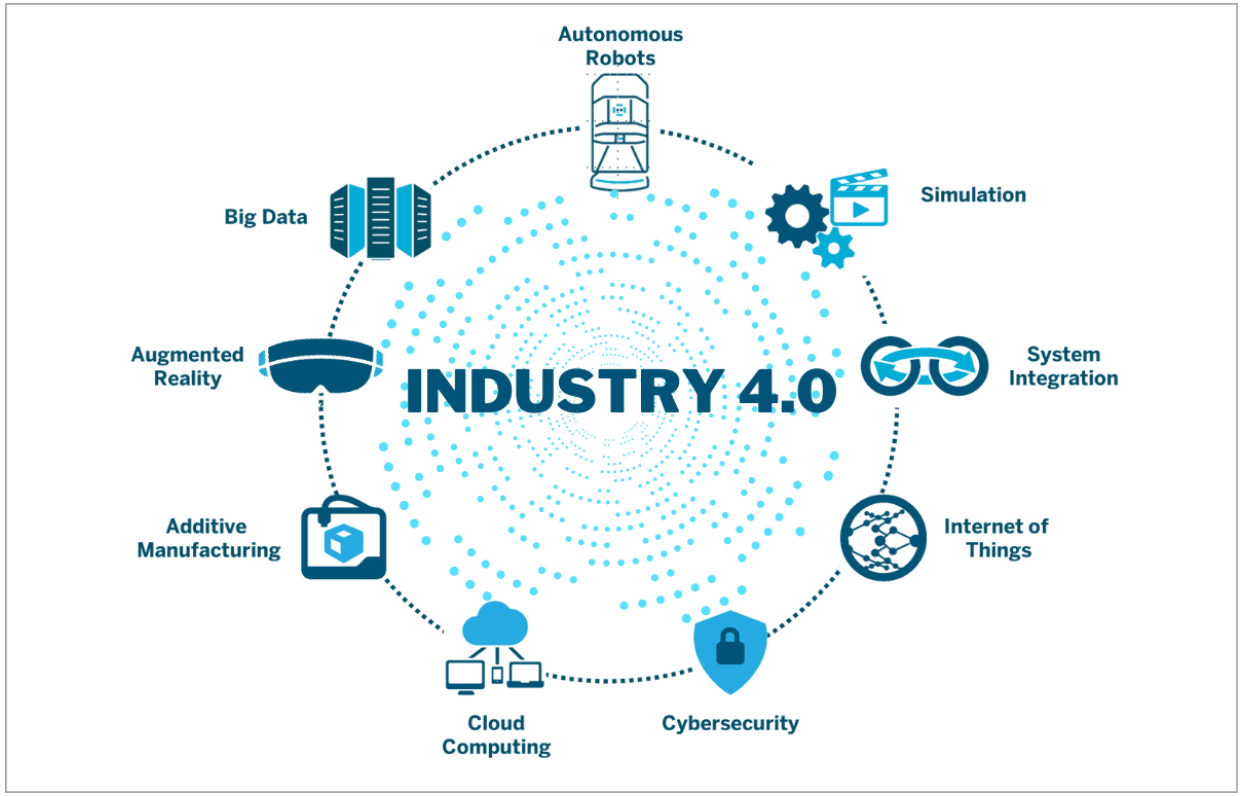
Case Study: Fiat Chrysler Automobiles Time Series Prediction in Welding Process Control

Sanja Stanisic, matr. 800409

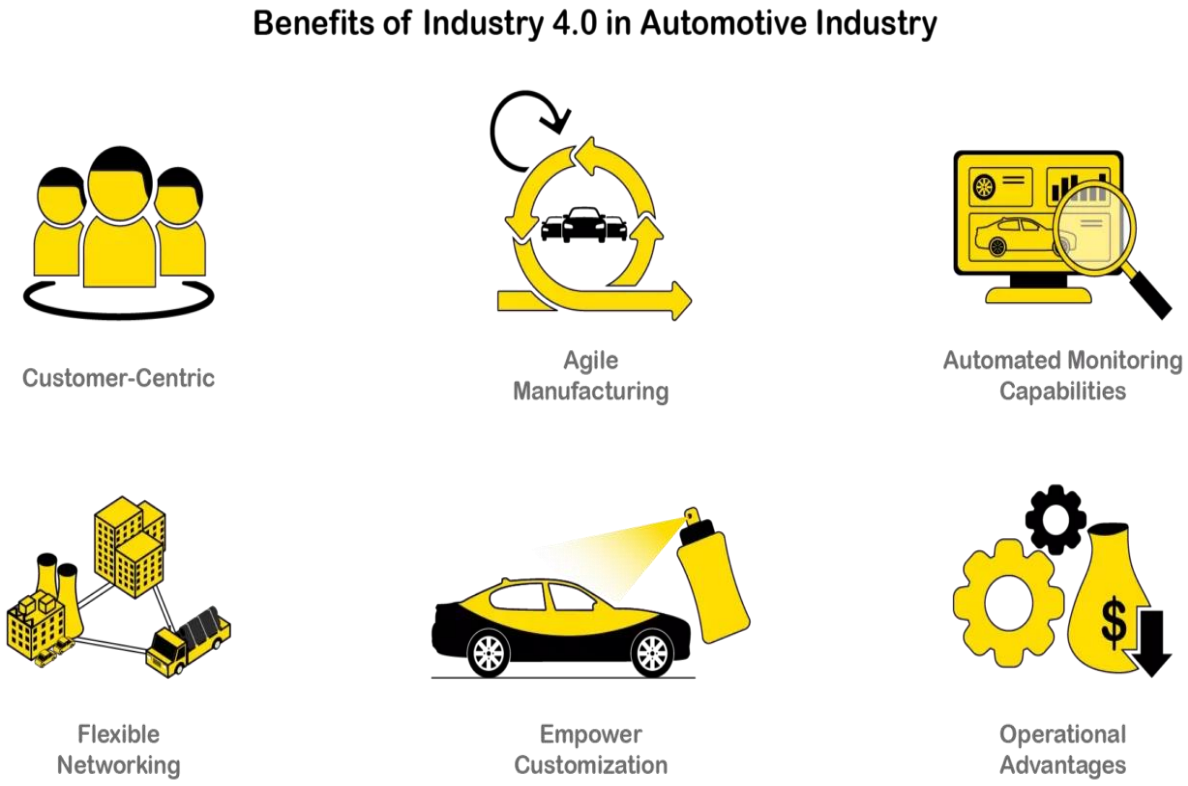
Contents

INDUSTRY 4.0



- Industry 4.0 represents a new phase in the organization and control of the industrial value chain, due to the integration of advanced technologies such as the Internet of Things (IoT), artificial intelligence (AI), cloud computing and big data analytics into the manufacturing process, as well as throughout the entire value chain.
- Products and means of production get networked and 'communicate', enabling new ways of production, value creation, and real-time optimization.

Industry 4.0 in Automotive Industry



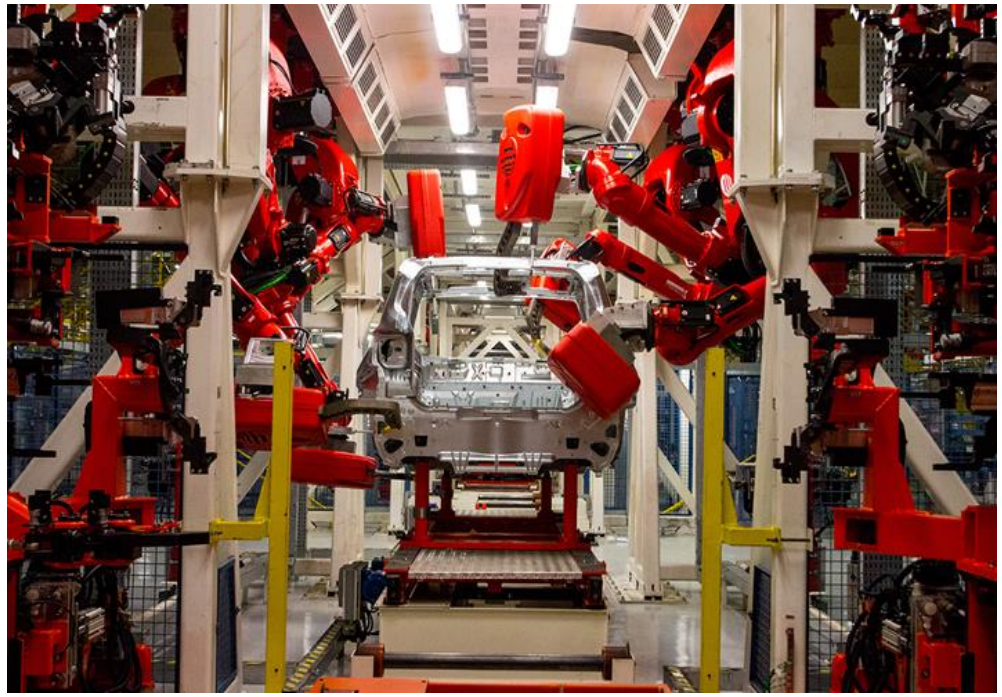
Goals

Primarily, the goal of industry 4.0 in automotive industry is to go beyond optimization and automation by transforming the overall production process to become customer-centric along with efficiency in logistics management.

2022

It was foreseen that by the end of 2022, automotive manufacturers expected **1** in **4** of their plants would be smart factories, and **49%** of automakers have already invested more than **250** million dollars in smart manufacturing.

FCA TIME SERIES PREDICTION IN WELDING PROCESS CONTROL



This case study comes from a production line of a **Fiat Chrysler Automobiles (FCA)** production plant, more precisely from a **welding station n. 005**, where shells of cars come to be assembled.

The heart of the system is the open gate station, a high-density cell where robots surround each car body and weld simultaneously. This robotic welding system is equipped with sensors and other devices that collect data relating to welding processes.

These data are collected by Weld Quality System (WHS), whilst Weld Management System (WMS) generates welding curves of a specific process. The data are subsequently collected and stored in a file system enabling their analysis aimed at identifying patterns and trends that can be used to optimize the process, reduce waste, and improve product quality

PROJECT GOAL

- This project aims at predicting the value of the voltage in different moments of time using time series.

- Voltage values bellow/above the acceptable range suggest a possible anomaly. Being able to predict the voltage therefore is of high importance in anomaly detection, which contribute significantly to real-time control of the welding process.

Case Study: Fiat Chrysler Automobiles
Time Series Prediction in Welding Process Control

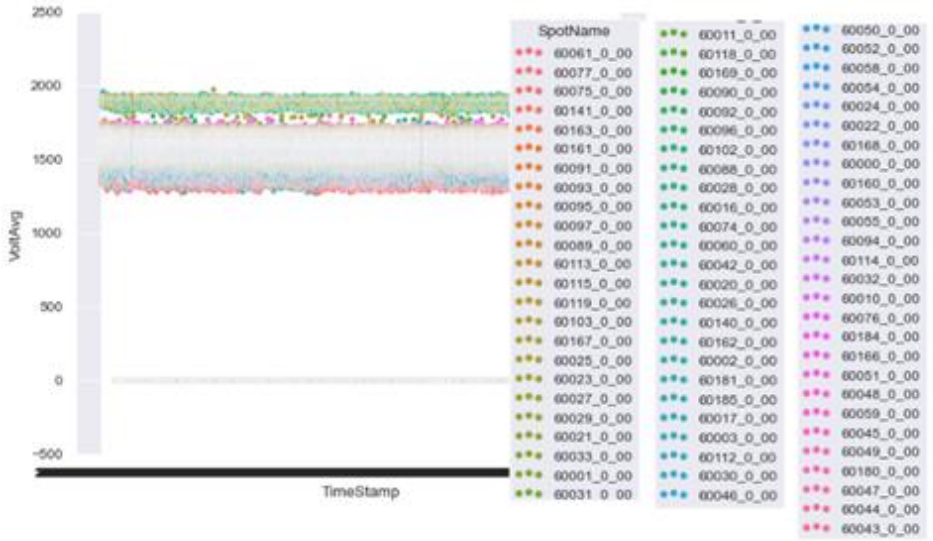
- The data provided are **222866** **.json** files extracted from WMS over 2019 and 2020 (September, November and December of 2019, and January, February, March and May of 2020).
- Every extracted .json file contains **4** **features**: Time Stamp, Spot Name, Current Curve and Voltage Curve.
- At the station n. 5 data are generated every millisecond after the initial welding time at **76** welding spots.
- The **first step** of data pre-processing was to collate all .json files in a single file of the same structure as original files, and save it as a .csv file.
- Due to the large amount of data, a special data set was prepared for the develop mode – i.e. for the trials, prior to running the code on the entire dataset. The appropriate dataset was saved in the file system after every pre-processing step.
- Further analyses have shown that the data were extracted from only **60** days over a period of **8** months
- It was checked whether the data have been registered on Sundays and, as well, at what time the registration of data had begun and ended.
- Correlation between the welding spots and the date was checked, and it was insignificant (0,000055).

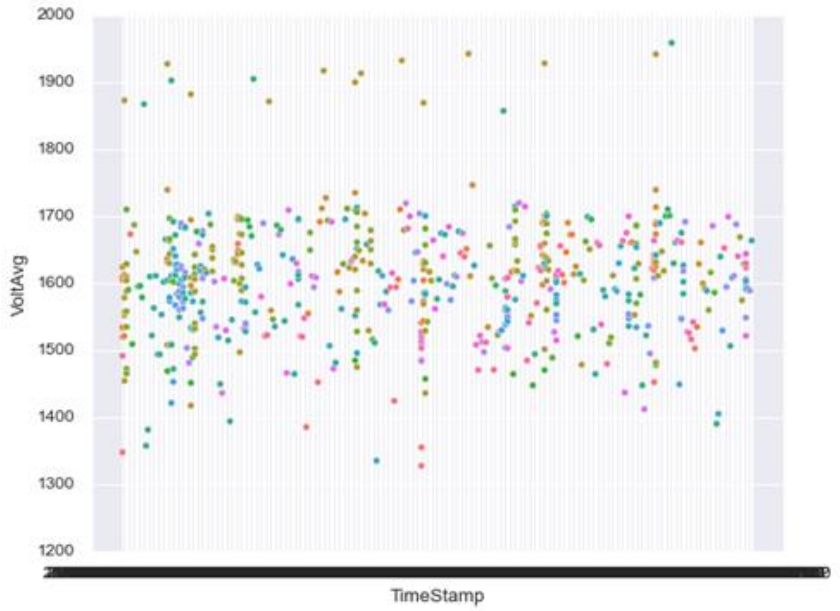
DATA EXPLORATION

2nd step in data pre-processing

A specific data frame was with summarized values of voltage and current curves, i.e. per every pair Spot Name – Time Stamp the voltage curve and current curve have been summarized and for each one a minimum, maximum and average value as well as count of list's elements were calculated.

The calculations have shown that there is no significant correlation between the voltage and the date, nor between the voltage and the spot name. Two following figures show the distribution of average voltage per Time Stamp, at different Spots.



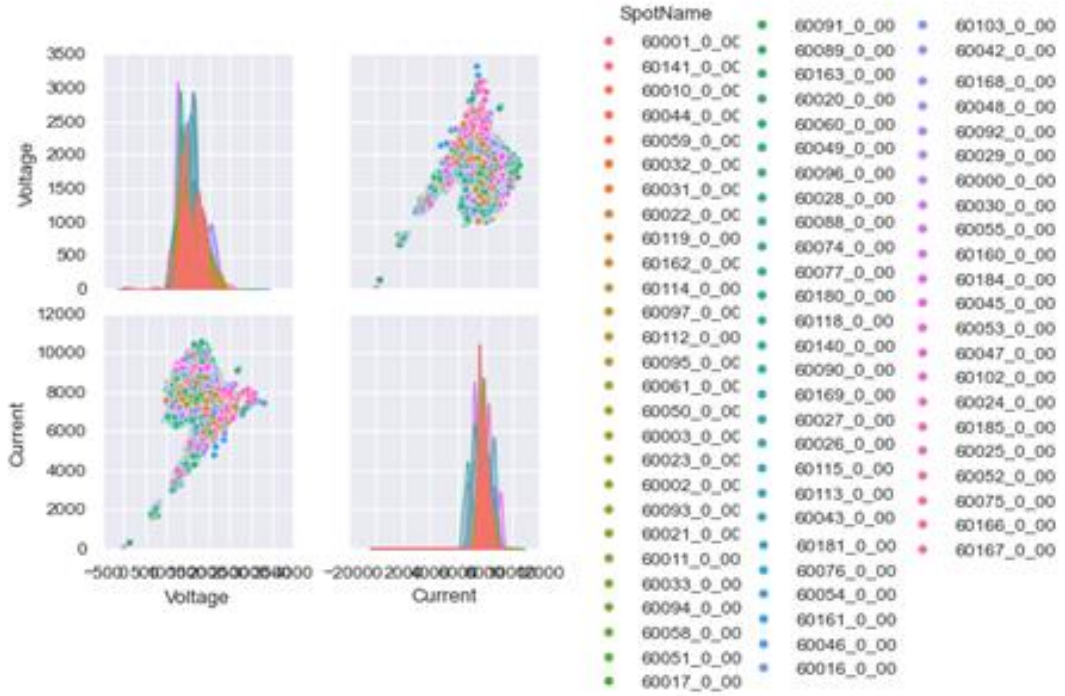


3RD STEP IN DATA PREPROCESSING

This step was aimed at creating a data frame of regular, tabular structure to serve in further analyses and transformation (all values from voltage and current curves that were in a form of array were taken out and put in a table while welding time was incremented by 1ms for each value of the array).

Visualization of the entire dataset was impossible, due to large amount of data. Therefore, data were sampled, and the adjacent figure shows the distribution of the sampled values (1% of all data) of voltages and current, as well as their correlation per welding spot.

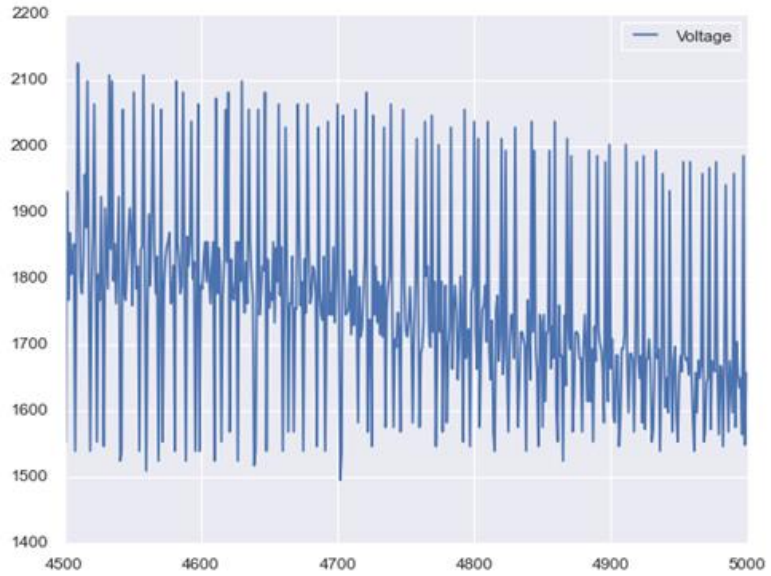
There is correlation between voltage and current.

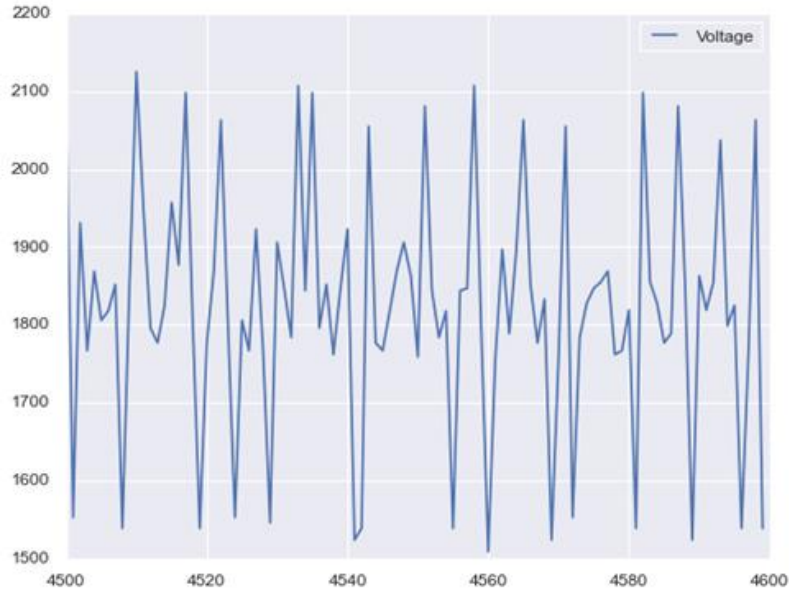


Even the sampled dataset was too big to obtain a meaningful visualization of the values of voltage fluctuation. Therefore, a filter was applied, and voltage values form record indexed as 4500 to record indexed as 5000, as well as voltages between the records 4500 and 4600 are shown in the following figures.

The voltage fluctuation in the sampled and filtered data indicated that it makes sense to use time series analysis for forecasting voltage values.

Due to high density and huge amount of data I decided to perform time series analysis separately for each welding spot.





FURTHER PRE-PROCESSING STEPS

4th step

Was aimed at creating a dataset with possible exogenous variables that might be needed subsequently (Current, Day of the Week, Hour of the Day, Time Stamp).

5th step

In this step, the data were divided by welding spot and therefore 76 .csv files were created, containing, in addition to voltage, other variables that might be needed.

6th step

Data registered in 60 days over eight months were transformed in 60 consecutive days at each spot. These datasets were resampled subsequently at the level of 1ms enabling them to be subjected to time series analysis.

MODELS

- Since data provided were extracts covering just part of a working day of welding guns, and covering only 60 non-consecutive days spanned over eight months, it seemed logical to proceed to direct application of the models, without checking first for the eventual seasonality, stationarity or autocorrelation.
- Therefore, I've chosen diverse models, ones more appropriate for stationary data and others more appropriate for data with seasonality or trends. The application of the models was made easier using the **sktime** Phyton library, which features a unified interface for multiple time series learning tasks. At the time being, this open-source framework support forecasting, time series classification, time series regression and time series clustering.
- Forecasting approaches used in this project can be classified in two groups, one encompassing traditional models (Naïve, ARIMA, Exponential smoothing), and one encompassing models based on ML techniques (Linear regression and Random Forest).

3.1 TRADITIONAL MODELS

Naïve model

This one of the simplest models for forecasting and here it will serve as a baseline model. According to this model, one step ahead is equal to the most recent actual value.

ARIMA (p,d,q) - Autoregressive Integrated Moving Average

This is a linear (i.e., regression-type) model in which the predictors consist of lags of the dependent variable and/or lags of the forecast errors. The parameter **p** is the number of autoregressive terms i.e. the number of "lag observations" ("lag order"); **d** is known as the degree of differencing (it indicates the number of times the lagged indicators have been subtracted to make the data stationary); q is the number of forecast errors in the model and is also referred to as the size of the moving average window. **Auto ARIMA** model from the **sktime** Phyton library was used

Exponential Smoothing (ETS)

This model encompasses a family of forecasting models each having the property that forecasts are weighted combinations of past observations, with recent observations being given relatively more weight than older observations. The weights are exponentially decreasing over time. The weights are dependent on a constant parameter, known as the smoothing parameter. In this project **Auto ETS Model (AETS)** and **State Space Model** have been used.

ML MODELS

ML models available within the sktime framework can be used for forecasting thanks to reduction. Reductionis the concept of using an algorithm to solve a learning task that it was not designed for, going from a complex learning task to a simpler one. Reduction can be used to transform a forecasting task into a tabular regression problem:

- A sliding window approach is used to split the training set into fixed-length windows: e.g. if the window length is equal to 11, the 1st window contains data from time points 0–10 (where time points 0–9 become feature variables and time point 10 becomes the target variable). The 2nd window contains data from time points 1–11 (where 1–10 become feature variables and 11 becomes the target variable), etc.
- Those windows are arranged on top of each other. As a result, data are given in tabular form, with clear distinction between feature and target variables.
- Forecasts can then be generated by recursive, direct or multi-output strategy can be used.

In this project two ML models were used to forecast the voltage values: Linear Regression and Random Forest. **Linear Regression** model assumes that the relationship between the dependent variable and regressors is linear. **Random Forest**, operates by constructing a multitude of decision trees at training time and outputting the value that is mean/average prediction of the individual trees.

TRAINING AND TESTING

Since data pre-processing was very time-consuming, and in order to avoid the situation of not being able to produce any results, I decided to proceed with the project by having two workflows: one basic workflow and the other cross-validation workflow.

Basic workflow was performed without cross-validation. The last dataset from data pre-processing, resampled at the level of 1ms was split into training and testing set, using the appropriate temporal function. Since there is a delay of 50ms from the welding time till data registration, the last 50ms of the training set were excluded and then training and testing were performed (however these 50 data will be used afterwards for validation of quality of the best models).

It was impossible to make predictions from all available data at each welding spot, so I had to cut the training dataset and it was set to be the **last 200.000 records** (i.e. records corresponding to **last 200s**). **Forecasting horizon** was set to be **100 steps**, corresponding therefore to **future 100 ms**. The dimensions of training set and forecasting window were set after trying to obtain predictions with bigger training set and forecasting window, and it was either impossible to obtain results or time for calculations was unacceptably long.

BASIC WORKFLOW: TRAINING AND TESTING

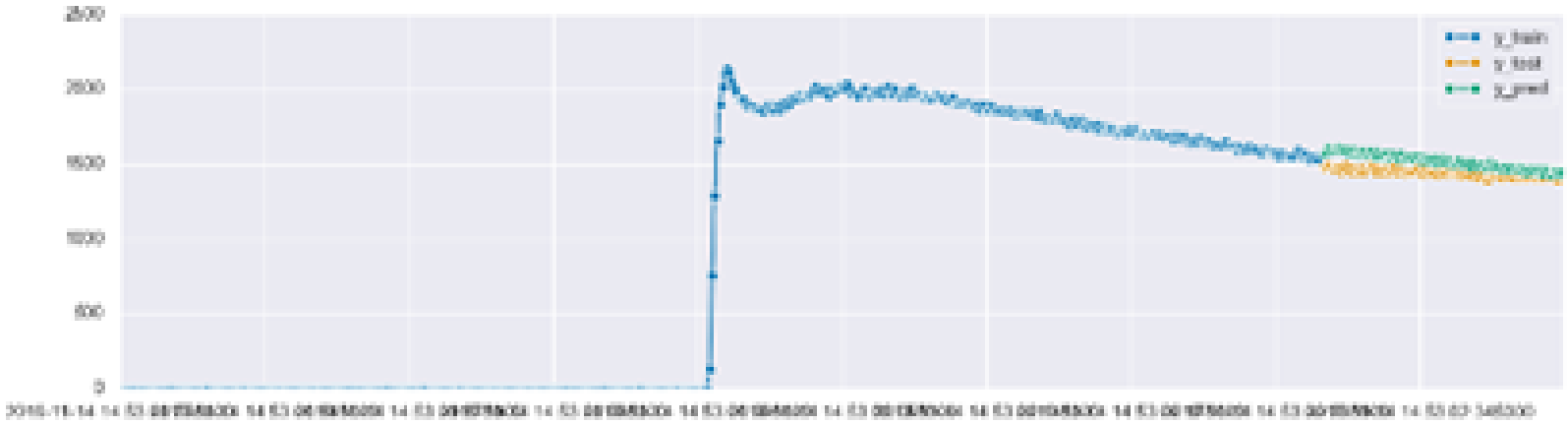
Six previously explained models were then applied to data from all welding spots. It was impossible to train the data from 76 spots parallelly, so only few at a time were trained.

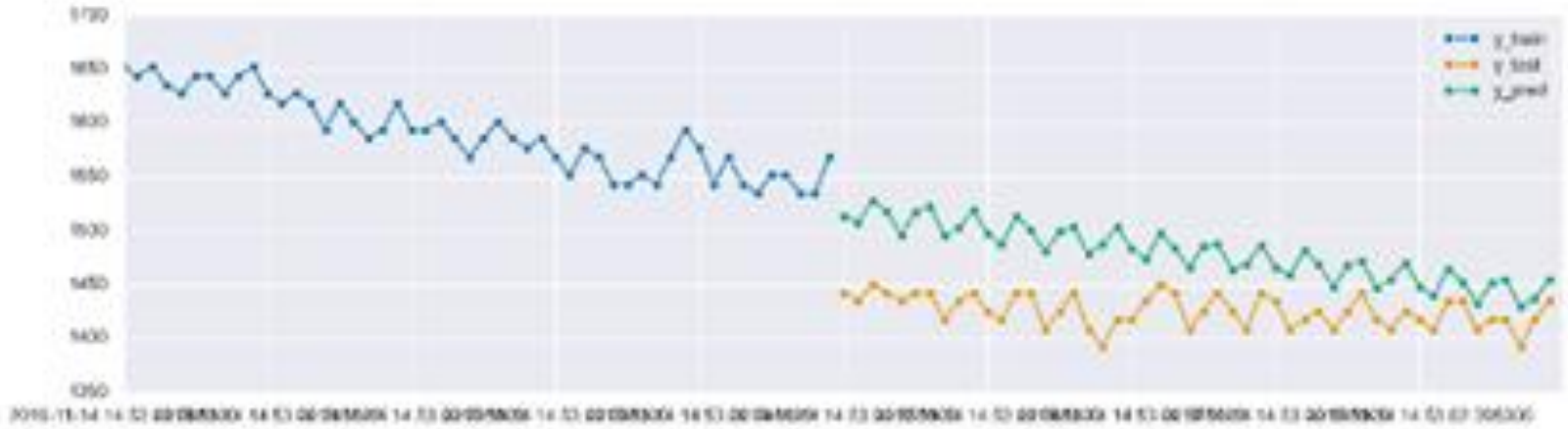
For each prediction the **Mean Absolute Prediction Error (MAPE)** was calculated as a measure of prediction accuracy and the **best model** **(with lowest value of MAPE)** was chosen for each welding spot, and then saved as a .csv files.

Testing was performed with the best model chosen for each spot. Now the testing set was modified: the 50 last records excluded from the training set in the previous stage were added, but in order to respect the delay between welding time and data recording, testing set i.e. forecast window was set to 50: 50 last records corresponding to last 50 ms of the previous testing set (which was set to 100).

As an illustration of this process, here are the visualization of training-testing of linear regression applied to spot named 60000_0_00: first figure relating to a model represent the last 500 records of training set and entire testing set, and the second one is relating to 50 last records of both training and testing sets). Training set is in blue, testing set in yellow, while predicted values are green.

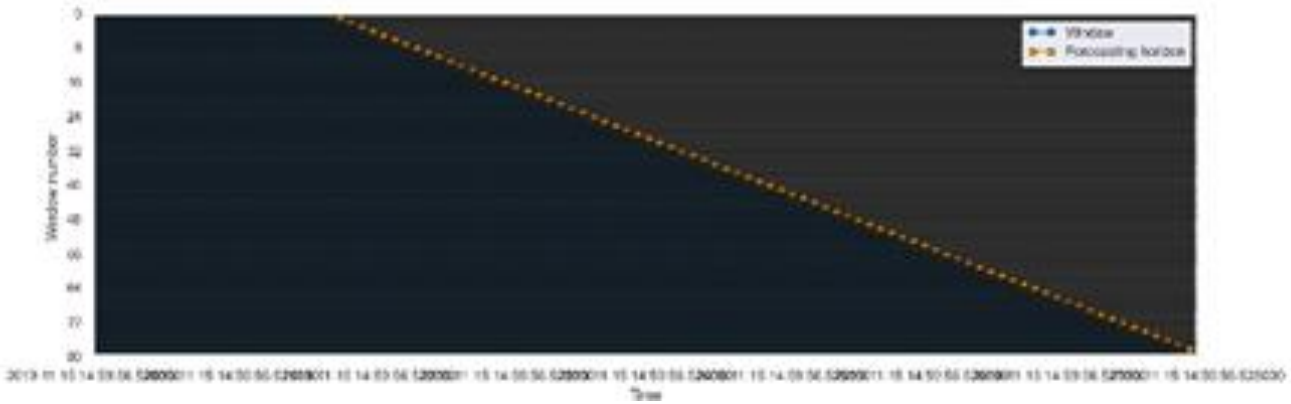
LINEAR REGRESSION VISUALIZATION AT SPOT 60000_0_00





CROSS-VALIDATION WORKFLOW: TRAINING AND TESTING

As far as the cross-validation workflow is concerned, in general, it follows the logic of the basic workflow. However, here, instead of the temporal split function which divided the data in training and testing set, expanding window technique was used for cross validation. Namely, training set was set at **8000** records, with initial window consisting of **1600** records (1/5 of the overall length of the training set), and step length being **80** (1/100 of the overall length of the training set).



Light-blue lines on the left hand side represent each of the 80 expanding windows, and orange dots represent forecasting horizon for each of the 80 steps.

Training-testing process was performed for each model at each welding stop 80 times in order to complete the cross-validation and MAPE was calculated

CROSS-VALIDATION WORKFLOW: TRAINING AND TESTING

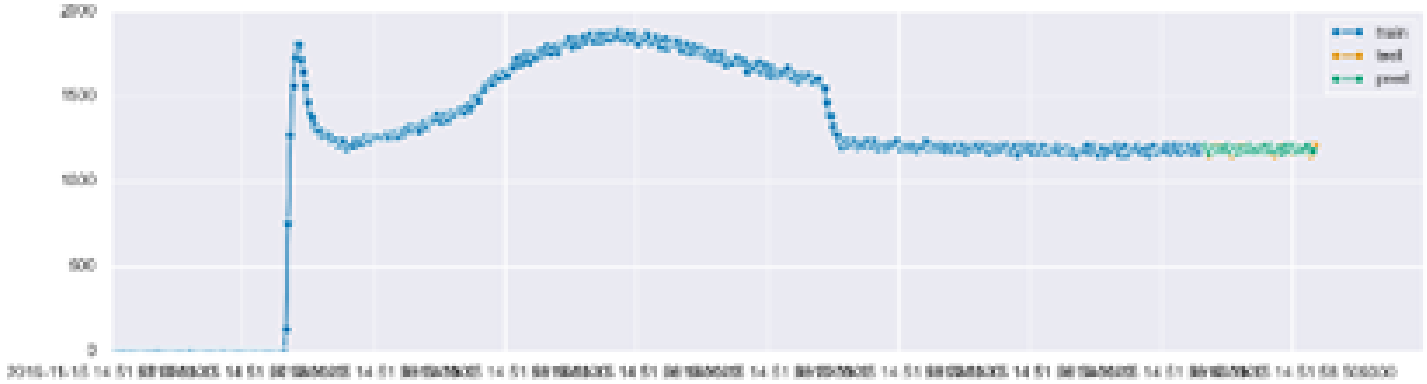
At each spot, **80 MAPE** were calculated for every model. An average value of MAPE was calculated subsequently per model, and the best model at each spot was chosen.

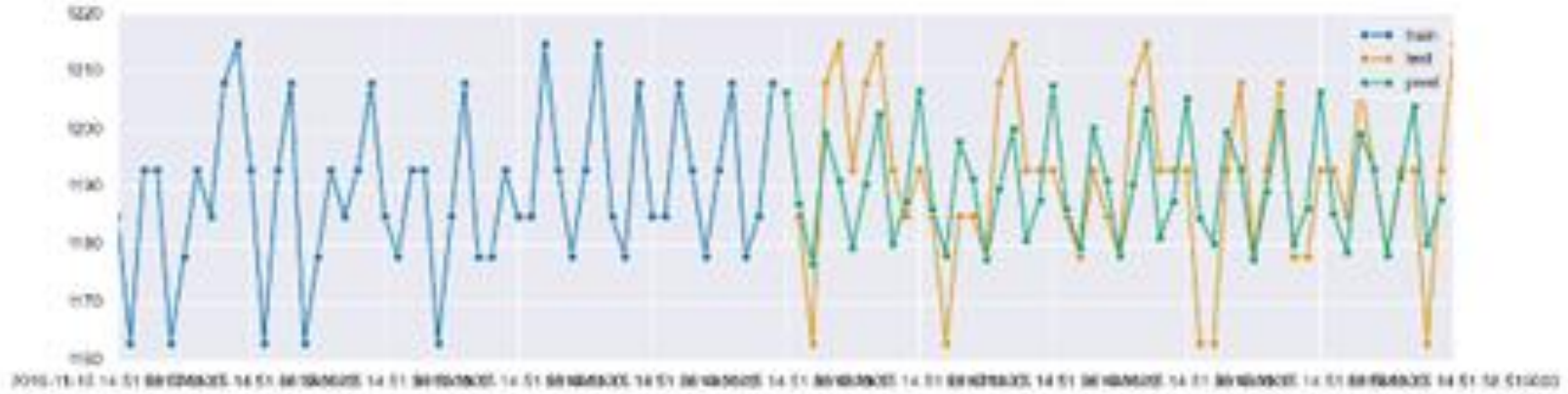
Unfortunately, the lack of processing power didn't allow the execution of the AARIMA model, as it had to automatically fit parameters 80 times per each of the 76 spots.

When it comes to testing, the logic of the basic workflow was followed and the dataset from the basic workflow was used (the one with 200.000 - 50 last records). Forecast horizon was reduced to 50 (the last 50 records) in order to resemble the delay in registering the data. However, it was impossible to perform the testing as in basic workflow – here the best model had to be fitted again when the 50 last records were added to the training set. After the fitting, testing of the best model was executed for each spot and the predictions and visualizations saved.

The following two figures illustrate the results of the testing at the spot name 60076_0_00, where the best prediction was obtained by random forest model. As before, training set is blue, testing set is yellow and prediction is green.

BEST MODEL AT SPOT 60076_0_00: RANDOM FOREST





RESULTS

Basic Workflow

Available results have shown that the best predictions were obtained:

1. at 40% of spots by Linear regression model
2. at 30% of spots by Naïve model
3. at 18% by AETS
4. at 7% by AARIMA
5. at 5% of spots by random forest model.

State Space model was proven to be inadequate as it didn't perform best at any spot. It must be taken into consideration that AARIMA was applied only to one third of the spots, and therefore the percentage of the spots where AARIMA was the best model would've been higher than the one obtained

Cross-Validation Workflow

The best results were obtained:

1. at 54% of spots by naïve model
2. at 30% by linear regression model
3. at 16% by random forest.

It was expected to have different results in this workflow:

→ the training set in the cross-validation model was 25 times smaller,

→ here the AARIMA model couldn't be performed due to lack of processing power.

CONCLUSIONS

Apart from limitations the lack of processing power have imposed on the execution of the project, several important issues have to be taken into consideration:

1. The data available aren't the best choice for time-series analysis, especially without relevant information about the production process (working hours/days…)

2. The biggest issue: how to create a regular time series? I have chosen to collate days (saving the information about the original day of the weak), maintaining the original working time. Since working time ranged from 2,5s to over 50 minutes per day, resampling at the level of 1ms, not only made the dataset enormous and hard for processing, but it made prevailing the periods without data (i.e. with 0s) and it made the MAPE less relevant

3. The approach to creating regular time-series influenced the most the rest of the project. Firstly, because of the size of the data, I wasn't able to proceed with the analysis of the potential exogenous variables. Secondly, not having the exogenous variables, restricted the choice of the adequate model, as some of them (e.g. Extreme Gradient Boosting, Prophet…) need exogenous variables for the forecasting.

CONCLUSIONS

4. Every idea I might've had to improve the presented forecasting process, was not applicable in practice, as I didn't have enough time/processing power after the data transformation was completed.

5. Should I have the possibility of redoing the project, I certainly would not make the same choice regarding the time series creation. Most probably, I would collate data directly based on welding time, and not the day as it was done. Surely, a bit of information would have been lost, but that very information I saved, I haven't been able to use in this project.