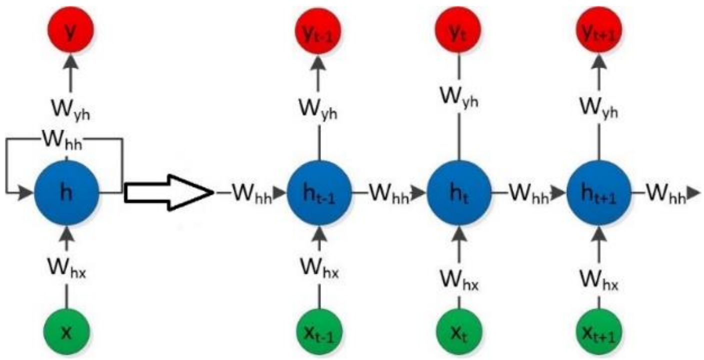


Figure 1. RNN Cell Structure (Zheng et al., 2017)

As shown in equations (1), (2), and (3), fw represents a function with w parameter, x_t layer input, y_t layer output, W_{hx} weight matrix in input nerve, W_{hh} weight matrix in recurrent nerve, W_{yh} weight matrix in output nerve (Elmas, 2018).

$$h_t = fw(h_{t-1}, x_t) \tag{1}$$

$$h_t = fw(W_{hx}x_t + W_{hh}x_{t-1}) \tag{2}$$

$$y_t = W_{yh}h_t \tag{3}$$

In an iterative neural network algorithm, input (x_t) is given to the network in a single time step. Next, the current state (h_t) is calculated using the current and previous state of the input. The current h_t, becomes h_{t-1} for the next step. As needed by the deep learning problem, steps are taken, and all previous state information is brought together. After the completed time steps, the output (y_t) is calculated using the final state of the current state. Then, error detection is made by comparing the actual output with the output value. The error value is used in updating the weights by giving it back to the network. With the update process, the network is trained (Elmas, 2018).

RNN architecture has some disadvantages. Gradient value is used to determine all weight values in architecture. However, when this process is used in long networks, the loss effect may decrease excessively, and the gradient value may be lost. Since all the layers in the architecture are related to each other by the multiplication process, their derivatives can be lost, called vanishing gradients. In the case of the opposite situation, the gradients increase excessively, and the gradient explosion (exploding gradients) event may occur (Aydogan, 2019).

Recursive neural networks store information with a hidden state structure corresponding to each time step they contain. The information held is the memory of the network. It is impossible to go back very far in the RNN architecture in this memory, which stores information about what operations are performed at all stages of learning. To overcome this disadvantage, LSTM architecture, a more advanced type of RNN, is used (Pervan & Keles, 2019).

Long Short-Term Memory (LSTM)

Long Short-Term Memory is based on memory cell structure rather than neural-based architecture. The memory cell, which can preserve its value in the short or long term, can also keep the previous values in its memory thanks to this feature. Thus, he learns about long-term dependencies.

LSTM networks were proposed by Hochreiter and Schmidhubr in 1997. The architecture of the network consists of memory cells. The Long Short Term Memory cell has three gates. At the entrance gate, the first of these gates, the time to enter new information into memory is controlled. The forgetting gate, another gate in the cell, performs its control task by ensuring that existing information is forgotten and new data is remembered. The exit door, the last door in the cell, controls the usage time of the information at the exit. However, there are also weights in the cell that control these gates. These weights are generally optimized by training with the Back Propagation learning algorithm (Elmas, 2018).

LSTM structures were developed to overcome the vanishing gradient problem in RNN architecture. In the RNN architecture, the problem of forgetting the first input arises as the information is lost at every stage. The longer the time series, the greater the complexity of the forgetting problem. To solve these problems, the network needs some long-term memory. However, as the complexity of the problems increases, the number of layers must also increase. In complex problems, the learning speed increases from the first layer to the next layer. Conversely, the first layers tend to learn fast, and the next layers tend to learn slowly. This explains the concept of exploding gradient. LSTM architecture has been developed to eliminate such situations that cause insufficient performance from the models (Durgun, 2018). Figure 2 shows the internal structure of the LSTM cell.

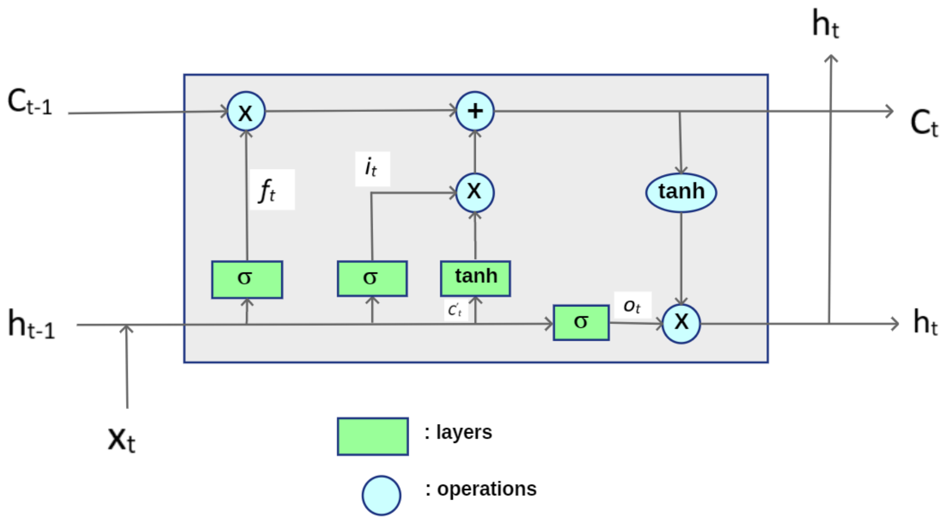


Figure 2. LSTM Cell Structure (Aydogan, 2019)

C_t (Cell State): This channel continuously transfers information from one cell to another.

f_t (Forget Gate): With cell states, LSTM acts as a filter in the transfer of information to other cells and decides which information will be forgotten with the help of this forget gate.

i_t (Input Gate): With this gate, it is decided to record the information with cell states and transfer it to other cells. Then the cell state is updated and ready to transmit information to the adjacent cell.

o_t (Output Gate): While determining the output value, it is decided by this gate. The value from the cell state is not directly output but filtered by *tanh* processing (Aydogan, 2019).

In the LSTM cell, the *sigmoid* function produces a value between 0-1. This generated value decides the part of the information that should be forgotten or transferred. 0 is used when this information will not be transmitted, and one is used if it is to be transmitted. Storing information is the next step. The *sigmoid* function used at the input gate decides what information will be stored. After this stage, the *tanh* function combines these two stages using $\tilde{C}t$ values. Next, the new state information is calculated in the memory cell to calculate the system output. Operations are expressed mathematically as in equations (4), (5), (6), (7), (8), (9) (Tan et al., 2015).

$$i_t = \sigma(\mathrm{w_i}.[\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_i} \quad (4)$$

$$\tilde{\mathrm{C}}_\mathrm{t} = tanh(\mathrm{W_c}.[\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_i} \quad (5)$$

$$\tilde{\mathrm{C}}_\mathrm{t} = tanh(\mathrm{W_c}.[\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_c} \quad (6)$$

$$\tilde{\mathrm{C}}_\mathrm{t} = f_t * \mathrm{C_{t-1}} + i_t * \tilde{\mathrm{C}}_\mathrm{t} \quad (7)$$

$$o_t = \sigma(\mathrm{W}_o.[\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_o} \quad (8)$$

$$\mathrm{h_t} = o_t * tanh(\mathrm{C}_t) \quad (9)$$

Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU), known for its similarity to the LSTM cell, has gate cells that modulate the information flow within the unit but do not contain a different memory cell. The purpose of designing a GRU is to detect adaptively different time-scale dependencies of each repeating unit (Chung et al.,2014).

The difference between the GRU structure, which is used as a kind of LSTM, from LSTM is that there is no cell state. Apart from that, the forget gate and input gate have been removed, and these two gate functions have been combined in the update gate. The structure of the GRU cell shown in Figure 3 is simpler than the LSTM (Aydogan, 2019).

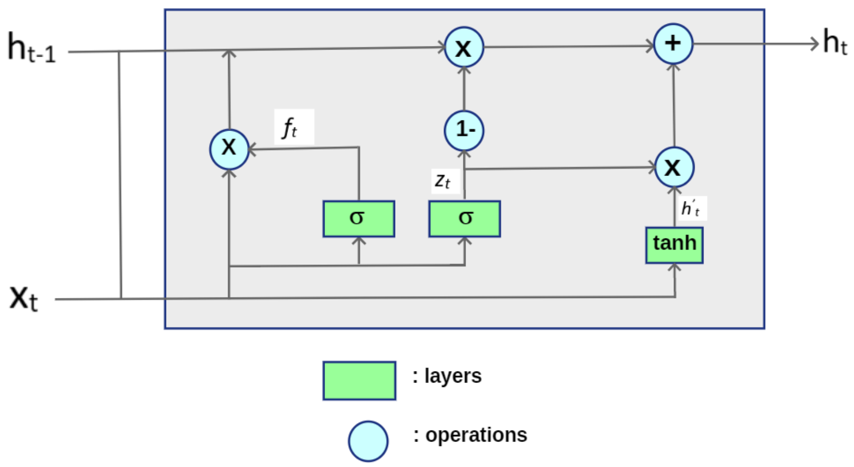


Figure 3. GRU Cell Structure (Aydogan, 2019)

The mathematical operations of the GRU model are given in equations (10), (11), (12), and (13).

$$z_t = (\mathrm{W}_z.[\mathrm{h_{t-1}}, \mathrm{x_t}]) \quad (10)$$

$$r_t = (\mathrm{W}_r.[\mathrm{h_{t-1}}, \mathrm{x_t}]) \quad (11)$$

$$h'_t = tanh(\mathrm{W}[r_t * \mathrm{h_{t-1}}, \mathrm{x_t}] \quad (12)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * h_t \tag{13}$$

Although the internal structure of GRU is similar to LSTM, it is simpler, and less computation is needed to update the hidden state due to this simplicity. The low computational cost facilitates the training of the model. It exhibits the same approach as LSTM to the gradient disappearance problem (Kumar et al., 2018).

Performance Evaluation

R^2 score, MAE, MSE, and RMSE metrics were used to measure the performance of deep learning approaches used in the models. Equation (14) shows the mathematical formulas of MAE, MSE, and RMSE metrics. R^2 score, another performance metric we use in forecasting models, is stated in Equation (15) (Li,2017; Lin et al.,2020).

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(Y_i - \hat{Y}_i)^2}$$

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}(Y_i - \hat{Y}_i)^2 \tag{14}$$

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}\left|Y_i - \hat{Y}_i\right|$$

$$SS_{tot} = \sum(p_i - \bar{s})^2$$

$$SS_{res} = \sum(s_i - p_i)^2 \tag{15}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Data Set and Modeling

In this study, Turkey's electricity consumption forecasting application has been developed by using various deep learning algorithms. The data used in our time series forecasting application are temperature and electricity consumption data of 16 regions based on 24 hours. In the dataset, EXIST Transparency Platform (EXIST, 2022) was used in terms of electrical load, and the POWER Data Access Viewer application (NASA, 2022) was used in terms of temperature data. Four years of data were used in the study, and the time range is between 01.01.2018 and 31.12.2021. The dataset consists of 35064 rows and 18 columns.

Before applying deep learning techniques to our multivariate forecasting problem, the variables of the month, day of the month, day of the week, hour, and public holiday were derived from the time variable, which is the index variable of the dataset, and added to the dataset as new features. Missing and outlier values in the dataset were checked, and it was observed that there was no such value. 90% of the dataset is training data, and 10% is test data. Figure 4 shows the split datasets.

Since functions such as *tanh* and *sigmoid* can be used in the deep learning algorithm, the data has been normalized in the range of 0-1 according to the Robust Scaling class of the Python sci-kit-learn library, taking into account the necessity of scaling in the input data.

In the dataset, the features indicating the temperature suitable for scaling (temp_area1, temp_area2, etc.) were normalized separately as independent variables, and the feature indicating the total electrical load value (load_MW) as the dependent variable.

After scaling the data, the time series data was divided into time steps, and before being given to the models, it was converted into a Numpy array in three-dimensional (3D) form (examples, time steps, features) and reshaped.

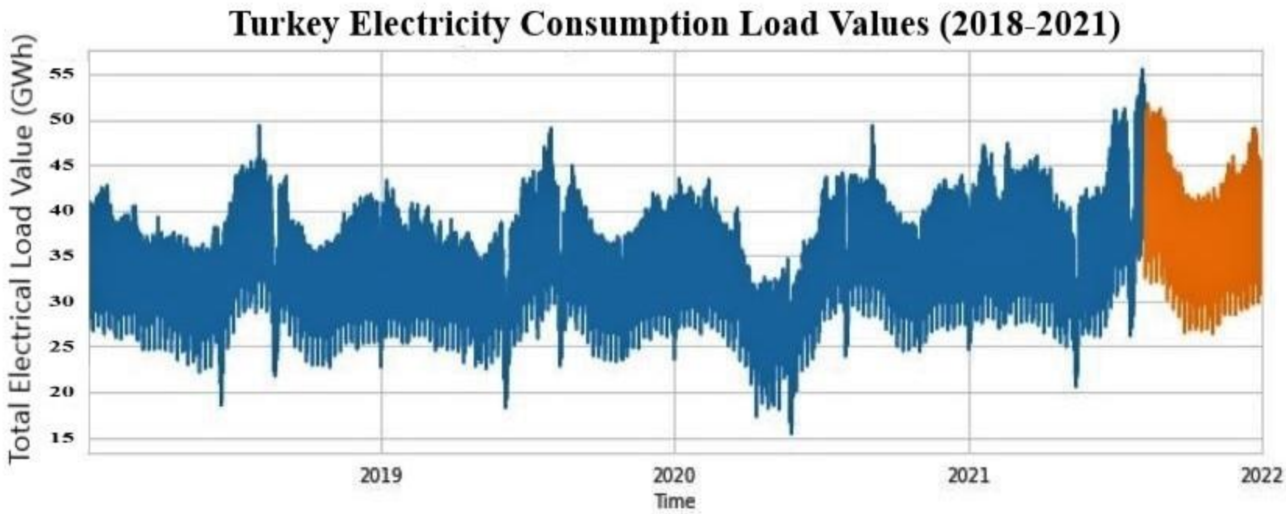


Figure 4. Display of Deep Learning Application Training and Test Data

For all the deep learning models we used in our electrical load estimation, a sequential model was created to give the data to the model in time steps. The models are 128 neurons in the first hidden layer and one neuron in the output layer. During the compilation of the models, mse (mean squared error) was used as the loss function. Adam optimization algorithm was preferred in the models. The batch size value is set as 32 to process the input data in parts of the model. The epoch value was applied as 30 in all models.

The Early Stopping function is used to stop the overfitting of the model when the validation loss (val_loss) value stops decreasing and increases during the model fitting phase. The function ensures that the training is stopped before the model reaches the optimum. The early stopping function parameter is set to patience=5.

When the variable dependencies in the data set are examined, it is observed in Figure 5. how the electrical load variable changes according to the temporal variables.

EXPERIMENTAL WORKS AND RESULTS

After the deep learning models were created and made suitable, the fitting process was applied. The parameters that occur after the model is compiled and the fitting to the data is completed are shown in Table 1.

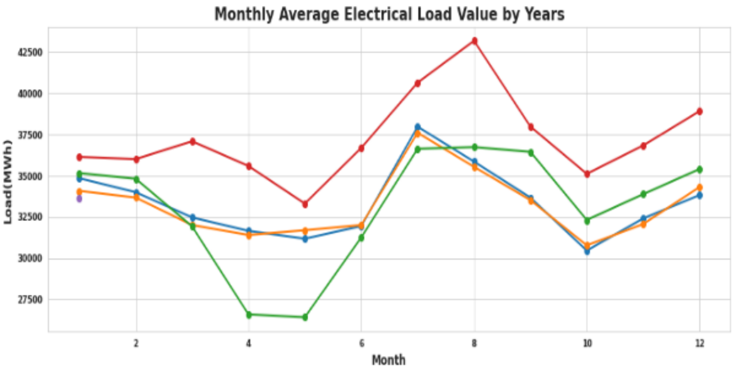
Table 1. Number of Parameters Calculated as a Result of Application of Models

Model	Number of Parameters
Recurrent Neural Network (RNN)	19.457
Long-Short Term Memory (LSTM)	77.441
Gated Recurrent Unit (GRU)	58.497

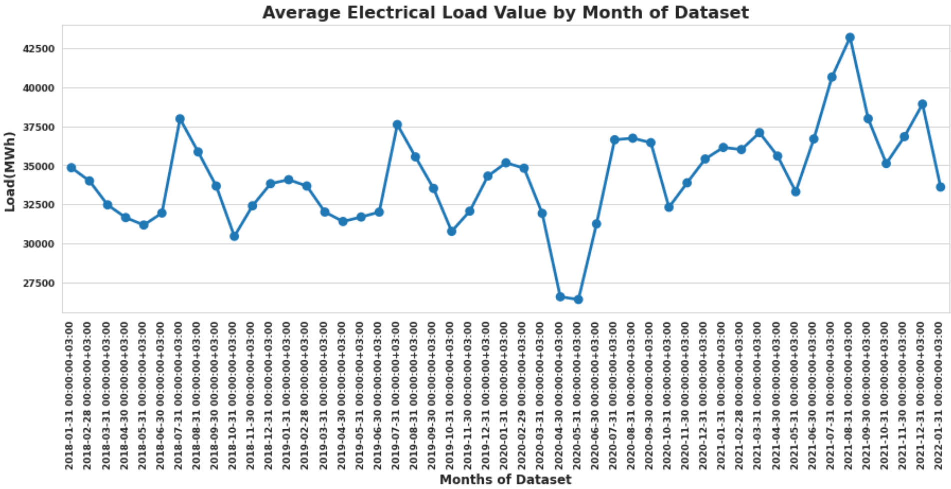
Learning curves were drawn to determine the learning and generalization situations of the deep learning models we designed in our application. From the learning curves of the models (mse/epochs); Considering the training loss representation of the model curves, it was observed that the RNN model had more training loss than the other two models in the first round, and the training loss decreased as the number of rounds increased. Based on the validation loss, it was observed that the loss value of the LSTM model was lower than that of the RNN and GRU models. The training and validation loss data during model training for the RNN, LSTM, and GRU methods were calculated in mse and shown in Figure 6.

At this stage, the test data was evaluated and the prediction success of the deep learning models was measured. Actual test data values and estimated values were compared. Comparisons were first made over scaled values. In order to make a more accurate comparison, the normalized data was inversely transformed and compared with the actual test values.

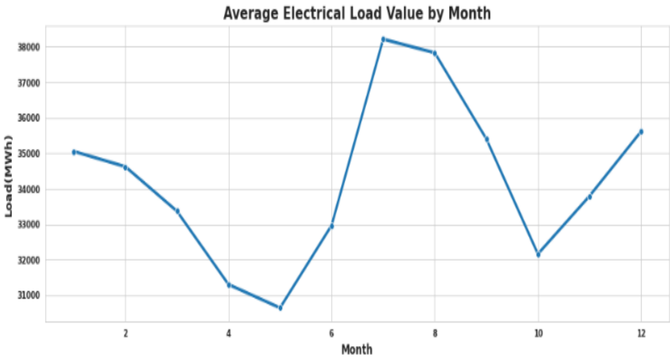
Table 2 and Table 3 show the performance values of the models with different metrics.



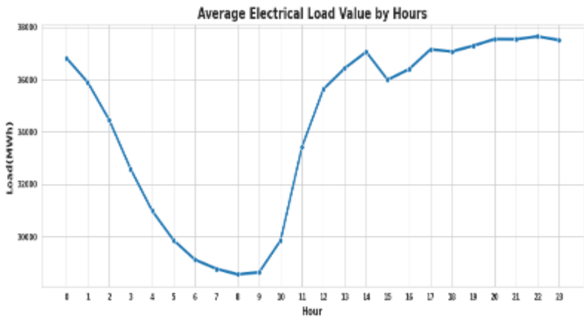
(a)

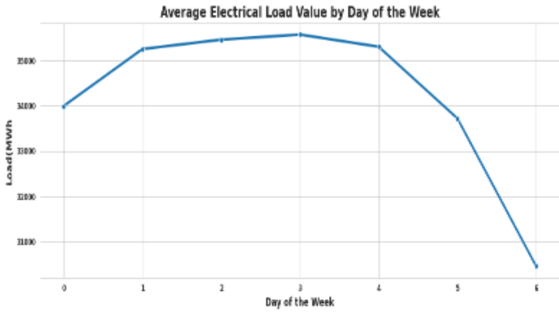


(b)



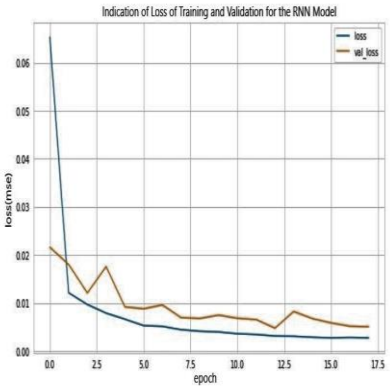
(c)



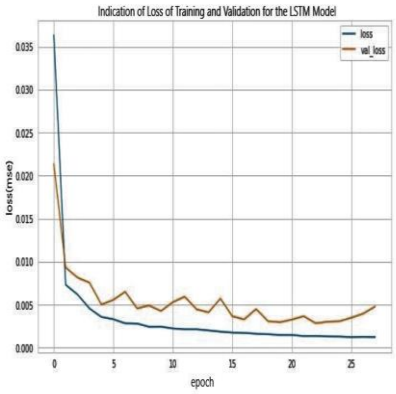


(d) (e)

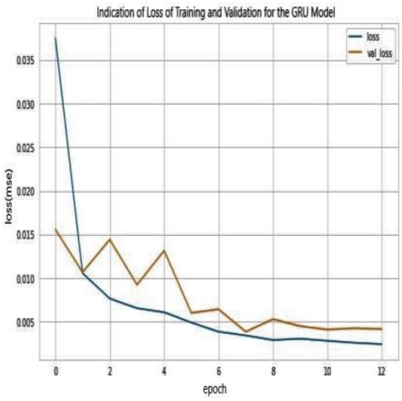
Figure 5. **a.** Monthly Average Electrical Load Value by Years **b.** Average Electrical Load Value by Month of Dataset **c**. Average Electrical Load Value by Month **d.** Average Electrical Load Value by Hours **e.** Average Electrical Load Value by Day of the Week



(a)



(b)



(c)

Figure 6. a. Learning Curve of RNN Model **b.** Learning Curve of LSTM Model **c.** Learning Curve of GRU Model

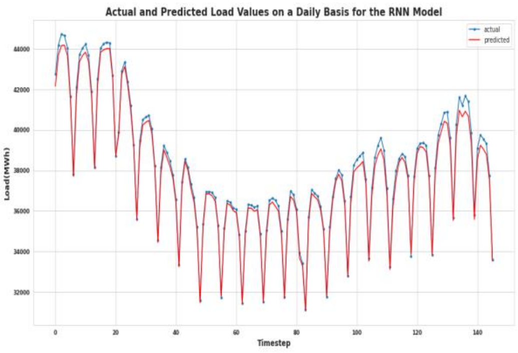
Table 2. Performance Evaluations with Normalized Data

Model Name/Metrics	R2 SCORE	MAE	MSE
RNN	0.9862	0.06274	0.00667
LSTM	0.9935	0.04081	0.00314
GRU	0.9908	0.05008	0.00446

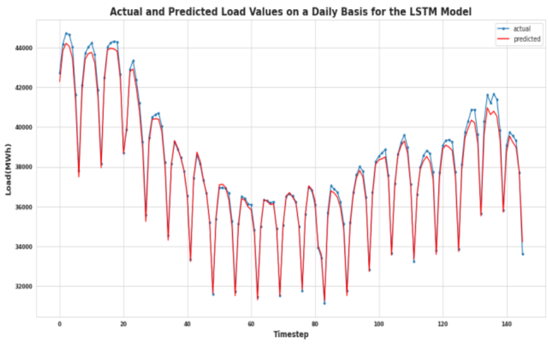
Table 3. Performance Evaluations with Inverse Transformed Data

Model Name/Metrics	R2 SCORE	MAE	MSE
RNN	0.9910	256.95	89637.70
LSTM	0.9946	155.37	53995.14
GRU	0.9925	202.48	74297.94

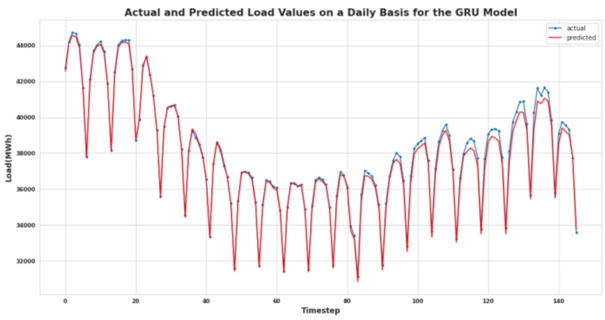
As a result of the evaluation of the model performances, the estimation results of the RNN, LSTM, and GRU models on the test data set and the actual data are shown in Figure 7.



(a)



(b)



(c)

Figure 7. a. Daily Actual and Predicted Load Values Graph for RNN Model **b.** Daily Actual and Predicted Load Values Graph for LSTM Model **c.** Daily Actual and Predicted Load Values Graph for GRU Model

RNN, LSTM, and GRU models were evaluated in their original scale in terms of electrical load value, which is our target variable, and were evaluated with the RMSE evaluation metric, which gave an error in the same unit as itself. Total Absolute Error Amount and Total Percent Error Amount of the models are compared with Energy Exchange Istanbul (EXIST) demand forecasting values in MWh and are shown in Figure 8.

(a) (b)

Figure 8. a. Display of RNN, LSTM, and GRU Models and EXIST Load Prediction in Total Absolute Error **b.** Display of RNN, LSTM, and GRU Models and EXIST Load Prediction in Total Percent Error

In our previous forecasting study, the Linear Model, Random Forest Model, and XGBoost models, which are classical machine learning models, were tested and the values of the models in terms of RMSE were compared (Gokce & Duman, 2022). The performances of all deep learning and machine learning techniques used for our time series electrical load estimation problem were measured in MWh using the RMSE metric and the values found are shown in Table 4.

Table 4. Prediction Performances of all Machine Learning and Deep Learning Techniques

Model	RMSE (MWh)
Linear Regression	2348.35
Random Forest	2075.32
XGBoost	2038.54
Recurrent Neural Network (RNN)	**299.39**
Long-Short Term Memory(LSTM)	**232.36**
Gated Recurrent Unit (GRU)	**272.57**

CONCLUSION

Deep learning models perform well on time series prediction problems. This study focuses on using deep learning methods to predict Turkey's hourly electricity demand in a way that is closest to the actual consumption values. In the dataset we used to train the models, EXIST and POWER Data Access Viewer applications were used as the source. The data consists of real-time electricity consumption data and climatological (temperature) data.

Python programming language and related libraries were used in all analyzes and evaluations for electrical load forecasting. The deep learning methods used for multivariate time series forecasting problems are RNN, LSTM, and GRU methods. Forecasting methods were applied to related dataset separately and their performances on the test data were compared. When the performances in terms of Absolute Error, Squared Error, and Percentage Error were compared, the success order was LSTM, GRU, and RNN. This work to find a solution to the same forecasting problem with classical machine learning approaches has not been as successful as deep learning methods.

Among the machine learning techniques we used, the most successful machine learning technique is XGBoost with an RMSE of 2038.54 MWh. The success of the XGBoost model was followed by Linear Regression with 2075.32 MWh RMSE and Random Forest Model with 2348.35 MWh RMSE.

When the results of all deep learning models applied for the time series forecasting problem are evaluated, LSTM is determined as the most successful technique among the deep learning models with an error value of 232.36 MWh according to the RMSE values of the predicted electricity load in its unit (MWh). The LSTM model was followed by the GRU model with 272.57 MWh RMSE and the RNN model with 299.39 MWh RMSE.

The performance comparisons of the deep learning methods used in this study were compared with the load demand forecast data made by EXIST. It has been observed that all deep learning models used gave better results than EXIST's load prediction results. Reasons such as irregular energy demand and temperature changes were effective in the partial times when forecast performances could be considered low. It was understood that diversifying the number of features and increasing the number of observations in the dataset used in the study can improve the accuracy of load demand forecasting in future studies.

REFERENCES

Aguilar Madrid, E. & Antonio, N.(2021). Short-term electricity load forecasting with machine learning. *Information*, 12(2), 50. https://doi.org/10.3390/info12020050.

Amjady, N. (2001). Short-term hourly load forecasting using time-series modeling with peak load estimation capability. *IEEE Transactions on power systems*, 16(3), 498-505. https://doi.org/10.1109/59.932287.

Aydogan, M. (2019). Performing Text Analysis Using Deep Learning Algorithms in Big Data. Doctoral Thesis. Inonu University Institute of Science and Technology, Malatya.

Banik, R., Das, P., Ray, S. & Biswas, A. (2021). Prediction of electrical energy consumption based on machine learning technique. *Electrical Engineering*, 103(2) , 909-920. https://doi.org/10.1007/s00202-020-01126-z.

Biskin, O.T. & Çifci, A. (2021). Forecasting of Turkey's Electrical Energy Consumption using LSTM and GRU Networks. *Bilecik Şeyh Edebali University Journal of Science and Technology*, 8(2), 656-667. https://doi.org/10.35193/bseufbd.935824.

Chan, S., Oktavianti, I. & Puspita, V. (2019). A deep learning cnn and ai-tuned svm for electricity consumption forecasting: Multivariate time series data. In 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) 0488-0494. IEEE. https://doi.org/10.1109/IEMCON.2019.8936260.

Cunkas, M. & Altun, A.A. (2010). Long term electricity demand forecasting in Turkey using artificial neural networks. *Energy Sources, Part B: Economics, Planning, and Policy*, 5(3) , 279-289. https://doi.org/10.1080/15567240802533542.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Durgun, S. (2018). Long-term forecasting of Turkey's energy demand with artificial intelligence techniques, Master's Thesis. Necmettin Erbakan University Institute of Science and Technology, Konya.

Elmas, C. (2018). Artificial intelligence applications. Seçkin Publishing.

EXIST, Real Time Consumption Data. (2022). https://seffaflik.epias.com.tr/electricity/electricity-consumption/ex-post-consumption/real-time-consumption

Gokce, M.M. & Duman, E. (2022). Performance Comparison of Simple Regression, Random Forest and XGBoost Algorithms for Forecasting Electricity Demand. In 2022 3rd International Informatics and Software Engineering Conference (IISEC) 1-6. IEEE. https://doi.org/10.1109/IISEC56263.2022.9998213

Hong, T. (2010). Short term electric load forecasting. North Carolina State University.

Kim, Y., Son, H.G., & Kim, S. (2019). Short term electricity load forecasting for institutional buildings. *Energy Reports*, 5, 1270–1280. https://doi.org/10.1016/j.egyr.2019.08.086.

Kumar, S., Hussain, L., Banarjee, S. & Reza, M. (2018). Energy load forecasting using deep learning approach-LSTM and GRU in spark cluster. In 2018 fifth international conference on emerging applications of information technology (EAIT) 1-4. IEEE. https://doi.org/10.1109/EAIT.2018.8470406.

Li, J. (2017). Assessing the accuracy of predictive models for numerical data: Not r nor r2, why not? Then what?. *PloS one*, 12(8), e0183250. https://doi.org/10.1371/journal.pone.0183250.

Lin, Z., Cheng, L. & Huang, G. (2020). Electricity consumption prediction based on LSTM with attention mechanism. *IEEJ Transactions on Electrical and Electronic Engineering*, 15(4), 556-562. https://doi.org/10.1002/tee.23088

NASA, Power Data Access Viewer. (2022). https://power.larc.nasa.gov/data-access-viewer/.

Nisanci, M. (2005). Electricity demand and the relationship between electricity consumption and economic growth in Turkey. *Journal of Social Economic Research*, 5(9), 107-121.

Pervan, N., Keles, Y. (2019). Making semantic inference from Turkish texts using deep learning approaches. Doctoral Thesis. Ankara University Institute of Science and Technology, Ankara.

Singh, A.K., Khatoon, S., Muazzam, M. & Chaturvedi, D.K. (2012) Load forecasting techniques and methodologies: A review. In 2012 2nd International Conference on Power, Control and Embedded Systems, 1-10. IEEE. https://doi.org/10.1109/ICPCES.2012.6508132.

Tan, M., Santos, C.D., Xiang, B. & Zhou, B. (2015). Lstm-based deep learning models for non-factoid answer selection. arXiv preprint arXiv:1511.04108.

Yan,K., Wang, X., Du, Y., Jin, N., Huang, H. & Zhou, H. (2018). Multi-step short-term power consumption forecasting with a hybrid deep learning strategy. *Energies*, 11(11), 3089. https://doi.org/10.3390/en11113089.

Yildiz, B., Bilbao, J.I. & Sproul, A.B. (2017). A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renewable and Sustainable Energy Reviews*, 73, 1104-1122. https://doi.org/10.1016/j.rser.2017.02.023.

Zheng, J., Xu, C., Zhang, Z. & Li, X. (2017). 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 (CISS) 1-6. IEEE.