Artificial Neural Networks and Deep Learning
Homework #2

(Fall 2023)

Team: **Perceptron Playoff**
Members: **Mohammad Amiri (10887256)**
Sara Limooee (100886949)
Dorsa Moadeli (10926114)
Mohamed Shala (10871548)

Politecnico Di Milano

Checking outliers:

As part of our exploratory data analysis, we implemented a method for detecting outliers within our time series data. Outliers, being abnormal or atypical observations, can profoundly impact the predictive performance of time series forecasting models. Therefore, identifying and understanding these outliers is a crucial step in our analysis process.

We utilized the Interquartile Range (IQR) method, a statistical technique well-suited for outlier detection in time series data. The IQR, calculated as the difference between the 25th percentile (Q1) and the 75th percentile (Q3) of the data, helps in determining the variability in datasets. In our approach, outliers are defined as those observations that fall below **Q1 - 1.5 * IQR** or above **Q3 + 1.5 * IQR**.

We applied this method across each time series in our training dataset, specifically focusing on the valid periods of data. This targeted application ensures that our outlier detection is both accurate and relevant, considering only the data points that are pertinent to our analysis.

Following the identification of outliers, we quantified them by counting the number of outliers in each time series. This quantification was then organized into a Data Frame for better visualization and interpretation. The Data Frame, titled **outliers_df**, presents a clear mapping of each time series against its respective count of outliers.

In our analysis, we visualized outliers in time series data using a custom function, **plot_outliers**, which overlays red scatter points on a line plot of each time series. This approach effectively highlights the outliers, providing immediate insights into their distribution and impact. The visualization of the first time series in our dataset demonstrated the utility of this method, allowing for quick identification of anomalies and informing subsequent data preprocessing decisions.

Using the **plot_series_before_after** function, we visualized the original data alongside its transformed state, applying either robust scaling or winsorization. Robust scaling adjusts data based on quartile range, effectively normalizing it, while winsorization limits extreme values.

Statistical Analysis:

We focused on determining the most common sequence length within our time series data. By calculating and analyzing the lengths of valid sequences, we identified the predominant sequence length, along with its frequency. Additionally, we computed basic statistical measures, including the average, median, minimum, and maximum lengths of these sequences. This analysis provided essential insights into the typical structure of our data, facilitating informed decisions in subsequent stages of data processing and model selection.

The histogram displayed visualizes the distribution of sequence lengths within our dataset, evidencing a predominant abundance of shorter sequences. The most common sequence length stands out at 51, as marked by the blue dashed line. This contrasts with the average sequence length, which is 198.30, depicted by the red dashed line, indicating the influence of longer sequences in the dataset. The median sequence length is observed at 184, shown by the green dashed line, offering a central tendency measure less susceptible to the skewing effect of outliers.

Stationary Status Checking:

Since the data is time series, we must check if it is stationary or non-stationary. Being stationary has significant impacts on the reliability and effectiveness of time series analysis and the models used. For this purpose, there are several methods.

- ACF and PACF Plots: ACF examines how a point in a series relates to earlier points, showing a quick drop in a stationary series. PACF focuses on the direct connection between a point and its past, dropping quickly in a stationary series.
- Plotting Rolling Statistics: This method involves calculating moving averages or moving variances over a certain time window and plotting them against the original time series. If the mean and standard deviation remain relatively constant over time, it suggests stationarity.
- The Augmented Dickey-Fuller (ADF) test checks if a time series is stationary. If the p-value is below a significance level, it rejects the idea of a unit root (indicating non-stationarity) and suggests the series is stationary.
- Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test: The KPSS test has the null hypothesis that the data is stationary around a deterministic trend. If the p-value is less than a predefined significance level (alpha), the null hypothesis is rejected, indicating non-stationarity.

After using the methods to check for stationarity, we had to change the data from nonstationary to stationary. We can use three approaches:

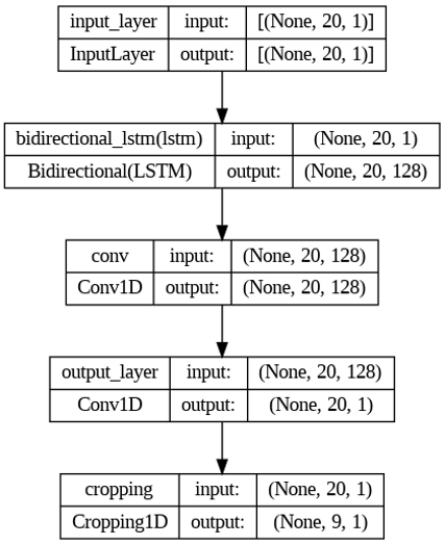
- Removing Rolling Mean (Differencing):
  One method involves subtracting the rolling mean from the original data. The rolling mean is calculated over a specific window, and this process helps eliminate trends and seasonality, making the data more stationary.
- Exponentially Weighted Moving Average (EWMA):
  This technique assigns different weights to different observations, giving more importance to recent values, helping in capturing trends, and making the data more amenable to modeling.
- Decomposition:
  Decomposition involves breaking down the time series into its components. Once separated, the trend and seasonality components can be removed, leaving behind a more stationary residual series.

The last two techniques are the ones we used to make time series data stationary.

Model Development:

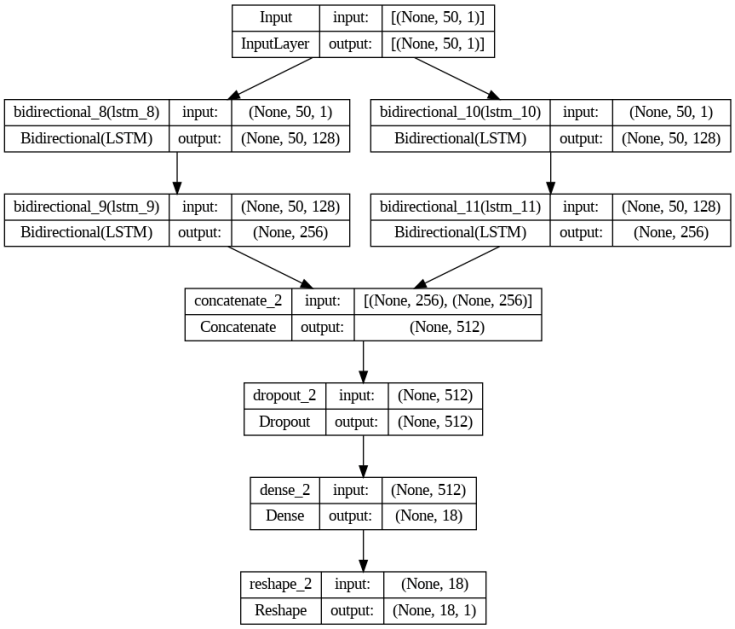
To tackle this problem, we use two main models and try to change the parameters of each to find the one that gives us the best result. Firstly, we must prepare the data to feed the models described in the following. So, we defined the build_sequences() function. It creates sequences of data with a window size of 50 and a stride equal to 10.

The first model has the below architecture:



The model starts with an input layer that specifies the input shape. Then we add the Bidirectional LSTM layer. Bidirectional layers process the input data in both forward and backward directions which means a whole greater understanding of the data. Following the LSTM layer, two 1D Convolutional layers are introduced that help in learning local patterns in the time series data. Finally, the last layer is the Cropping layer which adjusts the output length to match the desired output shape. We calculate the crop size based on the difference between the current output length and the desired output length.

The second model has an architecture as below:



The second model is a concatenation of two separate blocks, each of them is made of a Bidirectional LSTM layer. The concatenated output is later passed to a Dropout layer and a Dense layer. Then, to adjust the output length to match the desired output shape, we add a reshaped layer as the last layer. The reshape layer ensures the consistency of data with the target prediction dimensions. In other words, “Cropping 1D” is used for removing specific time steps from a time series, while “Reshape” is used for changing the overall shape of the tensor without discarding elements.
In the table below, you can see the results of models trained for this problem:

Model Name	Layers	Process	MSE
Model 1_1	Bi-directional LSTM+ConV1D + Cropping 1D	Raw data	0.0103
Model 1_2	-	Raw data + Removing outliers type1 by winsorizing	0.0111
Model 1_3	-	Raw data + Robust normalization + Removing outliers1 by winsorizing	0.136
Model 1_4	-	Raw data + Removing outliers type 2	0.0106
Model 1_5	-	Stationary Raw data	0.0236
Model 2	Bi-directional LSTM + Bi-directional LSTM + Concatenate layer + Dropout	Raw data	0.01201

Contributions:

Our collaborative effort on this project has been instrumental in achieving significant milestones, and each team member has played a distinctive role. Here is an overview of individual contributions:

Mohammad:

- Primarily focused on Exploratory Data Analysis (EDA) by addressing various aspects, including outlier detection, checking stationarity, and making time series stationary.
- Collaborated with Sara to establish the initial structure of the model.
- Took the lead in merging diverse code segments, ensuring seamless integration, and proactively resolving issues arising from different sections.

2. Sara:

- Took a prominent role in the modeling phase, exploring a spectrum of models from basic to advanced.
- Worked closely with Mohammad to develop the initial model structure.

3. Dorsa:

- Spearheaded the construction of the sequence part and contributed significantly to the development of the second model.
- Assumed a major role in crafting and compiling the main sections of the project report, alongside Mohamed.

4. Mohamed:

- Contributed to specific aspects of the EDA process, including analyzing data distribution, winsorizing, and robusting.
- Provided valuable support to Dorsa and Sara in the report-writing phase.