**CSE 655 / Deep Learning and Applications**

**Project Report**

Project Name: PJM Hourly Energy Consumption

Student: Bülent Karadeniz / 225023016010

# **Contex**

1. Introduction

2. Data Collection and Preprocessing

3. Model Development and Configuration

4. Results and Performance Evaluation

5. Conclusions

# **1. Introduction**

* PJM Interconnection LLC (PJM) is a regional transmission organization (RTO) in the United States. It is part of the Eastern Interconnection grid operating an electric transmission system serving all or parts of Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia, and the District of Columbia.
* PJM Interconnection’s operation of the high-voltage power grid, wholesale electricity markets and its long-term planning process provide significant value to the 65 million people in the region it serves. PJM operations, markets and planning result in annual savings of $3.2−4 billion. These savings represent the vital functions that PJM provides and that lead to less cost to consumers:
* The hourly power consumption data comes from PJM's website and are in megawatts (MW).
* The regions have changed over the years so data may only appear for certain dates per region.
* There are 13 regions. In this notebook we will continue using data from Dominion Virginia Power (DOM).
* Developing an RNN and LSTM energy consumption forecasting model and regularizing the model parameters using data from PJM electricity distribution company.
* The main objective of this project is to predict the energy needs of the company for the region in the coming years with the existing data set.

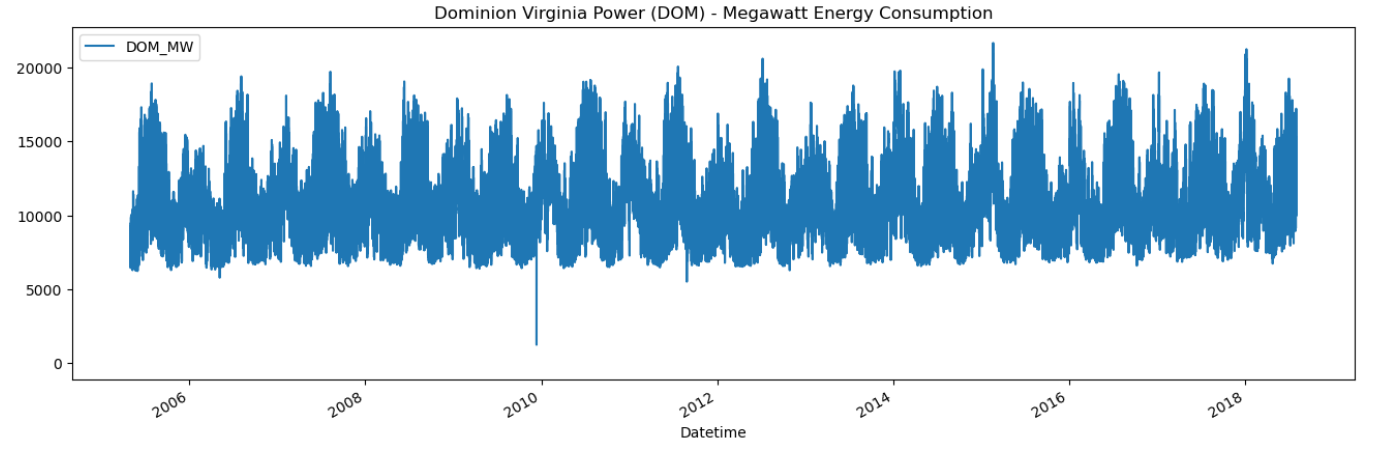
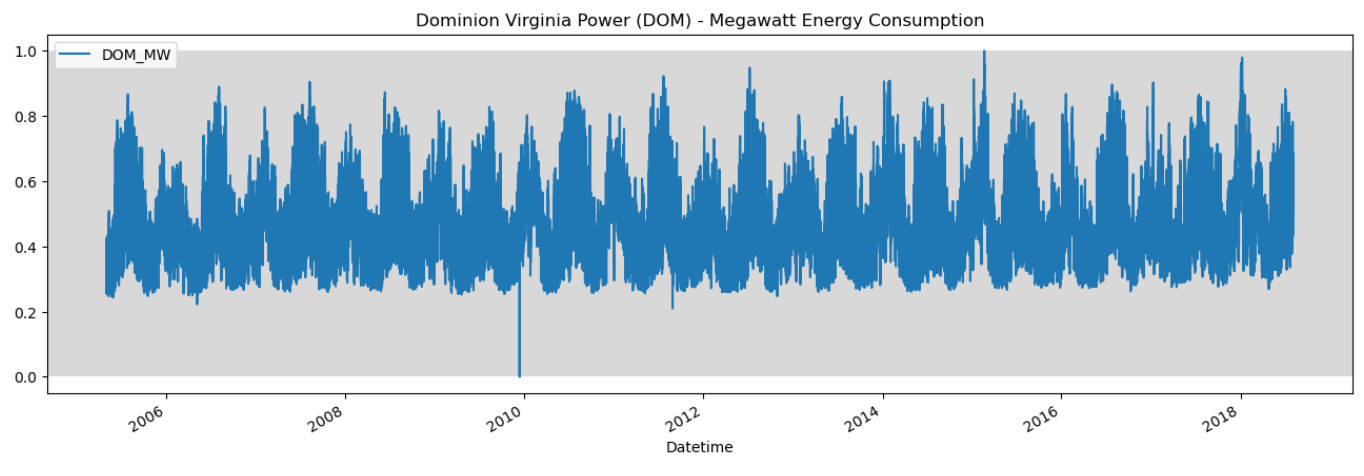
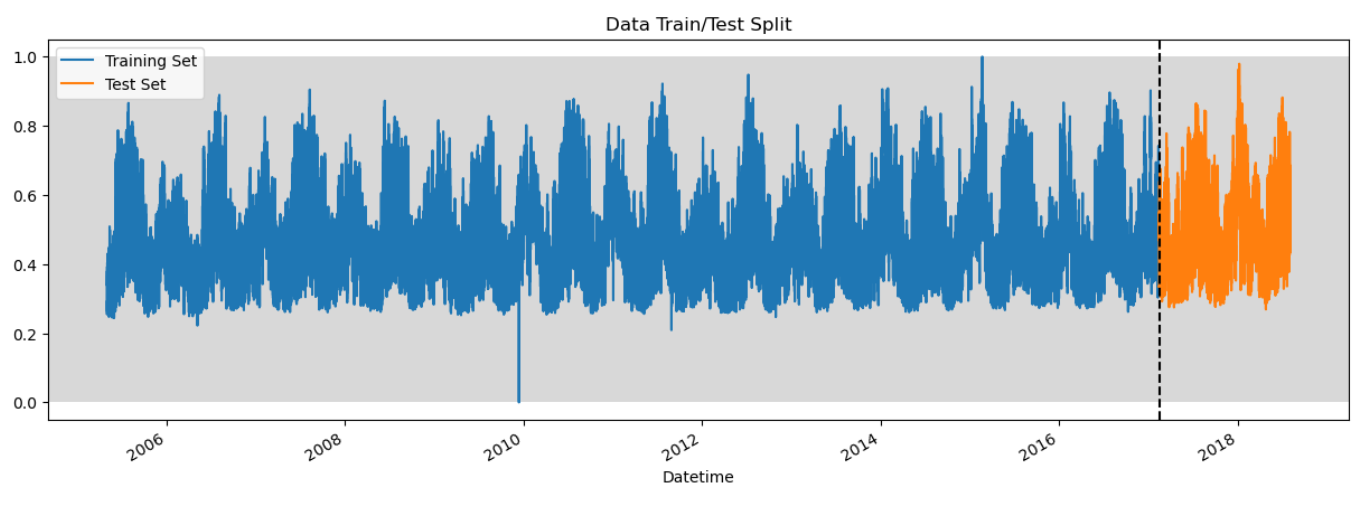
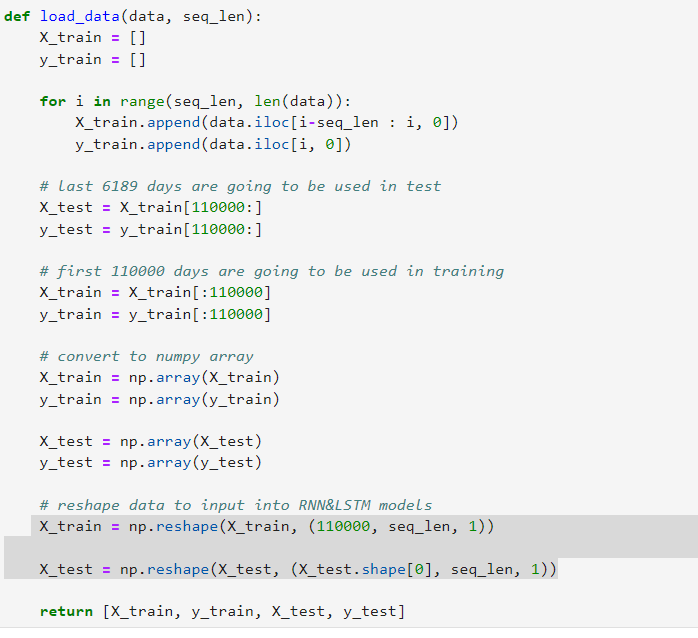
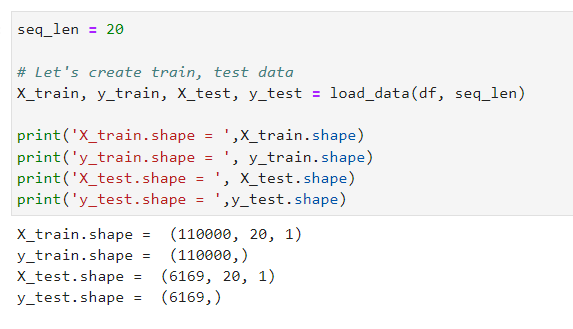
# **2. Data Collection and Preprocessing**

* The data set was obtained from Kaggle. Dominion Virginia Power (DOM) data was used. Data file format csv was used in the model.(DOM\_hourly.csv)
* The data set consists of datetime (hourly) and consumption (megawatt) columns.
* Data covers hourly energy consumption between 2005 and 2018. There are also no null values in the data set.
* Data columns (total 2 columns):

# Column Non-Null Count Dtype

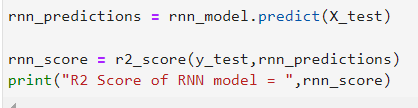
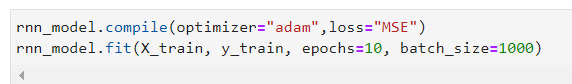
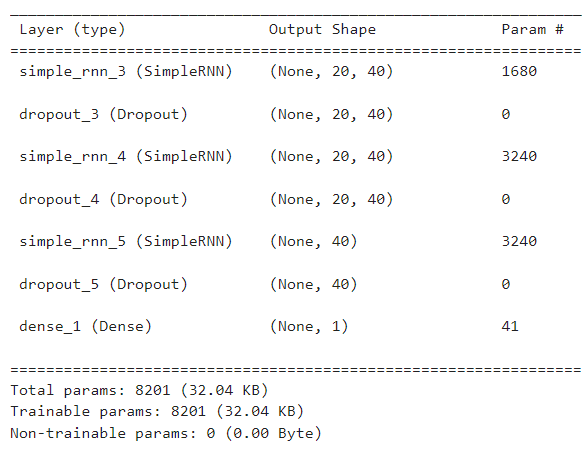
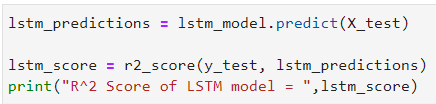
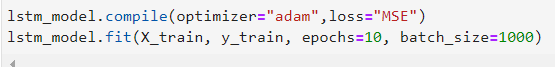
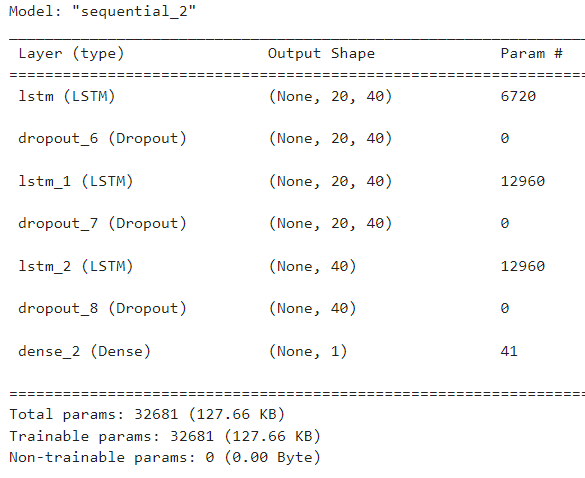
0 Datetime 116189 non-null object

1 DOM\_MW 116189 non-null float64

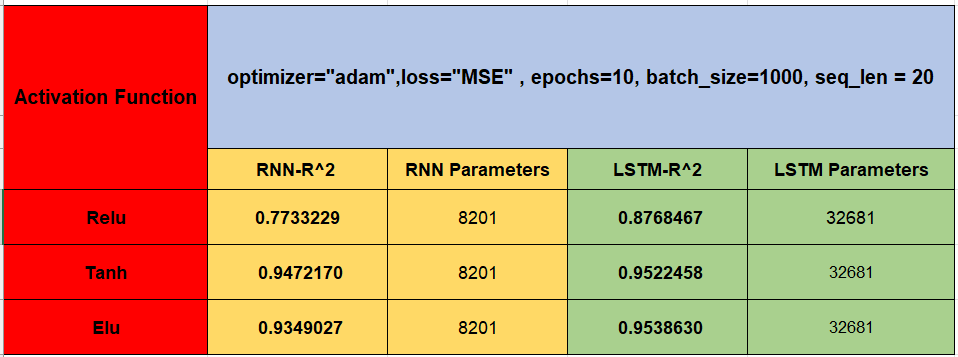
* In the preprocessing process, we first made a datetime index in the data set.
* Now let's observe our data set on the graph
* 
* We did a normalization process. We did normalization, we pulled our data between 0 -1 values.
* 
* 2017-02-13 after this date we will choose the test set
* 
* Prepare data for trainin the RNN and LSTM models
* With the following function block, let's set our data set as training and test data set in a model appropriate way
* 
* **The seq\_len** parameter determines how far back the model will look at historical data, helping the model to capture time dependencies in a memory-aware way.
* We should note that if "seq\_len" is too large, the model can become complex and prone to overlearning.
* We can specify separate seq\_len values for RNN and LSTM
* I Decided Seq\_len = 20, because high seq-len causes overfitting
* 

# **3. Model Development and Configuration**

* Build RNN, LSTM model with different activation functions Relu, Tanh, Elu,

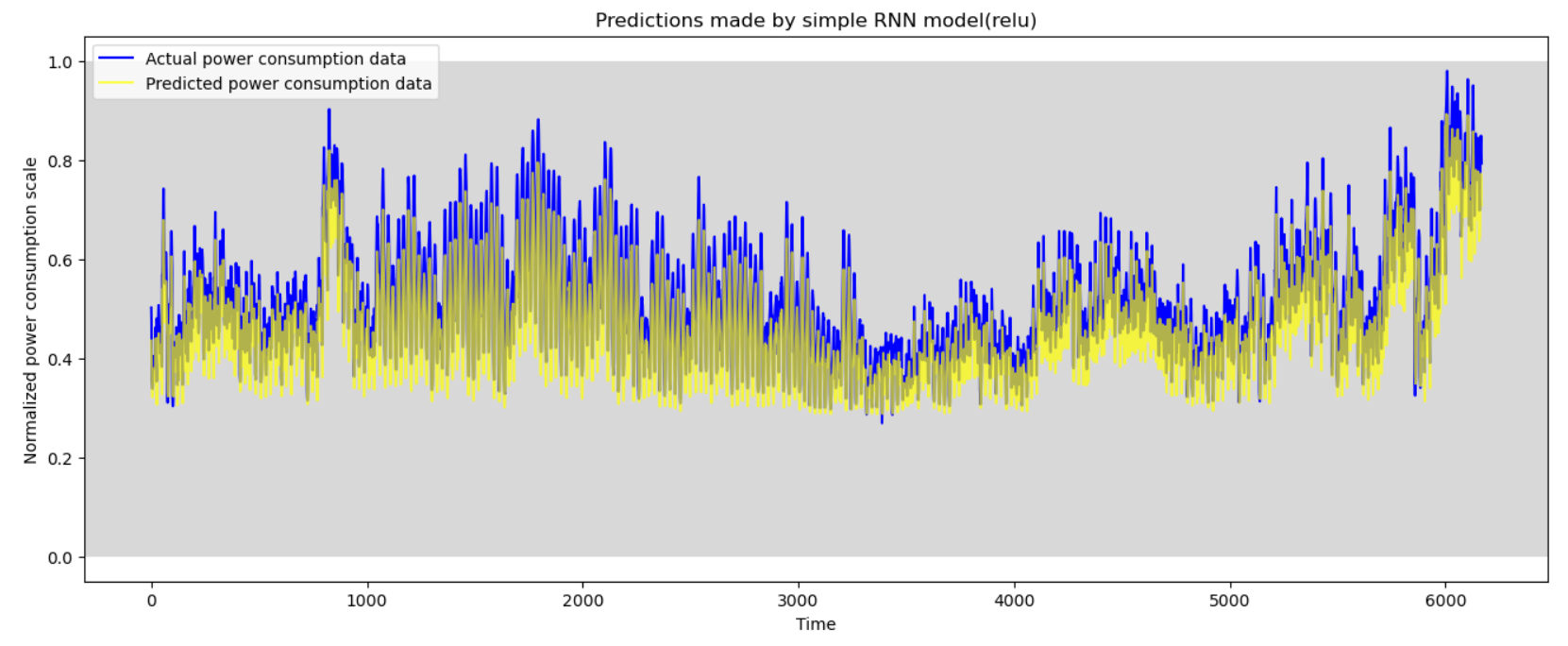


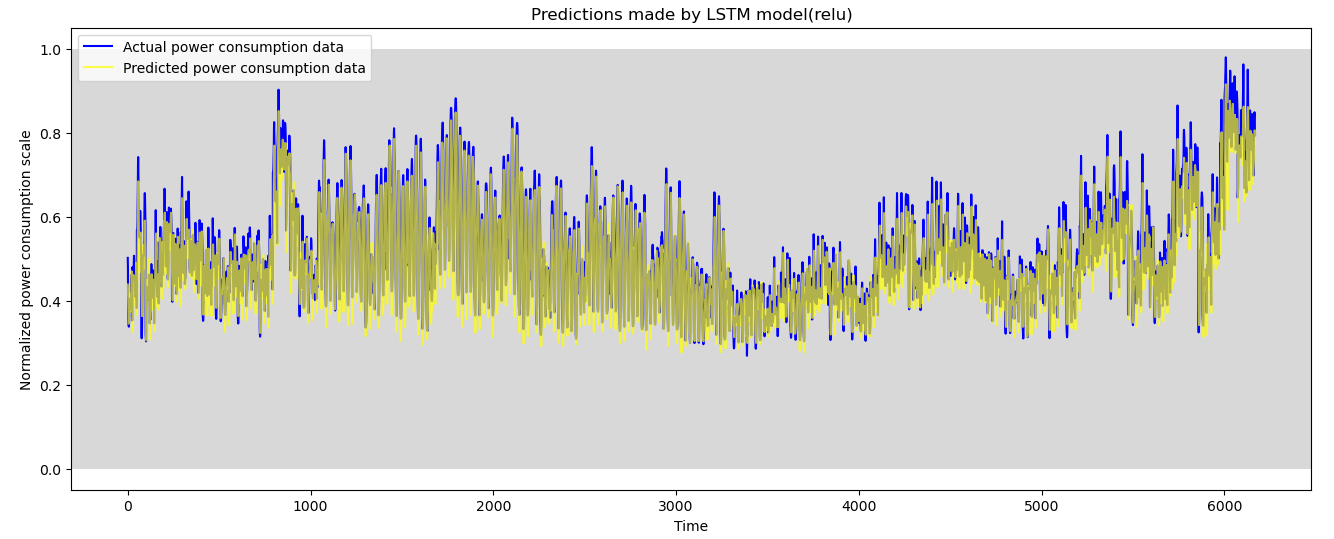
# **4. Results and Performance Evaluation**

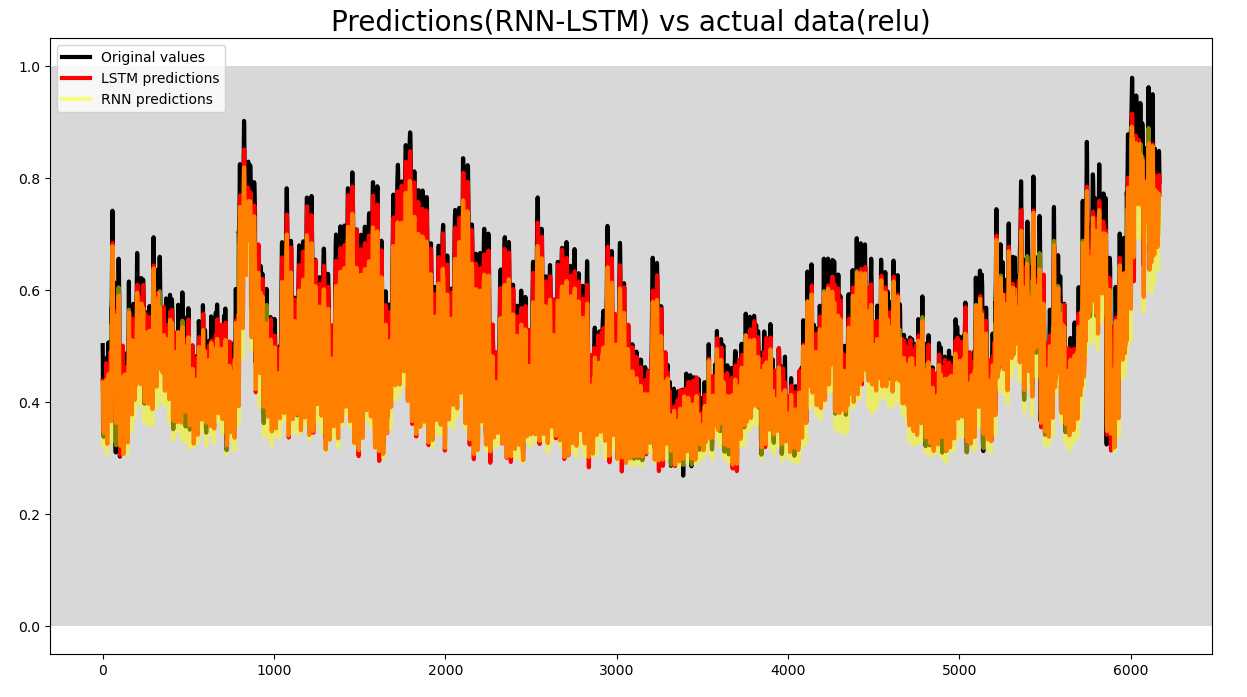


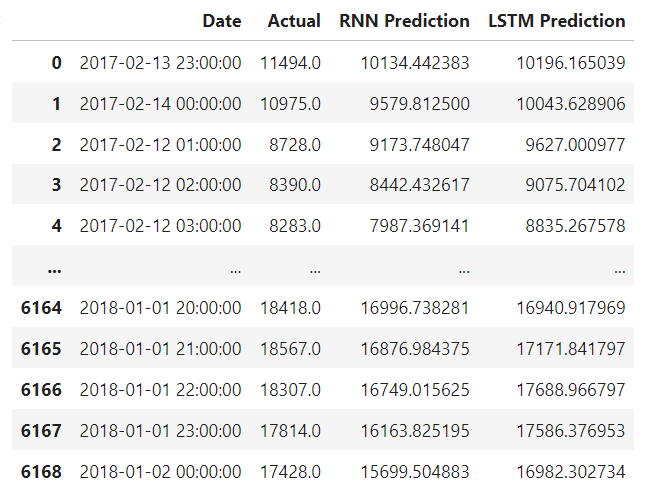
## **RELU**

Here, the RNN and LSTM results obtained by using the activation function relu are shown with graphs, while the prediction values and actual values are given together for comparison.



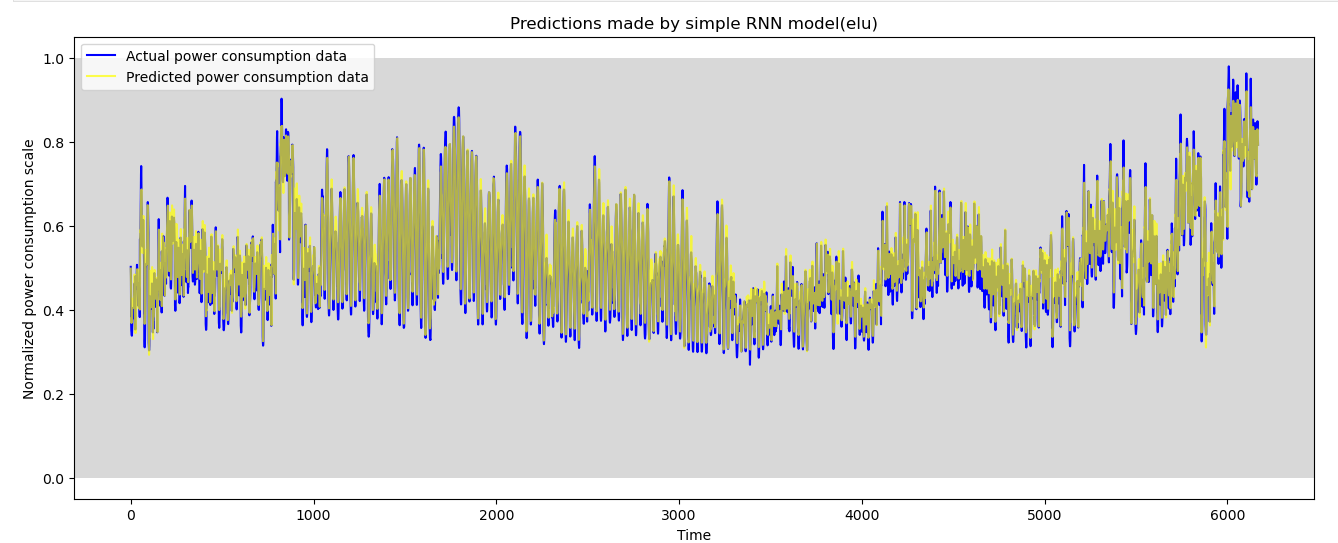


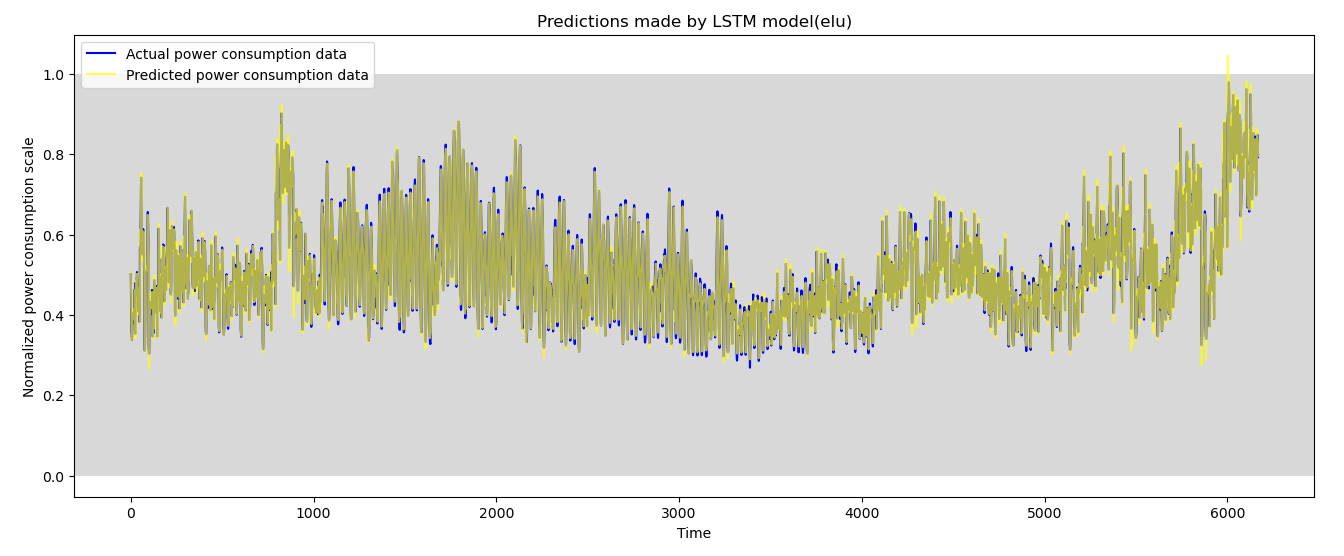


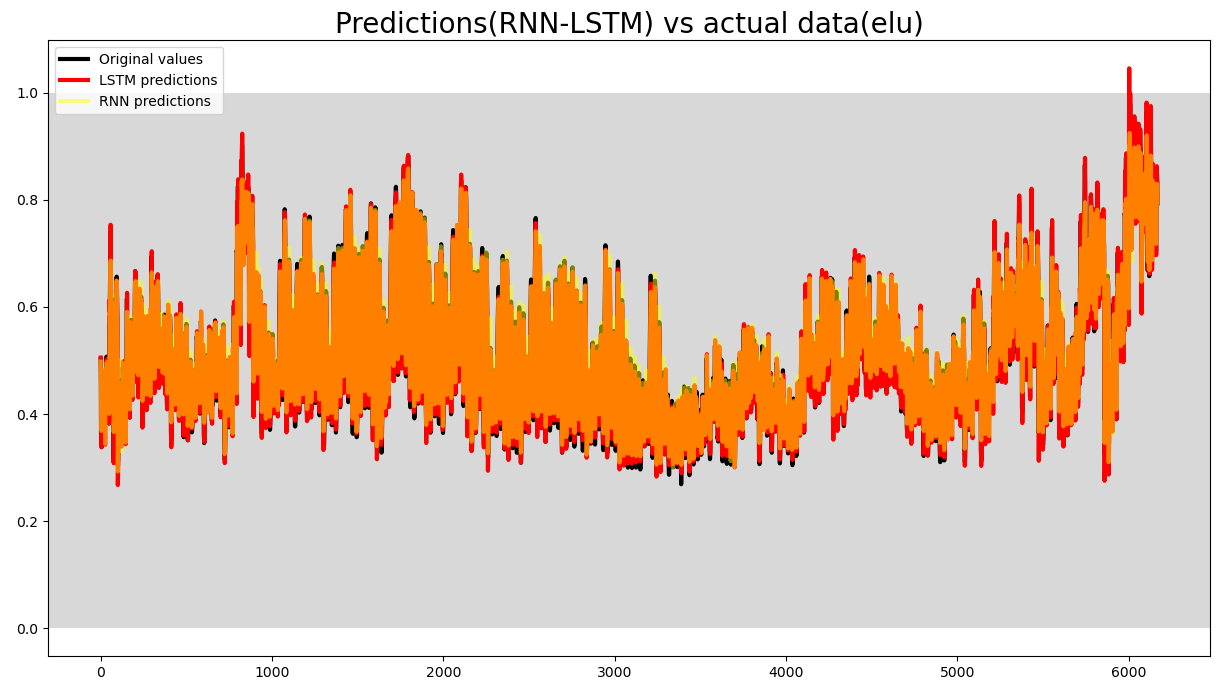


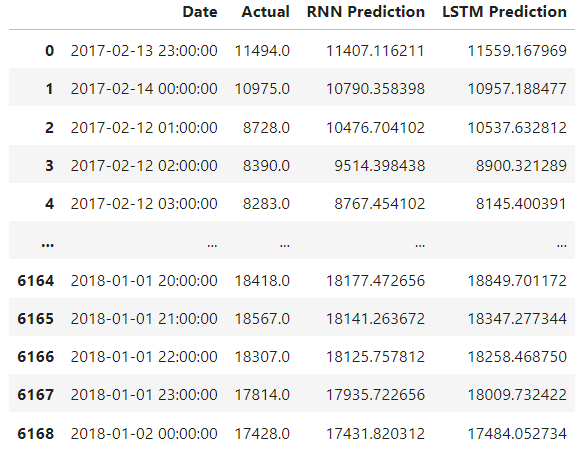
## **ELU**

Here, the RNN and LSTM results obtained by using the activation function elu are shown with graphs, while the prediction values and actual values are given together for comparison.



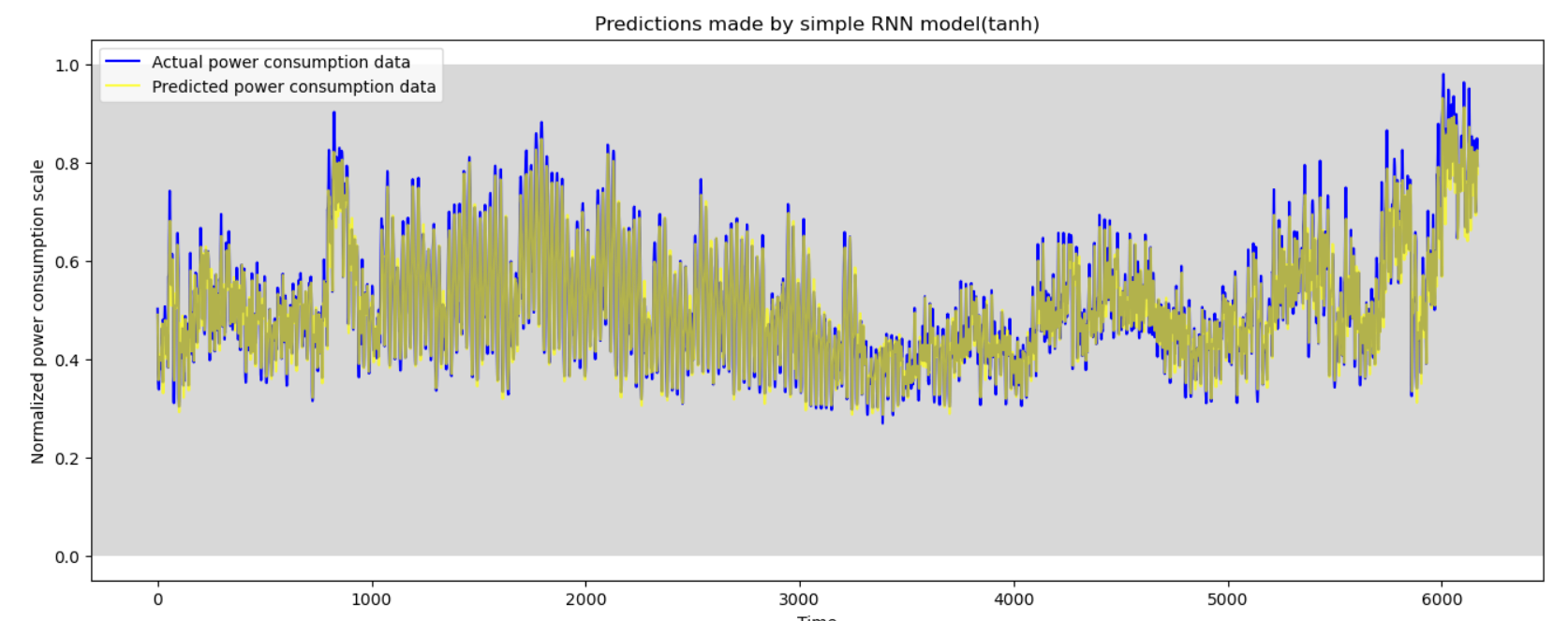


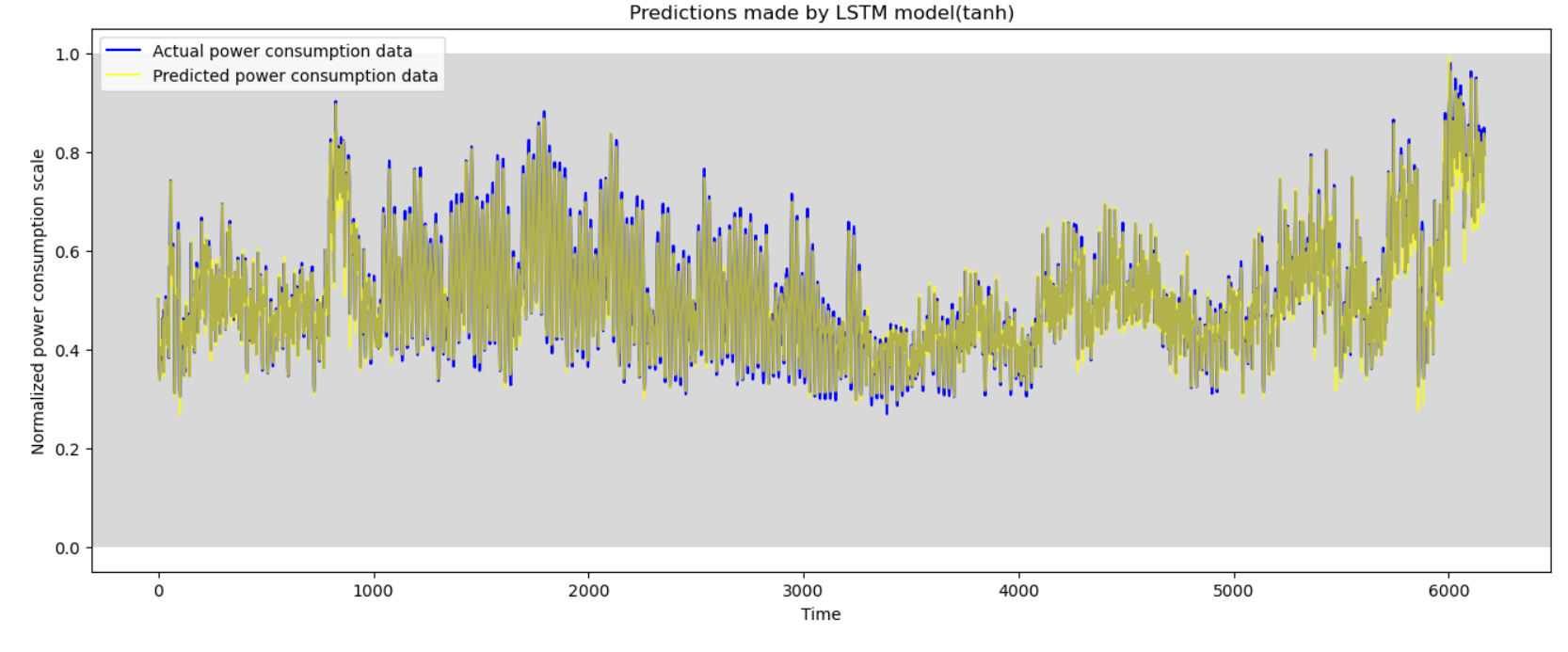


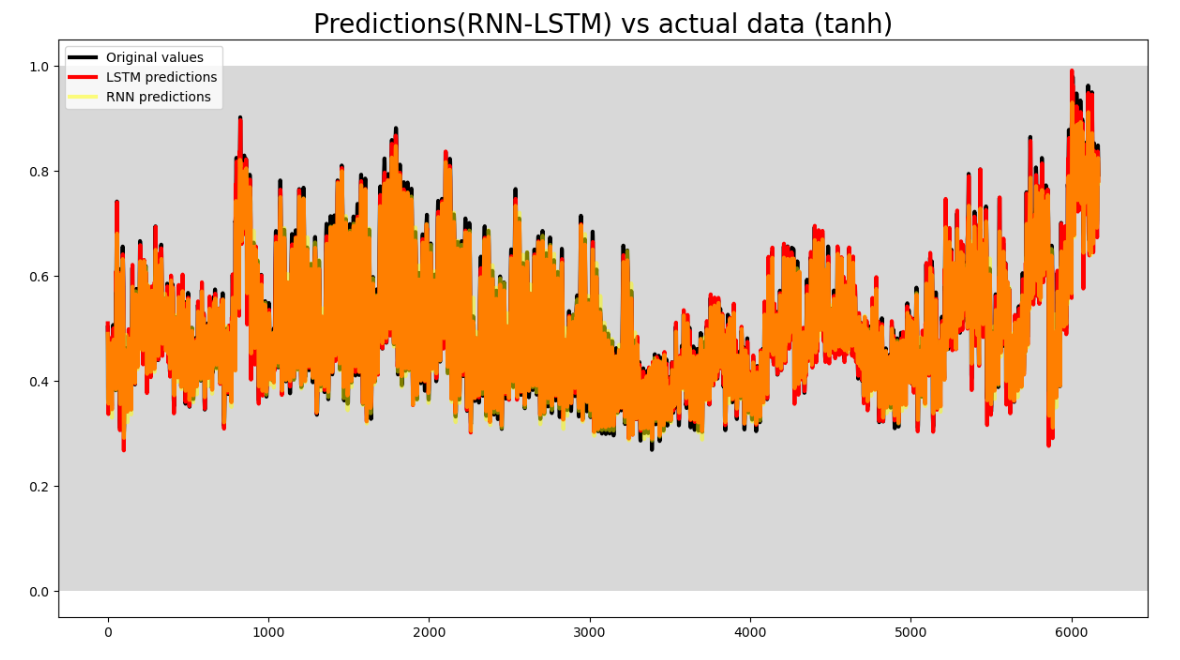


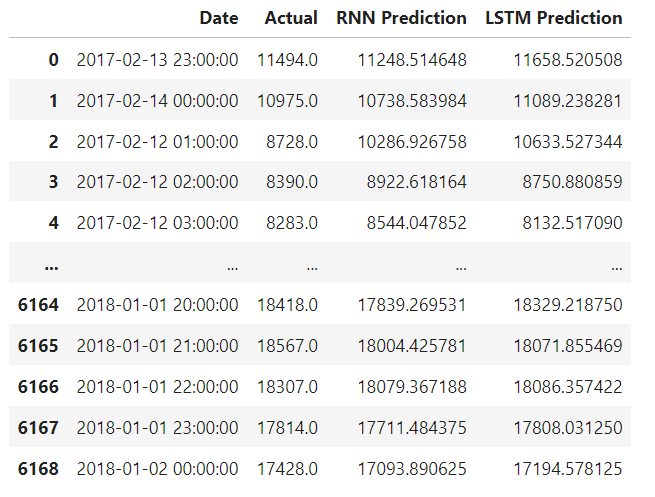
## **TANH**

Here, the RNN and LSTM results obtained by using the activation function tanh are shown with graphs, while the prediction values and actual values are given together for comparison.









When we compared the RNN and LSTM models built in the same architecture with different activation functions at the same parameters, I observed that the LSTM model made better predictions than RNN. Again, while the r2 score gave the highest score with the tanh function in the RNN model, similar results were obtained with tanh and elu functions in the LSTM model. Relu activation function lagged behind the others in both models.

# **5. Conclusions**

Different activation functions were used in the RNN and LSTM models, which were built in the same architecture and organized with the same parameters. the reasons why the tanh function gives better results than relu and elu may be as follows.

* Activation Functions and Performance Differences in Electricity Distribution Company Data.
* Activation functions are critical components that affect the performance of neural networks.
* The preference for the tanh activation function in RNN and LSTM models tends to better preserve past information, especially in LSTM layers .
* The tanh function produces an output that is more symmetric with respect to negative and positive inputs, whereas ReLU and ELU can produce zero output by focusing on positive inputs. This can especially help to better model the variations of energy data over time.
* The advantage of Tanh is that it shows a more sensitive response at values close to zero, which is especially useful for time series data.
* ReLU and ELU generally perform better on large data sets and deep neural networks, but special cases in energy data may cause tanh to be more effective .