To what extent will the Covid-19 pandemic contribute towards reaching goals stated in the Paris Climate Agreement for Germany?

Data Analysis Pipeline - Group 10

Florian Auinger 03659767
Florian Butsch 03663488
Florian Hölzl 03659355
Christoph Miller 03721920
Johannes Gensheimer 03721520
Johannes Kiechle 03735507
Johannes Gahr 03675464
Constantin Nowak 03676376

September 4, 2020

Submission structure

Uploaded on Moodle for this submission:

- This main submission document, which gives an overview of the models and approaches that have been tried and an initial assessment. Furthermore, it shows a mock-up of the front-end.

Provided in LDV GitLab repository under *deliverables/milestone3*:

- Implementation of mapping models:
  MappingModels.ipynb
- Implementation of emissions to emissions forecasting models:
  emissions_emissions_forecast.ipynb
- Implementation of features to emissions forecasting models:
  features_emissions_forecast.ipynb

Contents

1 Updated Approach

Our general approach stays the same as in Milestone 2. We use CO2 emission values for Germany until 2030 from the Bundesumweltamt and compare them to the data generated by our machine learning models, reflecting the impact of the COVID-19 pandemic. We will give a reasonable estimate on the emissions during and after the crisis based on current data. Additionally, we develop a lower and upper bound to give further room for discussions.

In order to accurately measure the impact of the pandemic, we can not only look at changes in recent CO2 emission data from during the pandemic but we also have to take into account possible future discrepancies. However, for a complex topic such as CO2 emissions which have a lot of different influencing factors, it is very challenging to conduct a valid forecast. As this uncertainty increases with a larger forecast horizon, we decided to limit our forecast to the end of 2020. Under the assumption, that with possible vaccines against COVID-19 and other counter-measures, the year 2020 will be the worst of the pandemic, we should be able to make an approximation on how the pandemic had an impact on reaching the climate goals for 2030.

Our main problem when it comes to predicting the impact of the COVID-19 pandemic on CO2 emissions is the lack of granular and recent data for the emissions itself. In order to solve this, we try out two approaches with varying complexity which will be explained in the following.

For our first approach, we use the feature data we collected in Milestone 2 and train a model to map their impact on CO2 emissions. As the feature data is more recent, we can use the mapping to get a reasonable approximation of the CO2 emission development during the pandemic so far. Afterwards, we take the previously generated CO2 emission data for the beginning of 2020 and use a simple forecasting model to forecast the emissions until the end of the year.

For our second approach, we directly use our feature data as the input to our forecasting model without mapping them to CO2 emissions beforehand. The forecasting model will then be trained to output the emissions for the end of 2020. This saves us one additional step in between and thus also additional uncertainty originating from the mapping. However, this approach also requires a more sophisticated model and it can thus be difficult to get reasonable results during the short span of this project.

The models we looked at for possible use in the two approaches will be compared in the following sections.

2 Reflection of Previous Milestone

2.1 Sub-sample CO2 data

Problem statement: Usually CO2 emission inventories are provided in yearly temporal resolution. For example our contact at the Öko-Institute states that he won't give monthly CO2 emission estimates due to the high inaccuracy. In order to generate more samples we came up and tested different approaches, e.g. just dividing the yearly values by 12 and assign the same value to each month (no seasonality, mapping won't work), using a Gaussian Regression approach to subsample the CO2 data (this method is purely data-driven and no real seasonality is taken into account).

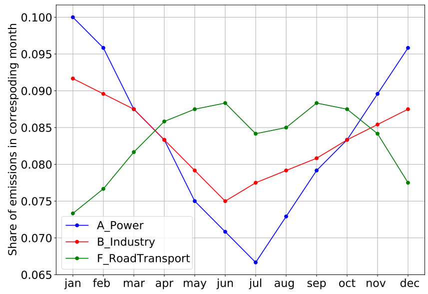


Figure 1: Seasonality of used sectors from TNO_MACco III

Solution to overcome problem: We've then asked Prof. Jia Chen from the Professorship of Environmental Sensing and Modeling at TUM what she can recommend. According to her TNO GHGco [1] is the most accurate and most accepted GHG and CO2 emission inventory used in environmental research. TNO-MACco III is the most recent release which also include three temporal profiles of CO2 emissions: month in the year, day in the week, hour in the day. Using these temporal profiles it is theoretically possible to sub-sample the yearly emissions to hourly emission values. For our approach monthly CO2 emissions are sufficient.
Inaccuracies come as the TNO-MACco III data is only publicly available for the year 2015, resulting in the temporal profiles are used to sub-sample the CO2 emissions from the Umweltbundesamt [2] in order to get a time series. As there are different categories in the emissions of TNO-MACco III and the Umweltbundesamt the assignment between the categories is done in the following manner. Also in the sector mobility the sectors F_RoadTransport, G_Aviation, and H_Shipping have to be merged. As the temporal profile of G_Aviation, and H_Shipping is constant we use the temporal profile from F_RoadTransport for the sector mobility.

Umweltbundesamt	**TNO-MACco III**
Energy and Household	A_PublicPower
Economy	B_Industry
Mobility	F_RoadTransport

Table 1: Assigning categories for the temporal profiles

2.2 Feature merging

Due to the investigation of the temporal coverage of the collected data different data sources needed to be merged in order to get continuous features. Please also refer to the Milestone 2 submission for exact information about the datasources.

2.2.1 Vehicle traffic feature

Problem statement: The main feature of the vehicle traffic sector is the traffic count data of BASt [3] (Bundesanstalt für Straßenwesen) which covers the time span of 2003 until 2018. Traffic count data for Bavaria from BAYSIS [4] (Bayerisches Staatsministerium für Wohnen, Bau und Verkehr) provides data from 2017 until April 2020. Also there are several mobility reports due to the Corona crisis available online (Google [5], Apple [6], TomTom [7], and BASt [8]). The BASt mobility report provides weekly values of traffic reduction for the months March, April, May, and June 2020 which relate to the traffic of February 2020. Also it was needed to separate different vehicle classes in order to estimate the traffic CO2 emissions more accurately.

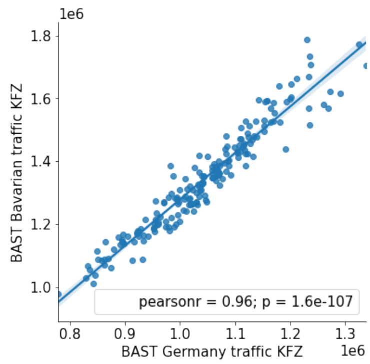


Figure 2: Correlation Bavarian and German traffic for KFZ

Solution to overcome problem: The traffic count data of BASt and BAYSIS and the mobility report of BASt are merged to a continuous traffic feature. The classes LV (light vehicle) and SV (heavy vehicles) are separated. The feature KFZ equals the equation KFZ = LV + SV.
Of all available stations in one month the average is taken which represents the average total number of vehicles per traffic counting station in the corresponding month. This value is independent by the number of counting stations and therefore enables a comparable time series.
The Bavarian traffic of BAYSIS is mapped with a Linear Regression to the German traffic of BASt. Then the BASt mobility report is used to estimate the traffic in the months April, May, and June 2020. Please also refer to the Jupyter Notebook M_CreateVehicleTrafficFeature.ipynb in the dataprocessing folder of the git repository for more details about this method. The applied steps result in three continuous features for the time span of 01/2003 until 06/2020: M_KFZ, M_LV, and M_SV where M relates to the sector mobility.

2.2.2 Aviation feature

Problem statement: On the one hand, the aviation data from the GENESIS database are not available until June inclusive, and on the other hand there are six different features with identical or irrelevant information content.

Solution to overcome problem: The number of take-offs and landings are summed up and form a representative feature that reflects the development of flight movements. In order to complete the data set by June, the data set from the GENESIS database [9] is merged with the data set from the OAG database [10]. Figure 3 shows the correlations of the two data sets and is therefore a proof that this step is valid. The OAG data set cannot be used alone because it only covers the years 2019 and 2020. If the GENESIS data is available until the final delivery by June, this step is no longer necessary.

Figure 3: Correlation between Aviation data from GENESIS database and OAG

2.2.3 Energy features

Problem statement: The electricity data from the GENESIS [11] database is available from 2002 until 2020. However, it is only complete at the beginning. For instance, data on wind energy cannot be found in later years.

Solution to overcome problem:

For this reason it is supplemented with data from the Information platform of the Federal Network Agency on the German electricity market "Smard.de" [12]. The data from 2015 onwards will be taken over completely from this database. Data before 2015 is not available on this site. This website provides data on the realised electricity generation in Germany every 15 minutes. The data in this fine resolution is integrated to get the value for each month.

2.3 Adding new data

Starting the modeling part and constantly researching literature the database of the project was adjusted. It became clear that data is the most important part and data collection is an iterative process.

2.3.1 MWV fuel sale data

Overview of the data
Sector: Mobility, Econonomy, Energy and Household
Time span: 2002 until May 2020
Number of samples: 232 (one only 119)
Features: 18
Frequency: monthly
Source: [13]

Why did we choose the data source and how it might help us
CO2 emissions are caused by burning fossil fuels. This counts for every sector.
The fuel sale of

- Diesel, Benzin, and airplane fuel is assigned to the traffic sector.
- Heating oil is assigned to the energy and household sector.
- Raw gasoline (Rohbenzing) and gasoline components (Benzinkomponenten) is assigned to the economy sector.

Data processing notebook: Process_MWV_Rohoel.ipynb
The data is downloaded manually and stored in one folder. The notebook reads all files in this folder and creates DataFrames for each fuel type which are separated in different sectors.

Limitations: E.g. heating oil does not directly correlate to the CO2 emissions of the month where it was bought and the time series has missing data from 2013 until 2018.
The data only reaches until April 2020.

3 Machine Learning Models

3.1 Mapping

As described in Milestone 2, we will build a model that calculates the CO2 emissions under the influence of the COVID-19 crisis. Therefore, we map identified indicators to CO2 emissions until June 2020.

In order to train the model we use a ground truth until 2017 and the estimates until December 2019 from Umweltbundesamt. The indicators date until June in monthly or even more granular frequency. With the trained model we map the CO2 emissions from 2018 until June 2020 but for now not all features are available until June 2020.

We chose five different algorithms for the mapping (MLR, NN, PLSR, Ridge, Lasso), with the aim to find the best performing model for this case. Algorithms like Random Forest and Support Vector Regression are not investigated as they are not able to extrapolate.

The CO2 emission data is available in annual estimates provided by the Umweltbundesamt. We then had the decision to sum up the monthly feature samples to yearly time steps which results in very few samples, or to find a method to get the month-in-the-year temporal profile of CO2 emissions. The best found estimate and approach is using TNO-MACco data like described in section 2.1. The features and the emission data is consistently used in monthly time steps.

We have divided our data into three sectors, namely mobility, energy/households, and economy. To test the performance of the models, we calculated the RMSE as metrics. Furthermore, we plotted the results of the prediction and compared them with the estimations from the Umweltbundesamt.

Linear Multivariant Regression

- In the first part of our model we have the possibility to transform our data to a polynom of higher order which results in a nonlinear regression
- We defined a pipeline with a StandardScaler, PCA and a Linear Regression Estimator. We chose LeaveOneOut as the validation method
- The GridSearch finds the best number of PCA components
- Possible hyperparameter optimization: degree of polynom transform, scaler type, validation method, number of PCA components

Neural Network

- First, we scaled the data with a MinMaxScaler
- Then we split the data into training and validation data
- As a model we chose initially a 2 layer neuronal network with Relu as activation function.
- As metric and loss function MSE was chosen and the optimizer is RMSprop
- Trainable parameters: 29
- Possible hyperparameter optimization: scaler, network size, batch size, optimizer, loss function, different regularization methods

Partial Least Square Regression

- We defined a pipeline with a StandardScaler and a PLSRegression. We chose KFold as the validation methode
- The GridSearch finds the best number of PLSR components
- Possible hyperparameter optimization: scaler type, validation method, number of PLSR components

LASSO

- First, we scaled the data with a StandardScaler
- We chose KFold as the validation method
- The GridSearch finds the best number of alphas
- Possible hyperparameter optimization: scaler type, validation method, alphas

Ridge

- First, we scaled the data with a StandardScaler
- We chose KFold as the validation method

- The GridSearch finds the best number of alphas
- Possible hyperparameter optimization: scaler type, validation method, alphas

First results

- In general the regressions show good results on training data (see Figure 5).
- Sector mobility: Predictions are similar to estimates from Umweltbundesamt and for Corona a clear drop is apparent. The NN estimates the same value independent of the input except for March 2020 (see Figure 4a).
- Sector economy: All the regressions seem to learn wrong correlations. In the corona month there is a peak but not in the intuitive direction (see Figure 4b).
- Sector energy and household: A decrease of CO2 emissions for Corona is visible but it is in the magnitude of the seasonal changes. Only NN shows a exceptional downfall (see Figure 4c).
- All sectors: All regressions show only a small decrease in February and an increase in March which is counter intuitive. More recent data is currently not available for all features (see Figure 4d).
- All regression methods show similar results. In some cases NN deviates from these models as the training of the NN is not deterministic due to random initialization.
- To overcome intuitively wrong correlations a feature selection by hand might be a possible solution.

	Economy RMSE		Mobility RMSE		Energy and Household RMSE		All sectors RMSE	
Model	train	test	train	test	train	test	train	test
LR	0.87	1.15	0.43	0.45	1.58	1.95	1.26	1.63
NN	1.18	1.18	0.88	0.88	1.70	1.70	2.57	2.57
PLSR	0.86	1.13	0.35	0.46	1.58	1.95	1.28	1.59
LASSO	0.85	1.10	0.38	0.44	1.59	1.88	1.27	1.55
Ridge	0.88	1.12	0.36	0.44	1.60	1.94	1.29	1.58

Table 2: Mean RMSE of 10 fold cross validation in million tons CO2. In case of a optimization the results of the best performed hyperparameters are used.

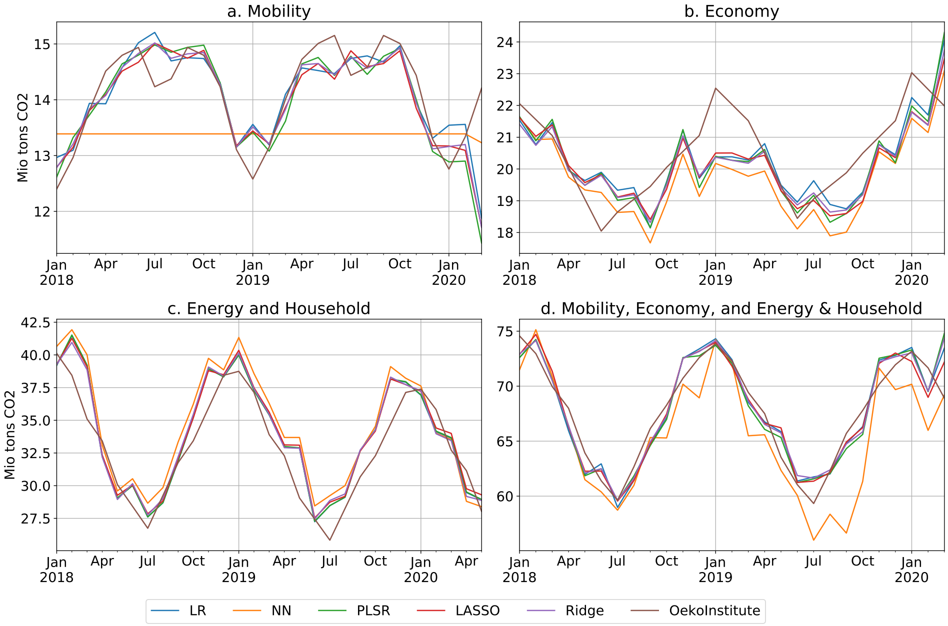


Figure 4: Results of models for 2018 until today

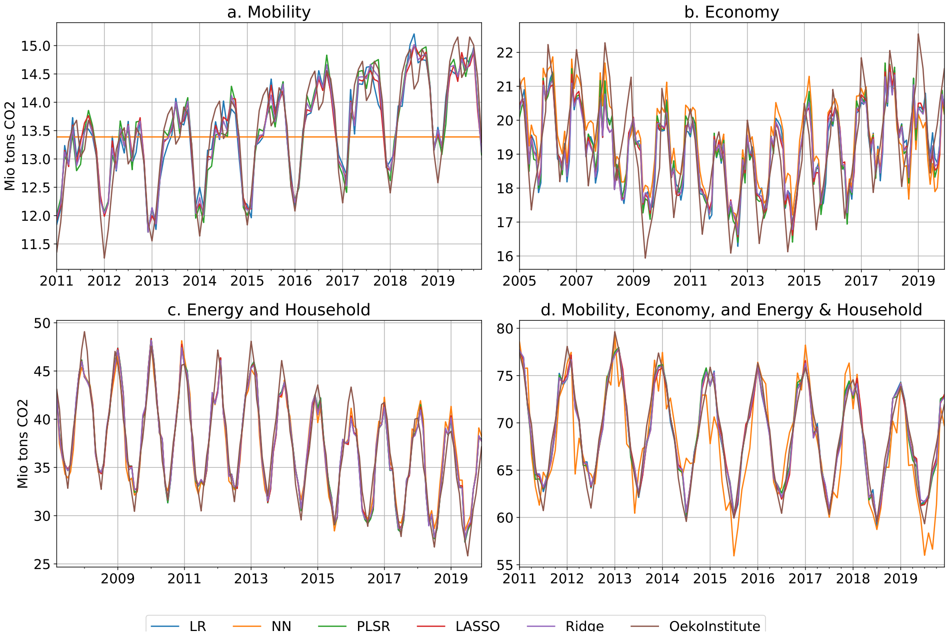


Figure 5: Results of models for training period until 2019

Next steps

- Manual feature selection for each sector that shows intuitively wrong correlations and that are not available until at least June
- Hyperparameter optimization of the models
- Selection of the best model for all three sectors
- In order to map to the overall CO2 emissions there are two possibilities:
  - Map all features to total emission in one model
  - Add the results of the respectively best model for each sector
- Possibly merging of the sectors economy and energy/household

3.2 Time series analysis / Forecasting

The input to all following models, expect for SARIMA, is generated by a sliding window over past data. The windows have a size of 12 months and get shuffled within training and testing set. The forecast based on our windows is done for 6 months. All data is preprocessed with a standard scaler beforehand.

3.2.1 Emissions to emissions forecasting

As shown in Section 3.1, we can use our features to get recent emission data. We can then use the predicted recent emissions as the input to our forecast models to forecast the possible emissions for the rest of 2020. This forecast is rather simple and can be solved easily with a variety of models. To compare different types of models, we chose a regression model with SARIMA and a neural network with the MLP.

Multilayer Perceptron (MLP)

- Input Layer: 1D array - for multiple features, 2D array gets flattened
- Three Dense layers with ReLU activation
- Number of parameters: 690
- Learning Rate: 0.001
- Epochs: 250
- Batch size: 4
- Batch normalization used
- Possible hyperparameter optimization: different learning rates, different scalers, different batch sizes

SARIMA

- Use Auto SARIMA to get p and q values by optimizing over AIC
- Takes all available emission data as input for forecasting
- Includes seasonality in comparison to ARIMA

First results

The implementation of our models with the corresponding results can be found in *emissions_emissions_forecast.ipynb*.

- In general both the regression and the neural network approach show good results on training data (see Figure 6).
- The ARIMA model gave poor results due to the high seasonality in the data and was thus not further evaluated. This was to be expected and already described in the literature about the models.
- A direct comparison of the two approaches can be seen in Table 3 and Figure 7.
- As we will later use data as input which itself was already predicted for the first half of 2020 by our mapping models, we have to take into account, that we have a high chance of uncertainty.

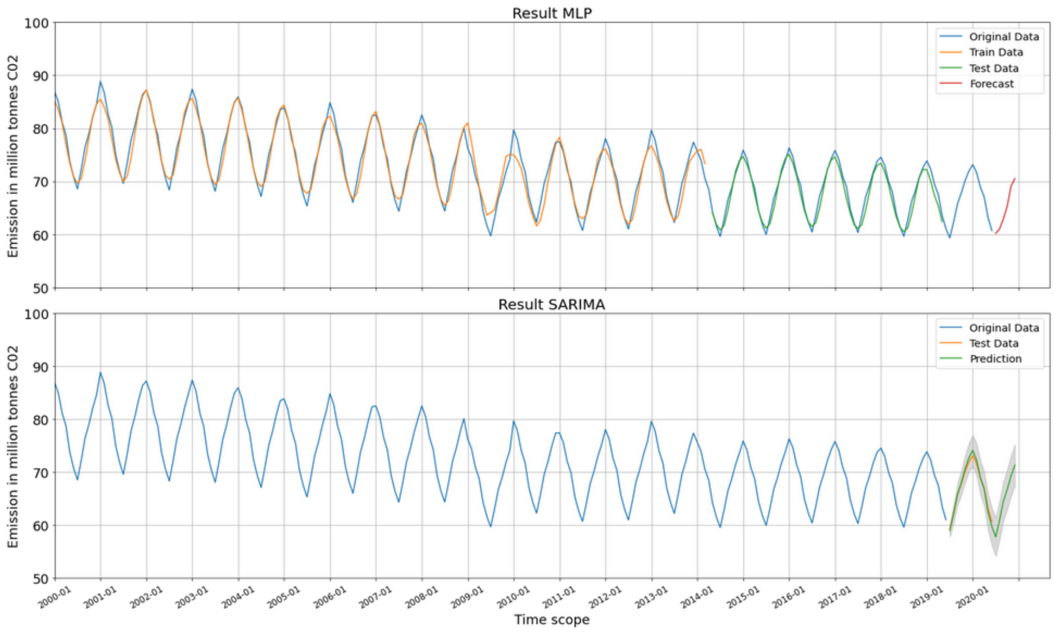


Figure 6: Resulting MLP and SARIMA forecast for monthly CO2 emission data without any COVID-19 information.

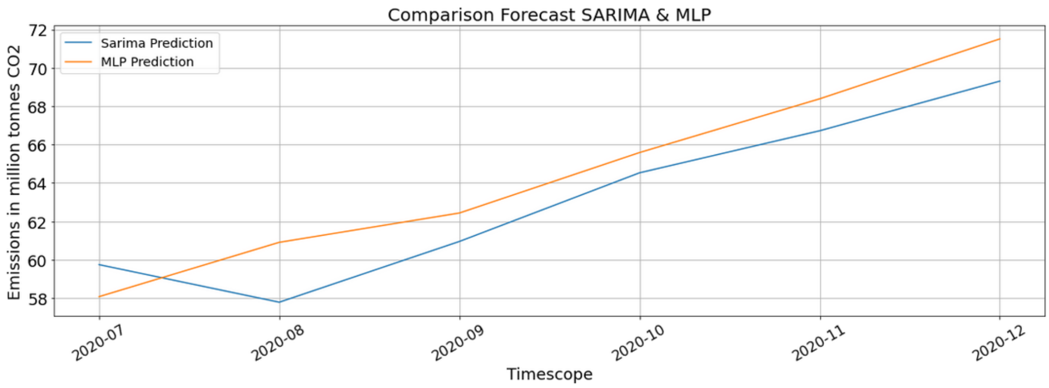


Figure 7: Comparison of the Forecast of the Models

	CO2 Emissions RMSE	
Model	train	test
MLP	2.32	1.74
SARIMA	0.57	

Table 3: RMSE in million tons CO2. In case of a optimization the results of the best performed hyperparameters are used.

Next steps

- Selection of the best forecast model by also comparing the forecast values to the data from the Bundesumweltamt
- Use CO2 data generated by the mapping models to forecast CO2 emissions for the end of 2020
- Compare Mapping + Forecast approach to direct forecasting with the best model from the following section

3.2.2 Features to emissions forecasting

For our second approach, we use our features with recent data and directly predict a forecast of our emissions until the end of 2020. This way, the model not only has to learn the temporal correlation of a single feature but also weights the importance of the features itself. For this approach, a more complex model is required but the results should also be more representable.
For training, we use the Adam optimizer with mean squared error as loss and the R2 metric.

Multilayer Perceptron (MLP)

- Input Layer: 1D array - for multiple features, 2D array gets flattened

- Four Dense layers with ELU activation
- Number of parameters: 167 286

Long Short-Term Memory (LSTM)

- Input Layer: 2 dimensional input with number of features and look back horizon
- Three LSTM layers followed by three Dense layers with ReLu activation
- Dropout Layer: No
- Number of parameters: 559 318

1D-Convolutional Neural Network (CNN)

- Input Layer: 2 dimensional input with number of features and look back horizon
- Three Conv1D layers with ReLU activation
- Dropout Layer: No
- Number of parameters: 74 966

WaveNet (CNN like Neural Network Architecture)

- Inspired by the paper "Conditional Time Series Forecasting with Convolutional Neural Networks" [14]
- Input Layer: 2 dimensional input with number of features and look back horizon
- Conv1D layers with ReLU activation
- Dropout Layer: No
- Number of parameters: 26 630

Used hyperparameters for all models

- Learning Rate: 0.001 (default)
- Epochs: 250 with Early Stopping
- Batch size: 8
- Batch normalization used
- Possible hyperparameter optimization: different learning rates, different scalers, different batch sizes, apply feature selection

First results
The implementation of our models with the corresponding results can be found in *features_emissions_forecast.ipynb*.

- As shown in Figure 9 and Table 4, the WaveNet and MLP outperformed the other models.
- However, Figure 8 shows that WaveNet seems to be more feasible to adopt to our collected data.
- Even though WaveNet is also build on a CNN structure it is more capable of modeling the complexity of our task compared to our vanilla CNN.
- With a look at Figure 10, we can see that LSTM in comparison to our vanilla CNN was more capable but couldn't compete with WaveNet nor MLP.

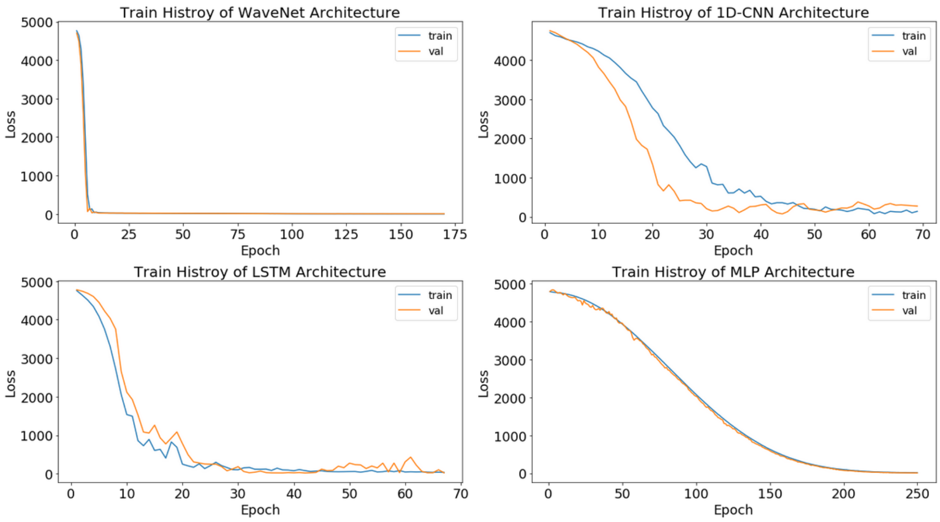


Figure 8: Training history of our direct forecasting models.

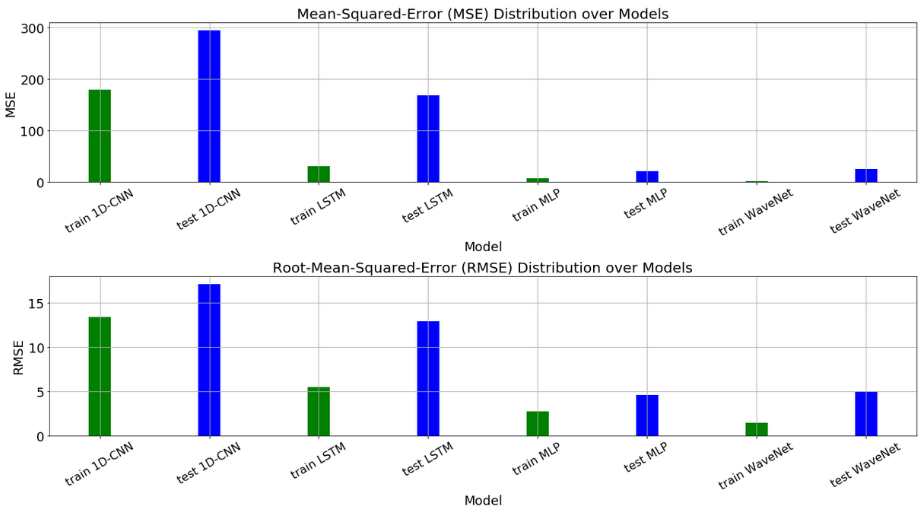


Figure 9: Error Comparison Train/Test for MSE and RMSE among all Models.

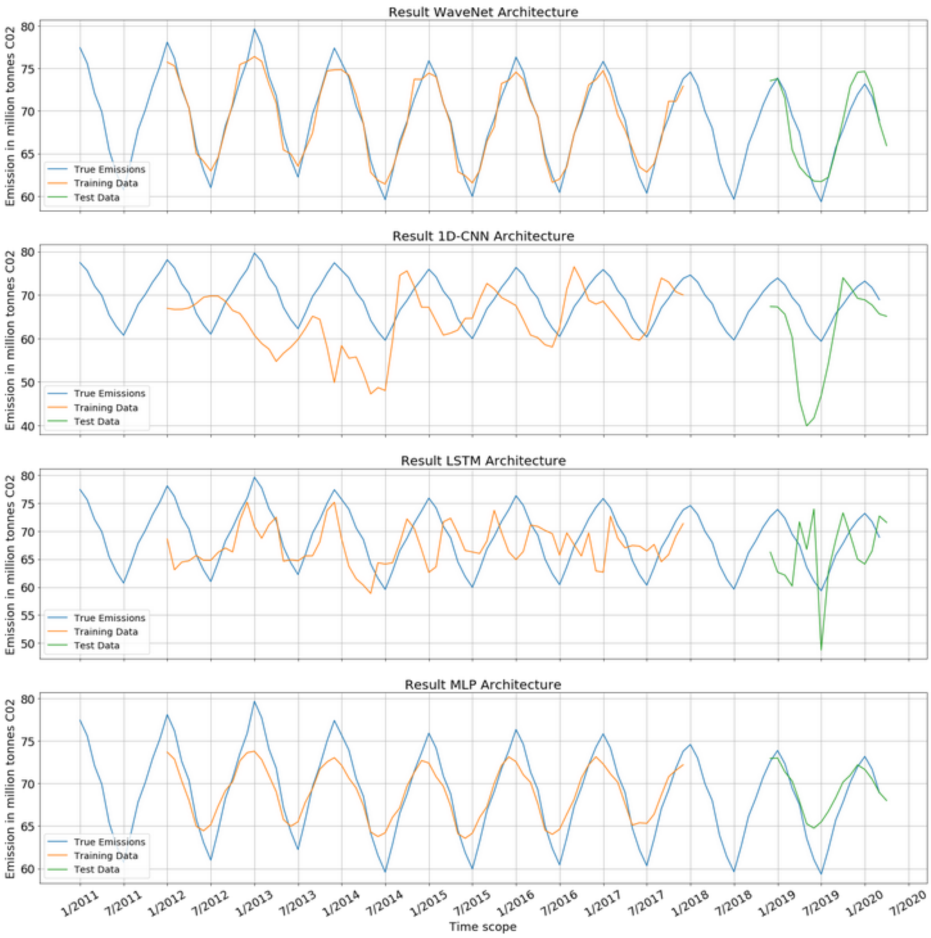


Figure 10: Resulting Performance - Overview Neural Network Architectures.

	All sectors			
	MSE		RMSE	
Model	train	test	train	test
MLP	7.32	21.42	2.71	4.63
LSTM	30.41	168.11	5.51	12.97
CNN	179.79	295.06	13.41	17.18
WaveNet	2.02	24.66	1.42	4.97

Table 4: MSE and RMSE in million tons CO2. In case of a optimization the results of the best performed hyperparameters are used.

Next steps

- Selection of the best forecast model by also comparing the forecast values to the data from the Bundesumweltamt
- Manual feature selection for each sector that shows intuitively wrong correlations and that are not available until at least June (same as in Section 3.1)
- Hyperparameter optimization of the models
- Possibly merging of the sectors Economy and Energy & Household
- Compare features to emissions forecasting approach to emissions to emissions forecasting approach with mapping from Section 3.1

4 Front-End

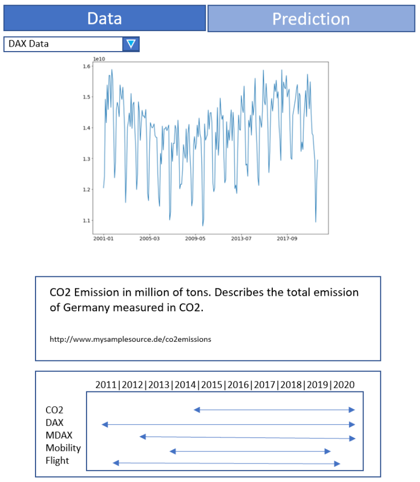
In this section a first design draft for the front-end interface is provided as well as some general information about the software libraries which are planned to be used for its development.

4.1 Goal

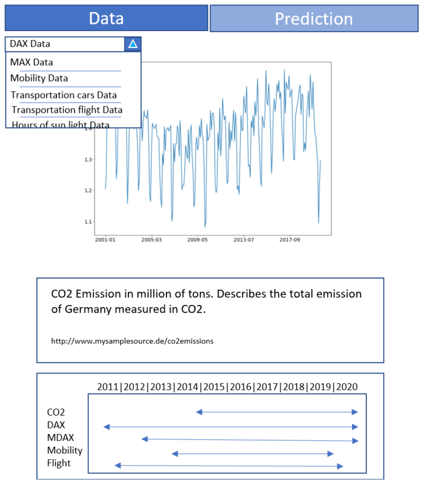
The goal is to develop a front-end that gives the user a graphically appealing illustration of the data calculated with Machine Learning. The user can influence the input of the algorithms and thus feed different data into the analysis. For the Graphical User Interface (GUI) the environment Dash is used. Dash is a productive Python framework for building web applications. Dash is written on Flask, Plotly and React and is perfect for creating data visualization applications with highly customizable user interfaces in native Python. This is especially useful for anyone working with data in Python. Using a few simple patterns, Dash abstracts all the technologies and protocols required to create an interactive web-based application. Dash applications are rendered in the web browser. They can be deployed to servers and then shared using URLs. Because Dash applications are rendered in the web browser, Dash is inherently cross-platform and mobile. There is a lot behind the framework. Dash is an open source library published under the permissive MIT license. Plotly develops Dash and provides a platform for managing Dash applications in an enterprise environment.

4.2 Mock-Up

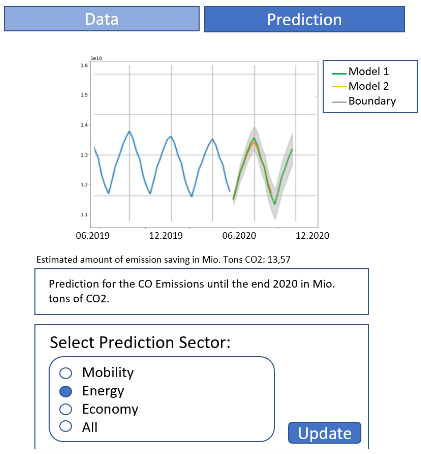
The user interface is split into two main pages/tabs. In the first page the different input data can be selected via a drop-down menu, as can be seen in Figure 11b. The selected data is then visualized and background information as well as its source is provided below (see Figure 11a). At the bottom an overview off all available input data-sets and the time-spans of availability are provided. The second page/tab focuses on the prediction of the trained models, as can be seen in Figure 11c. The predictions are visualized together for easier comparison and some kind of uncertainty region is provided. Below the estimated amount of saved CO2 emissions compared to the predictions without corona is stated. On the bottom of the page the input data segments can be chosen and the visualization updated via an button. Additionally it was planned to provide some parameters which the user can tune to change the COVID-19 infections data for the lower bound calculation (see Figure 11d). From the current point of view, we're not sure if such an simulation can easily be implemented since some doubts about the models have accrued during the latest development phase. Despite this uncertainties this part of the visualization is included for sake of completeness.



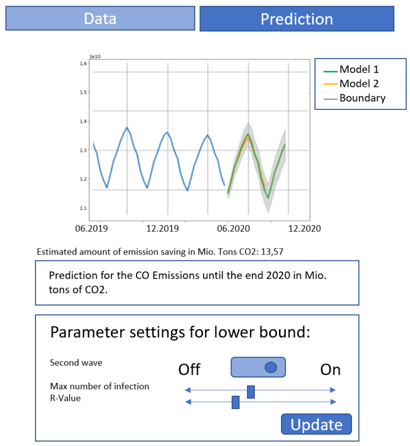
(a) Data can be selected for visualization and background information is provided with it



(b) Example for the drop-down menu which is used to select an input data.



(c) Prediction-frontend page. Models can be compared and input data can be selected for these models.



(d) Prediction-frontend page. Different corona scenarios for the lower bound can be computed.

References

[1] TNO-MAC temporal profile. https://www.che-project.eu/sites/default/files/2019-01/CHE-D2-3-V1-0.pdf. Accessed: 2020-08-11.

[2] Umweltbundesamt CO2 Emissionen. https://www.umweltbundesamt.de/indikator-emission-von-treibhausgasen#die-wichtigsten-fakten. Accessed: 2020-08-11.

[3] BASt traffic count. https://www.bast.de/BASt_2017/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/zaehl_node.html. Accessed: 2020-08-11.

[4] BAYSIS traffic count. https://www.baysis.bayern.de/web/content/verkehrsdaten/dauerzaehlstellen.aspx. Accessed: 2020-08-11.

[5] Google mobility report. https://www.google.com/covid19/mobility/. Accessed: 2020-08-11.

[6] Apple mobility report. https://www.apple.com/covid19/mobility. Accessed: 2020-08-11.

[7] TomTom traffic index. https://www.tomtom.com/en_gb/traffic-index/. Accessed: 2020-08-11.

[8] BAST Corona report. https://www.bast.de/BASt_2017/DE/Statistik/Verkehrsdaten/Verkehrsbarometer.html?nn=1820340. Accessed: 2020-08-11.

[9] GENSIS aviation database. https://www-genesis.destatis.de/genesis//online?operation=table&code=46421-0012&bypass=true&levelindex=0&levelid=1597315214933#abreadcrumb/. Accessed: 2020-08-13.

[10] OAG corona aviation schedule data. https://www.oag.com/coronavirus-airline-schedules-data/. Accessed: 2020-08-13.

[11] GENSIS energy data. https://www-genesis.destatis.de/genesis//online?ope. Accessed: 2020-08-14.

[12] SMARD energy data. https://www.smard.de/home/downloadcenter/download_marktdaten/726#!?downloadAttributes=%7B%22selectedCategory%22:1,%22selectedSubCategory%22:1,%22selectedRegion%22:%22DE%22,%22from%22:1559340000000,%22to%22:1622584799999,%22selectedFileType%22:%22CSV%22%7D. Accessed: 2020-08-14.

[13] MWV fuel sale Germany. https://www.mwv.de/statistiken/mineraloelabsatz/. Accessed: 2020-08-11.

[14] A. Borovykh, S. Bohte, and C. Oosterlee. Conditional time series forecasting with convolutional neural networks. *arXiv: Machine Learning*, 2017.