In the following we will describe what we did in the data preperation step. The very first step was to scrape our data from the Internet. We received the dayahead-price and the intraday-price as well as the data about the consumption for the current day. The data was scraped from the site "https://www.epexspot.com/de/" market data. It should now be mentioned that the intraday data was available within 15 minutes and the dayahead data within 1 day. In addition to these dates, the German holidays were also scraped from the page "https://feiertage-api.de/api/?jahr=". With these data it can be determined if the prices and especially the price\_premium, which was calculated later, is dependent on holidays or not. In our dataset the holidays are reflected by a boolean value, which indicates whether the day in question was a holiday or not. There are also two other features that indicate how far in front of and behind the day a holiday was. In addition to the holidays, the participants were also scraped to the market for each day. Of course, they also have some influence on the price and the price premium for a certain day. The data for the participants were scraped from the page "https://www.acer-remit.eu/portal/register-download?fileType=XML&euregId=".Last but not least we have scraped various weather data, as the weather also has an enormous influence on the electricity price of the intraday and dayahead market. The data was scraped from the site "https://www.weatheronline.de/Deutschland/" for all available German locations and then averaged for each day. From these weather data many different features with the names daily average temperature in Celsius, daily maximum temperature in Celsius, daily minimum temperature in Celsius, number of freezing days, number of icy days, monthly rain volume in mm, number of rainy days, the average of daily sunshine hours, the average monthly wind speed in km/h, the monthly snow days and the average of the daily snowfall in cm have developed for us. All the features are summarized so that they can be displayed on a monthly level, so that a value stretches over the whole month. The next step was the interpolation of the data. All data had to be adjusted to the same 15-minute granularity. For this a series with a 15-minute granularity was created, which was later connected as a dataframe with the actual data by a left merge via the respective datetime features to a dataframe. The result of this merge was that we created a dataframe with a 15-minute granularity, but the interesting features are only filled for a full hour. This was solved by the forwardfill and we get our 15-minute single dataframes for the already mentioned scrapped data. Next, then all the still unrelated scraped files could be merged into one large file via their datetime column. Also, the price premium was added, which was calculated by the difference of the dayahead price and the intraday price at time t. The following formula better represents this calculation:

To fix notation, let *Fit* denote the electricity dayahead price observed on day *t* for delivery during hour *i* of day *t* + 1, and let *Si*,*t*+1 denote the intraday price for hour *i* of day *t* + 1. This data set described is used for the descriptive tasks of our group. For the predictive tasks of our group, the dataset has been further edited. This editing is described below.

Recurrent Neural Network:

For our predictive task we have tested different models to compare them with each other. Here the neural net receives an input, exactly like with a normal neural net. In addition, as with a neural network, various complex calculations are carried out in the network and we finally get a prediction for a certain input. First the weights at the knots must of course be trained. The special thing about a recurrent neural network is that the previously calculated values still affect the future values. This is especially useful in our case, because we are looking at values at a certain point in time, which affect the future points in time. For our RNN we first and foremost have the data we need for a prediction into a supervised learning problem. It's about copying the target feature and if we want to shift from time t to t+1 predicting by one to the top, just like we do. For example, if we want to predict from time t to t+2, we have to copy the time variable and shift it upwards by two values.