Forecasting next month's heating demand

Industry project 2 - Technical Report

Figure 1 Cover Image, source https://www.umweltbundesamt.de/en/topics/health/environmental-impact-on-people/special-exposure-situations/emissions-from-wood-coal-burning-stoves-in#use-of-wood-and-coal-stoves-in-residential-areas

Author: Joël Tauss
Version: 2.1
Date: 10.01.2024

First Supervisor: Prof. Dr. Angela Meyer
Second Supervisor: Prof. Dr. Stefan Grösser

Berner Fachhochschule
Technik & Informatik
Wirtschaftsingenieurwesen

Abstract

Within Switzerland, two of the main resources used for heating residential homes are heating oil and gas. Due to the fact, that gas resources need to be “streamed” over pipelines, compared to the bulk storing possibilities of heating oil, buyers are much more susceptible to price changes and fluctuations.

One of the major drivers of the heating demand is the outside air temperature. Meaning, that if it would be possible to predict next month’s average air temperature, a prediction of the needed gas resources could be made, therefore generate a better pre-calculation, benefiting the buyers. Thus, the goal of this thesis was to generate a model, which can predict if the next month’s median temperature will be above or below the historical average. To achieve this goal, multiple different machine learning models were tested.

The data or features, used to train the models include temperature data, the ENSO, the MJO and the polar vortex data:

- ENSO stands for El Niño-Southern Oscillation. It is a climate phenomenon that, located in the tropical Pacific Ocean.
- MJO stands for Madden-Julian Oscillation. The MJO is an eastward-moving wave of precipitation and atmospheric convection, traveling around the equator.
- The polar vortex is a fast-moving wind, located above the northern pole cap. They occur at altitudes ranging from 20 to 50 kilometres.

Four different models were trained and evaluated to predict, if the next month’s heating demand will be above or below the historical median. The models included a random forest, a support vector machine and two different types of neural networks. The results of the models were sobering. Although all of them were tuned and refined, none of them was able to create a prediction better than a random guess. Noteworthy is, that the random forest and support vector machine slightly outperformed the artificial neural networks.

Henceforth concluding, that the prediction of the outside air temperature, based on the MJO, ENSO, and polar vortex for Switzerland is not achievable with the applied methods.

Table of contents

	Abstract	2
	Table of contents	3
1	Introduction	4
	1.1 Context and problem	4
	1.2 Research question and project goal	5
	1.3 Data	6
2	State of research	6
	2.1 Weather and temperature	6
	2.1.1 ENSO and MJO	6
	2.1.2 Polar vortex	7
	2.2 Temperature forecasting	8
	2.2.1 Neural networks for time series data	8
	2.2.2 Weather and temperature characteristics and forecasting	8
	2.2.3 Heating demand influence factors	9
	2.3 Heating demand	9
3	Methods	10
	3.1 Tools (Programming language and libraries)	10
	3.2 Research design – Data science approach	10
	3.3 Model selection	11
	3.4 Feature engineering	11
	3.5 Training and improving machine learning models	13
	3.6 Model evaluation	13
	3.7 Machine learning models	14
4	Results	15
	4.1 Data gathering and pre-processing (main_0)	15
	4.2 Data exploration (main_1)	17
	4.3 Feature engineering and data preparation (main_2)	23
	4.4 Model selection and model creation	26
	4.4.1 Data splitting and standardizing	26
	4.4.2 Random forest (main_3)	28
	4.4.3 Multi-layer perceptron (main_4)	30
	4.4.4 Recurrent neural network (main_5)	32
	4.4.5 Nu support vector machine (main_6)	34
5	Conclusion	37
	5.1 General findings	37
	5.2 Discussion and research questions	37
	5.3 Further steps and bachelor thesis	38
6	Literature	39
7	Table of figures	41
8	Table of tables	42
9	Declaration of Authorship	43

1 Introduction

1.1 Context and problem

For energy traders to make suitable acquisition of power, the needed energy has to be forecasted as precisely as possible. The energy consumption within Switzerland averaged in at about 225 terawatt hours for the last 5 years. Around 70% of this energy consists of imports, and 30% stems from domestic production. The most used energy sources are petroleum products, nuclear power, hydropower as well as gas (see Figure 2). [1]

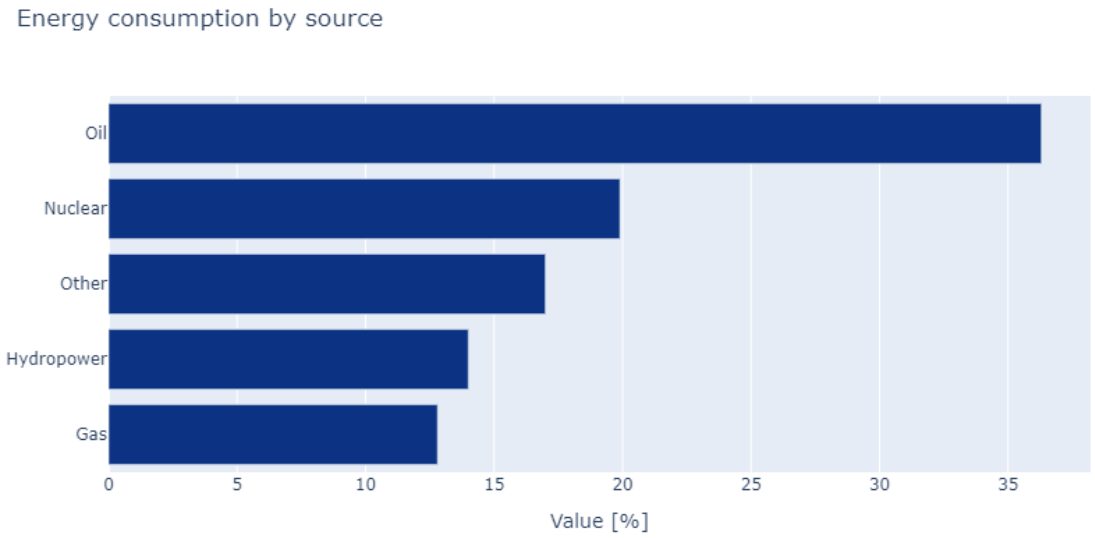

Figure 2 Power consumption by source

With the above-mentioned import rate of energy as well as the main sources, Switzerland heavily depends on the gas and petroleum prices. Regarding the political situation in Ukraine, this is a very current topic. Multiple countries in Europe heavily relied on gas imports from Russia which could have led to serious problems for the concerning countries. [2]
Switzerland's gas imports are sourced over Trans Europa Naturgas Pipeline (TENP), which is connected with the Transitgas pipeline. Supplies of the later mainly stem from Russia. Around 10% of the Transitgas's are used in Switzerland. [3]

One main driver of the energy consumption of Switzerland, especially in the colder months, is heating of residential and industrial properties [4]. When looking at energy types used for heating, it is viewable why the forecasting of the needed energy is important. Around 30'000 households rely on gas as a power source for heating. The locally stored gas resources are quite low when compared to the domestic consumption. [5] [6]
The focus the project is on gas as a power source. The reason being, that gas is most often "streamed", as previously stated, to where it needs to be. This contrasts with heating oil which, can be stored more easily in tanks. Because of this, the impact of short-term pricing changes have a bigger impact on buyers.

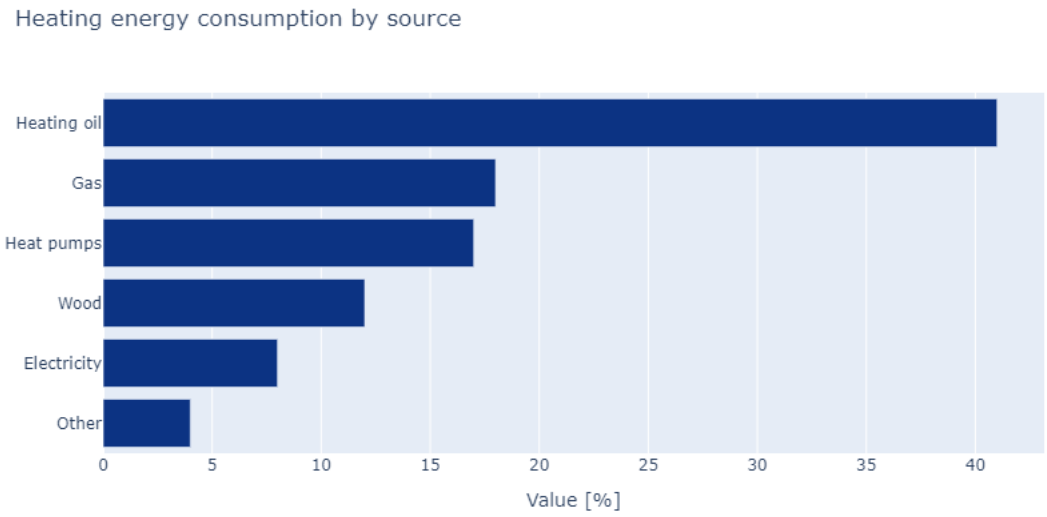

Figure 3 Heating energy consumption by source

As previously stated, the biggest factor influencing the heating demand is the outdoor temperature. For this thesis, factors like insulation and building shapes are neglected, because the forecasting is done on a national scale and will not be broken down to regions or even cities. [7]

The current models of forecasting the temperature typically drop in accuracy the longer the forecasted period is. The accuracy of typical weather forecast is at about 50% reaching the 10-day mark [8]. Meaning a gas demand forecast can only be done for up to 1 or 2 weeks. Such weather forecasts are done by first analysing and understanding the current atmospheric conditions and what is causing them. Different data and data sources, like surface observations, radar data, and satellite images are needed to create said understanding. This data is updated and checked regularly. Afterwards, meteorologists make use of different tools to track the evolution of weather fronts, jet-streams, as wells as cyclones and anticyclones. Pattern recognition also plays a role in this process. The weather forecast then gets compiled accordingly. [9]

Because of the limited nature of “conventional” weather forecast with the range of about 2 weeks, the heating demand can only be predicted for the same time span. In order to achieve the goal of this thesis, a different approach for predicting the temperatures is needed.

1.2 Research question and project goal

The prediction of the heating demand is done with different machine learning models, based on historical weather and climate data. Goal being to predict if the temperature of the next month will be below or above average. The research question can be derived as following:

- What data is available and suitable for the forecasting of the next month’s average temperature and how does the data need to be transformed?
- How can the heating demand for the next month be forecasted, based on the temperature, categorized by below or above average temperatures (classification model)?
- What is the forecast accuracy and how can it be improved further (influence factors)?

A side objective is to analyse and predict polar vortexes. Every two to three years strong polar vortexes can occur. At the time of slowing down, the break apart and carry a cold wave with them. Resulting in lower temperatures and higher heating demands in Europe. [10] [11]
This information could provide further insight and aid in anomaly detection, and so add value to the machine learning models.

1.3 Data

Four different types or sets of data will be considered for this project. All of these are taken from different sources, but can be linked up by date time indexes. The data sources as well as the pre-processing is part of chapter 2 and 4.1. The four datasets used are:

- El Niño Southern Oscillation index (ENSO)
- Madden Julian Oscillation index (MJO)
- Air temperature 2 meter above ground
- Polar Vortex data as wind speed components of the northern polar cap

The explanation of the indexes and data sets can be found in chapter 2.1.

2 State of research

2.1 Weather and temperature

2.1.1 ENSO and MJO

In order to create a viable forecast and comparing it to a benchmark, both components have to be calculated. The baseline for the forecasting consists of a climatology benchmark: monthly mean and standard deviation of the air temperature from 1979 to 2022 in Switzerland. This benchmark is calculated by aggregating ERA5 reanalysis data. ERA5 data provides hourly, or monthly information about metrics regarding land and oceanic climate [12].

Two features of the machine learning models, used to predict anomalies in air temperature (and the temperature itself) will be:

- El Niño Southern Oscillation index (ENSO)
- Madden Julian Oscillation index (MJO)

El Niño Southern Oscillation index (ENSO) describes the fluctuation in sea surface temperature (El Niño), as well as the air pressure of the atmosphere above (southern oscillation) across the equatorial Pacific Ocean. The southern oscillation is a bimodal variable in the bevor mentioned air pressure of the atmosphere, expressed as the Southern Oscillation Index (SOI). [13]
The ENSO is calculated from the El Niño and La Niña, representing the warm and cold phase respectively. It shifts in an irregular interval of two to seven years from to the other. The corresponding effects on the weather, temperature and rainfall are well explored and documented. [14]

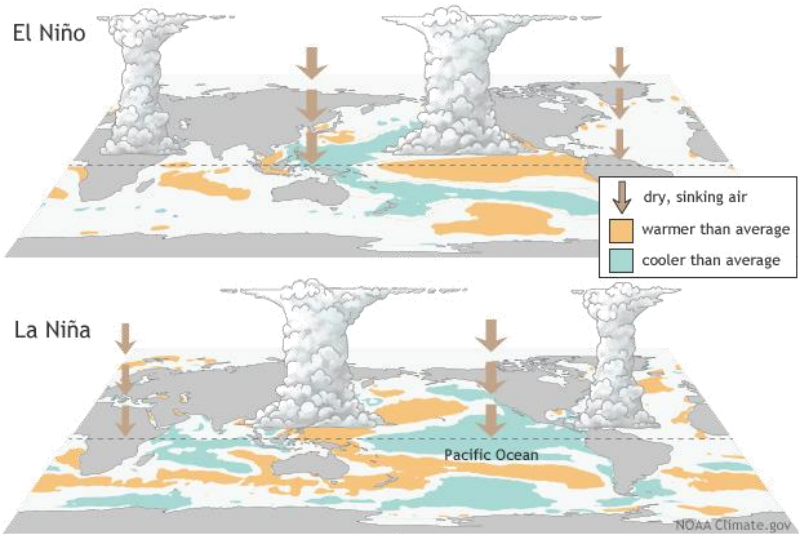

Figure 4 ENSO, source: https://www.climate.gov/enso

The Madden Julian Oscillation index (MJO), in contrast to the stationary ENSO, is an east moving disturbance of pressure, winds, and rainfalls. After traveling for 30 to 60 days on average, it returns to

the starting position. It is possible for multiple MJO events to occur during a season. So, it could be described as an intrapersonal tropical climate variability. [15]
The progression of the MJO is split up into 8 phases, describing its eastward progression and effects on different regions. As it is the case with the ENSO, the MJO's effect on weather and temperature are well documented. It is important for extended weather forecasting for north America and Europe, due to the changes in windspeeds and rainfall which effect the tropics and extratropics. [15]

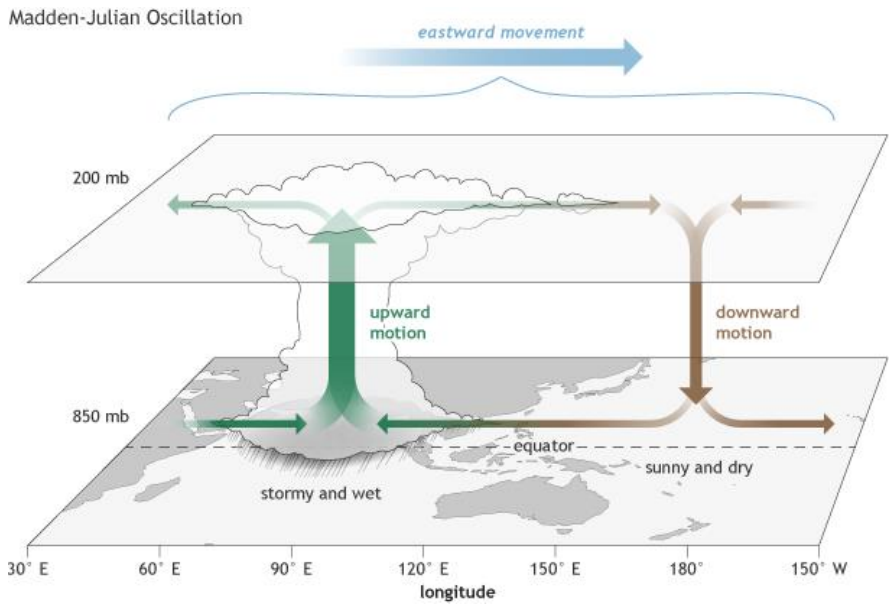

Figure 5 MJO, source: https://www.climate.gov/news-features/blogs/enso/what-mjo-and-why-do-we-care

2.1.2 Polar vortex

As the name suggests, the polar vortex is a circular wind, occurring in the stratosphere at heights of up to 50 km over the polar regions. It is almost always present, although the wind speeds can differ heavily, whereas a stronger vortex, refers to a more stable state and vice versa. A weaker polar vortex can lead to weakened and southward shifted jet streams, as well as allowing the colder air to be carried south and therefore cause colder temperatures. Concluding, that the wind speeds correlate negatively with temperature in the northern hemisphere. [10] [11]

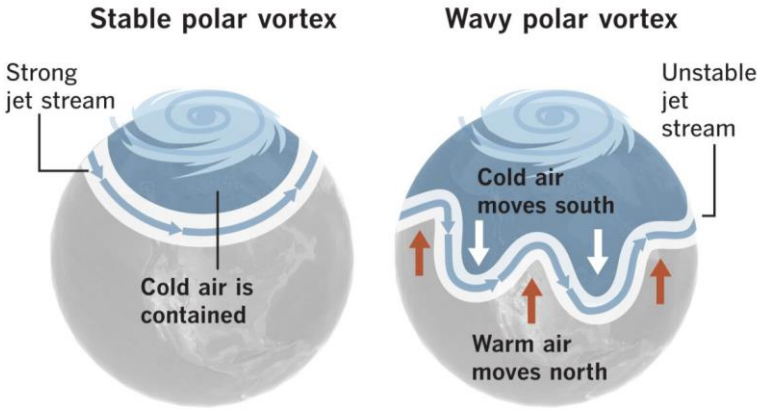

Figure 6 Polar vortex, source: https://www.latimes.com/california/story/2020-03-28/if-a-warm-u-s-winter-was-a-preview-of-global-warming-what-part-did-a-polar-vortex-play

2.2 Temperature forecasting

2.2.1 Neural networks for time series data

In the past few years, many different machine learning models and neural network architectures have been developed for time series data in multiple different subject domains. Those vary from single step to multiple step forecasting. The basic building block for one-step ahead forecasting models consist of the following logic: [16]

$$\hat{y}_{i,t+1} = f(y_{i,t-k:t}, x_{i,t-k:t}, s_i)$$

$\hat{y}_{i,t+1} := model\ forecast$
$y_{i,t-k:t} := observations\ of\ the\ targetover\ a\ look\ back\ window\ k$
$x_{i,t-k:t} := observations\ of\ the\ exogenous\ inputs\ over\ a\ lookback\ window\ k$
$s_i := static\ metadata\ associated\ with\ the\ entity$
$f := the\ prediction\ function\ learnt\ by\ the\ model$

Deep neural networks are trained to recognise relationships beneficial for predictions by using nonlinear layers to construct intermediate features. This is what happens on the hidden layers of a deep neural network. When working with time series data, said process can be viewed as encoding and decoding the given historical data to predict the parameter. [16]

There are multiple different neural networks, which are suitable for time series data. Some examples include:

- Convolutional neural networks (CNNs):
  The originally intended use of CNNs was image processing and recognition. They extract local relationships between individual pixels (or spatial dimensions). To use a CNN for time series data, multiple layers of convolution are used in order to incorporate past data when creating predictions. [16]

- Recurrent neural networks (RNNs):
  In the past RNNs were used for sequence modelling. An example and often occurring use case of this would be natural language processing. To use RNNs on time series data, the data can be modelled as sequences of inputs and target vectors. All by keeping the structure intact. The distinguishing feature of the RNNs are the individual stm nodes (short term memory cells), which contain an internal memory state used for saving past information. These get update when data passes through the network. Long short-term memory cells were later implemented to combat the disadvantages of the short-term memory cells. With these, dependencies over greater timespans can be covered. [16]

2.2.2 Weather and temperature characteristics and forecasting

Models for forecasting global ENSO-related climate anomalies have already been compiled to some degree. These are based on multiple different models, such as the Tropical pacific coupled model based on statistical methods and atmospheric models. By creating predictions of the pacific sea surface temperature first and compiling these results with the atmospheric model to produce a climate forecast. [17] The forecasting of the MJO also have been explored. Each of the different methods having its advantages and disadvantages. Most of them relying on different measurements, combined with mathematical methods and established models. [18]

Further methods, concerned with the MJO location and intensity, displayed by the real-time multivariate MJO (RMM) index, are predicting its annual and seasonal variability. One approach classified these into 4 categories: inactive, active, very active, and extremely active. Other factors were also researched. Statistical methods and correlations were used to determine these factors. [19]

When it comes to temperature forecasting, statistical approaches may also pose an interesting approach. Indicators and methods based on historical data such as the following were used with success: [20]

- Autoregressive Fractionally Integrated Moving Average
- Exponential smoothing
- Trend and seasonal components
- Simple exponential smoothing
- Theta and prophet methods

The idea of forecasting air temperatures with neural networks is not new. In the past years multiple different studies made use of RNNs with long short-term memory cells as well as multi-layer perceptron's (MLP) to create forecasts. Reviewing these papers and approaches, it shows that it is a promising methodology for temperature predictions. But no consensus yet exists on which is the best methodology in this field.
Interesting to note is, that deep neural networks were found to be more viable for short term air temperature predictions, rather than long term. When it comes to input variables for models, most often the following were used: air temperature, wind speed and direction, air pressure, precipitation, solar radiation, relative humidity, cloudiness, latitude, longitude, and altitude. But because this is a highly complex topic, other or more input variables may yield better results. [21] [22]

2.2.3 Heating demand influence factors

Although temperature is one of the most obvious factors influencing the heating demand, others should also be mentioned. Some key aspects influencing the heating demand include the following: [23]

- Total population, demographic distribution, number of households
- Household sizes and type
- Income and liquidity
- Heating choices of inhabitants:
    - Chosen preferred temperature
    - Size and characteristics of internal areas
    - Amount of heating hours
    - Extent of occupants heating control
- External weather conditions and location dwelling
- Characteristics of the building (i.e., isolation)

2.3 Heating demand

Heating demand can be split up and categorized, to create a more conclusive and detailed picture. Differencing factor are (listing not conclusive): [24]

- Building archetypes within Switzerland. These can be split up into 100 residential and 45 non-residential types.
- Different scenarios, under which the heating demand changes.
- Climate zones. In total 54 climate zones, relevant for the heating demand can be identified

This separation and profiling may not be needed in the definitive paper. The granularity is too fine. But this could pose a possibility for further research and improving the model.

3 Methods

3.1 Tools (Programming language and libraries)

For this project, the programming language python is used. It is one of the more powerful tools for machine learning with a vast variety of libraries to choose from. The following packages are used (listing not conclusive):

- pandas
- plotly.express
- sklearn
- tensorflow / keras

3.2 Research design – Data science approach

The steps of the accompanying jupyter notebooks, as well as the base research design follows the standard approach for data science. Note, that the order was adjusted to fit the project. [25]

- Data collecting:
  Because there are different types of data sets needed, different sources will be used. The used data is: Surface air temperature 2 meters above ground, ENSO index data, MJO index data, compiled wind speed data to approximate information about the polar vortex. This step includes the pre-processing of the data. Some of the data sources provide .nc files (NetCDF), which must be compiled to be used in further steps. Furthermore, the data will be aggregated onto monthly intervals if needed, by averaging the data, calculating standard deviations (mean or mode), or using the last datapoint.

- Data cleaning:
  Part of the data cleaning is done during the data collecting state. Missing data is handled by not including it in the data set or filling it with the mean value of a respective region or time-period. This results in a continues data set, ranging from 1979 to 2022.

- Data exploration:
  To gain an understanding and overview of the data at hand, it will be thoroughly explored in form of numerical values as well as plots.

- Data preparation and feature engineering:
  The main preparation, which is needed, is the creation of the categories “above” and “below” average monthly temperature. How a month is classified depends on whether the mean air temperature value of a month is above or below the median of said month of the given dataset. This ensures a split of exactly 50 / 50. Additionally, the time features are transformed with a cyclic function. In this case the sine and cosine functions are applied on a yearly interval. This ensures, that each datapoint can be identified uniquely in regard to the year. For the neural network models, the data also must be standardized (see chapter 3.4).

- Model building and evaluation:
  The different machine learning models are created and optimized for several different parameters, depending on the model type. Each model is compared to one another to conclude the best way to go about the task at hand. Library built in functions like cross validation are used where applicable.

3.3 Model selection

Four different machine learning models are applied to create the classification model. A brief explanation can be found in chapter 3.7:

- Random forest classifier
- Multi-layer perceptron classifier
- Recurrent neural network with long short-term memory cells
- Nu support vector machine

These models were chosen for different reasons. The random forest model is a simple to understand and easy to set up model, which grants a lot of insight on each feature and its importance within the model. The support vector machine works well on data with many dimensions, even if the training data is low in number of datapoints. Which is the case with only 528 datapoints, with 26 features each.
Neural networks on the other hand, are highly applicable for more complex problems. [26]

The multi-layer perceptron poses a simpler solution in contrast to the recurrent neural network and can be used as a base line indicator when it comes to neural networks. The advantage of the recurrent neural network are the long short-term memory cells or nodes. These can retain information and therefore take previous time steps into account. Which makes them suitable for any time series related task. [16]

One of the more important factors when building a neural network is the activation function. This function defines, how node inputs get transformed and allow nonlinearity within the model. For the sake of simplicity and due to the time constraint, the activation functions of the two neural networks will be fixed and not optimized. [27]

During this project, the multi-layer perceptron uses the ReLU action function exclusively. This activation function provides fast computation when training the model and granting a quick gradient descent. One disadvantage of the ReLU function is, that is not zero cantered and therefor its outputs are not normalised. Additionally, because of their nature, it is possible that the ReLU function ends up in a dead state. Meaning that high input data during training can cause a left shift of the function and producing zero values as output on consecutive data points. This is also called the dying ReLU problem. The ReLU function is defined as: [27]

$$f(x) = \max(0, x) \qquad 1$$

The recurrent neural network makes use of the SeLU activation function. Mainly to negate the bevor mentioned problems with the ReLU function, while still having access to its advantages. [28]
The multi-layer perceptron does not use the SeLU function, because the sklearn library does not support it. The SeLU function is defined as: [29]

$$f(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha e^x - \alpha, & x \leq 0 \end{cases}$$

3.4 Feature engineering

When working with time series data, it is useful to engineer some features, based on the previous data point. Some examples which could aid with the prediction are:

- Moving average (MA): Rolling mean of the last n data points. Can hint if the current value is a rising or falling trend, depending on how n is chosen. [30]
- Exponential moving average (EMA): Is somewhat similar to the MA but places more weight on more recent data points. It also fulfils the same purpose. [31]
- Relative strength index (RSI): Is a momentum or strength indicator of the current trend. It measures the speed and magnitude of a value. Can hold information on how long a trend will last. [32]

Because machine learning models cannot really make sense of timestamps or date time objects, these features are transformed to convey the cyclic nature of years. To achieve this, cyclic functions are used. Typically, the sine and cosine functions get applied on different time scales. Because of the nature of these two functions, both are needed to create uniquely identifiable points. [33]

A common practice when training neural networks is to standardize the datapoints in order to achieve a better result. For standardizing data, the following equation gets applied [34]:

$$z = \frac{value - mean}{standard\ deviation}$$

Because the ENSO and MJO are two of three main features used during this thesis, an analysis on outliers is conducted. The smoothed z-score algorithm is applied to detect positive or negative peaks within the time series. The implementation in python of the algorithm (code snippet is cited from source): [35]

```
def thresholding_algo(y, lag, threshold, influence):
    """Robust peak detection algorithm (using z-scores)

    Args:
        y (_type_): y_vector / time series
        lag (_type_): the lag of the moving window
        threshold (_type_): the z-score at which the algorithm signals
        influence (_type_): the influence (between 0 and 1) of new signals on the mean and

    Returns:
        _type_: dict {
            singals
            avgFilter
            stdFilter
        }
    """

    signals = np.zeros(len(y))
    filteredY = np.array(y)
    avgFilter = [0]*len(y)
    stdFilter = [0]*len(y)

    avgFilter[lag - 1] = np.mean(y[0:lag])
    stdFilter[lag - 1] = np.std(y[0:lag])

    for i in range(lag, len(y)):
        if abs(y[i] - avgFilter[i-1]) > threshold * stdFilter [i-1]:
            if y[i] > avgFilter[i-1]:
                signals[i] = 1
            else:
                signals[i] = -1

            filteredY[i] = influence * y[i] + (1 - influence) * filteredY[i-1]
            avgFilter[i] = np.mean(filteredY[(i-lag+1):i+1])
            stdFilter[i] = np.std(filteredY[(i-lag+1):i+1])
        else:
```

```
            signals[i] = 0
            filteredY[i] = y[i]
            avgFilter[i] = np.mean(filteredY[(i-lag+1):i+1])
            stdFilter[i] = np.std(filteredY[(i-lag+1):i+1])

    return dict(signals = np.asarray(signals),
                avgFilter = np.asarray(avgFilter),
                stdFilter = np.asarray(stdFilter))
```

3.5 Training and improving machine learning models

For creating the best possible machine learning models within the given time frame, each model will be fine-tuned (also called hyper parameter tuning) by an automated process. Because it is not possible to test all permutation with the given numbers of possible factors, this is done in sequence. Custom classes are written to automate this process.

3.6 Model evaluation

To assess and evaluate the classification models, two metrics are used in combination with one another. The model accuracy and the confusion matrix. The accuracy describes how many of the classes were identified correctly (number of correct prediction / total number of predictions). But if the given data sample favours one category, this value alone can be misleading. Hence, the confusion matrix is needed as well. It is consisting of a 2 by 2 grid, providing detailed information on the actual class and the predicted class. [36]

The built-in function of the library sklearn, used to create the confusion matrix, seems to switch the two axis of the actual and predicted values. That is important to keep in mind when interpreting the plots.

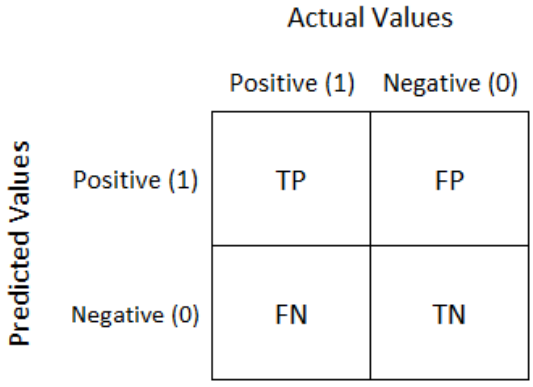

Figure 7 confusion matrix, source : https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

3.7 Machine learning models

The random forest model is an ensemble machine learning model, which can be utilized for classification, regression and other tasks. It consists of multiple decision trees and combines them to obtain a more accurate prediction, by averaging the output of all trees together. Each tree is built using a different subset of the provided data, as well as a random number of features. [37]

The nu support vector machine is a type of binary classifier that finds the hyperplane which best separates the given data points into the two classes. It is a variation of the traditional support vector machine. The main parameter nu is used to control the trade-off between the training errors and the number of support vectors used in the model. The nu support vector machine is able to handle non-linearly separable data by using different kernel functions. The kernel functions map the data into a higher-dimensional feature space. [38]

The multi-layer perceptron is a type of feedforward neural network, with one input, one or more hidden, and an output layer. Each layer consists of nodes. All of them are fully connected to the nodes of the following layer, meaning each node from a layer feeds into every node of the next one. Each node or cell applies an activation function to its input to transform the value bevor passing it on. The model learns through backpropagation. [39]

The recurrent neural network is also a neural network, but the nodes are so called long short-term memory cells. Additionally, the model is not a feedforward neural network, meaning information can flow back as well (bidirectional). The long short-term memory cells are able to learn and remember (or store) information for a certain period of time. Each cell contains multiple gating mechanisms, which allow them to forget or remember certain previous information, as well as update their internal state, based on the current input. [40]

4 Results

4.1 Data gathering and pre-processing (main_0)

The data gathering is split up into three parts, depending on the data source. The temperature data as well as the wind speeds for deriving the polar vortex data are sourced form Copernicus over their python API library module [37]. The ENSO [38] and MJO [39] data is downloaded as text-files form their respective sources. All four datasets were chosen in a way, that there is no missing datapoints within. The date range includes 1979-01-01 until 2022-12-31.

To restrict and simplify the matter for the air temperature data (2m above the ground), a fixed square was taken as an outline of Switzerland, defined by the following boundaries:

Latitude north:	48.0°
Latitude south:	45.7°
Longitude east:	11.0°
Longitude west:	5.8°

Figure 8 Switzerland outline for temperature measurements

The API returns multiple data points in a grid from and hourly resolution. These datapoints are first averaged together over the area and in a second step, aggregated over time. The aggregation over time is done in 2 values: Mean and standard deviation.

For the poler vortex data, a similar approach is used. Because this feature is much more experimental during this project, only a small grid is taken as a sample. The data within the grid contain the three components of wind speeds: u, v, w in meters per second. The method of aggregations is the same: Averaging over the area in a first step, and calculation the mean and standard deviation on a monthly time interval in a second step. The relevant grid is defined as follow:

Latitude north:	82.0°
Latitude south:	81.0°
Longitude east:	13.0°
Longitude west:	12.0°

Figure 9 Geolocation of polar vortex data

An offset was chosen by shifting the grid from the geographical north into the direction of Switzerland, because it is possible for the wind speeds at the centre of the vortex to be much slower. Therefore, the datapoints would hold less relevant information. Important to note is, that the given data source does not provide data at certain elevations in meters. Instead, the data is provided by pressure level. Because atmospheric pressure changes not only with elevation, but also with temperature and humidity, it is not possible to get the measurements at an exact elevation. [37]
For the given sample, the pressure level of 10 hPh is chosen, which approximates to an altitude of 25 kilometres. [38]

The ENSO data consists of single index with a monthly interval. Thus, no aggregation is needed, and the values are taken as provided by the data source.

The MJO data set contains two types of information: the RMMs value and phase value. The RMMs contain two components, representing the x (rmm1) and y (rmm2) axis. Combing those values, indicates where exactly the disturbances are located. By treating these values as a vector, the amplitude can be displayed as the length of the vector. The grid on which the rmm1 and rmm2 are drawn, consist of 8 phases, which correlates to a certain region.

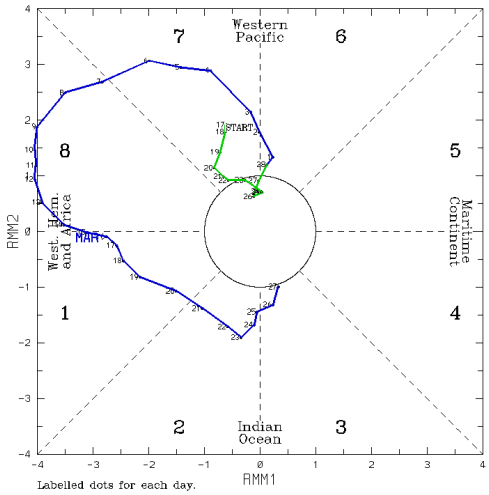

Figure 10 MJO data sample, source: http://www.bom.gov.au/climate/mjo/

A problem which could arise when aggregating the data only by averaging it out, is that the location ends up in a weird place. Because of this, two separate data frames are created:

- An aggregated one with means, standard deviations and mode for the phase
- A data frame with the last value of each month

The data is then joined by means of a date time index to create data frame, which looks as follows:

	0	1	2	3	4
index	1979-1	1979-2	1979-3	1979-4	1979-5
year	1979	1979	1979	1979	1979
month	1	2	3	4	5
enso	0.47	0.26	-0.08	0.2	0.27
pv_u_mean	-7.705095	-10.069668	0.974978	-2.369439	-2.055928
pv_u_std	27.281851	23.430696	11.492888	4.466771	2.138089
pv_v_mean	-4.846178	-19.538984	13.36226	-2.108374	-0.777698
pv_v_std	12.622939	10.261012	15.734375	4.275214	2.176962
pv_w_mean	-0.000723	-0.001674	0.00016	-0.000007	-0.00009
pv_w_std	0.003739	0.005229	0.003275	0.00194	0.001966
t2m_mean	266.677674	271.801301	274.849874	276.097354	281.669298
t2m_std	4.648611	3.107695	2.938513	3.591025	5.561861
rmm1_mean	-0.750029	0.458212	0.636508	0.03863	-0.114053
rmm2_mean	1.106029	-0.745376	-0.322627	-0.206593	0.409473
phase_mean	5.774194	3.285714	4.0	3.866667	5.677419
amplitude_mean	1.719532	1.063415	1.389601	1.844991	1.677857
rmm1_std	0.795243	0.490798	1.202352	1.433251	1.295421
rmm2_std	0.988432	0.587141	0.615136	1.2141	1.133189
phase_std	2.390393	1.049061	1.788854	2.255007	1.868816
amplitude_std	0.632004	0.454729	0.595723	0.229275	0.489784
phase_mode	7	3	4	3	4
rmm1_last	0.128084	0.891485	0.338955	-1.12556	-2.01605
rmm2_last	-0.5824	-0.65551	-1.18882	-1.15801	0.075006
phase_last	3	4	3	2	8
amplitude_last	0.596314	1.10654	1.23619	1.61489	2.01744

Table 1 merged data frame

4.2 Data exploration (main_1)

In order to get a grasp of how each factor correlates with each other, a correlation matrix is plotted. The most noteworthy correlations are:

- Polar vertex v and w component: This is not surprising at all. The higher the wind in one direction, the more likely it is to be equally into the other. The same goes for the standard deviation values.
- Mean air temperature and standard deviation of polar vortex's: This could be explained by seasonal changes. Because, as previously stated, the polar vortex's strength shift with change in seasons.

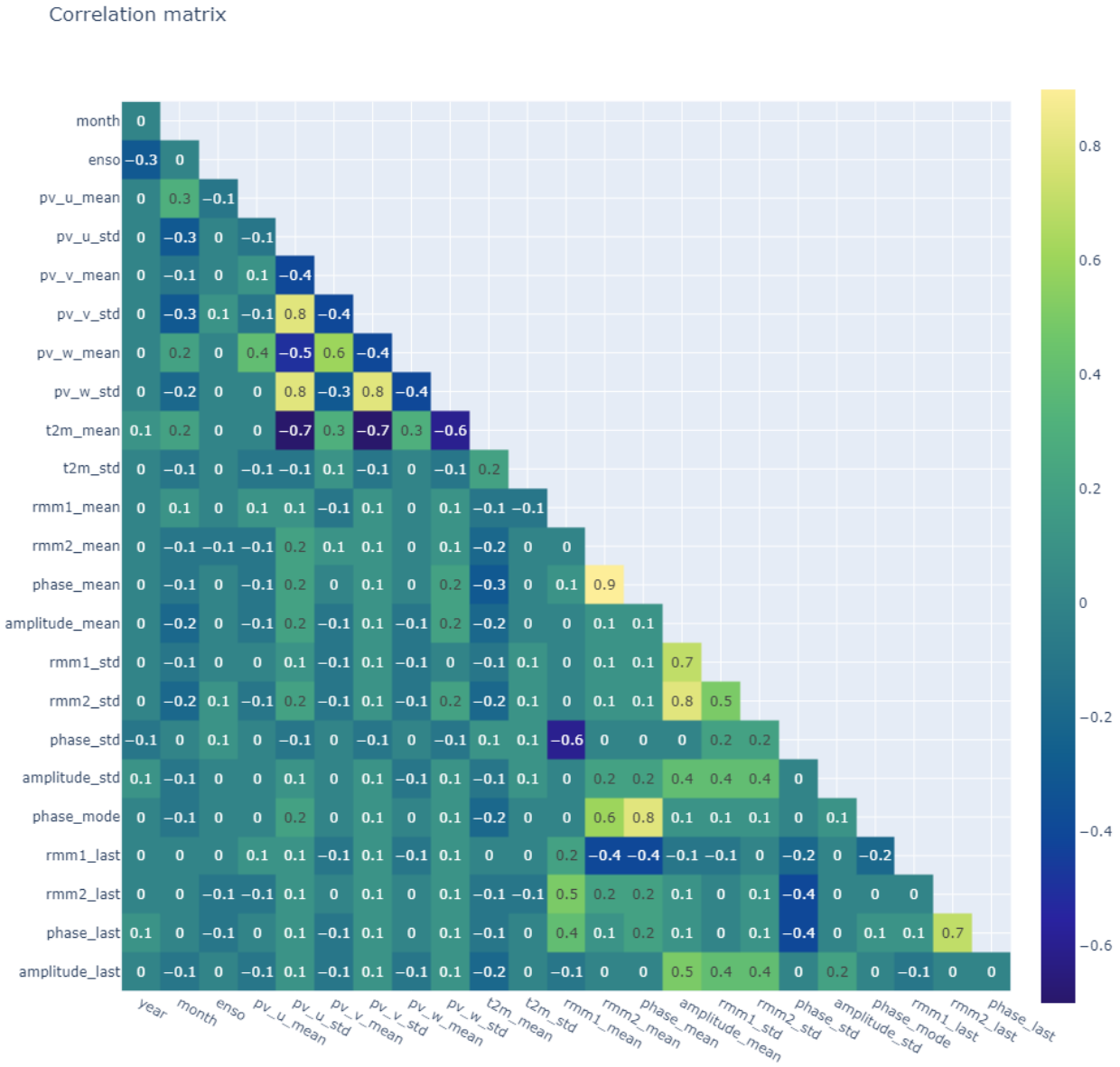

Figure 11 Correlation matrix

The target variable, or y, will be transformed into 2 categories: Above and below average. But first it is crucial to check how the data is distributed among the months themselves. The probability density shows that the data suggest a normal distribution for each month. Although not too many data points are available for each month.

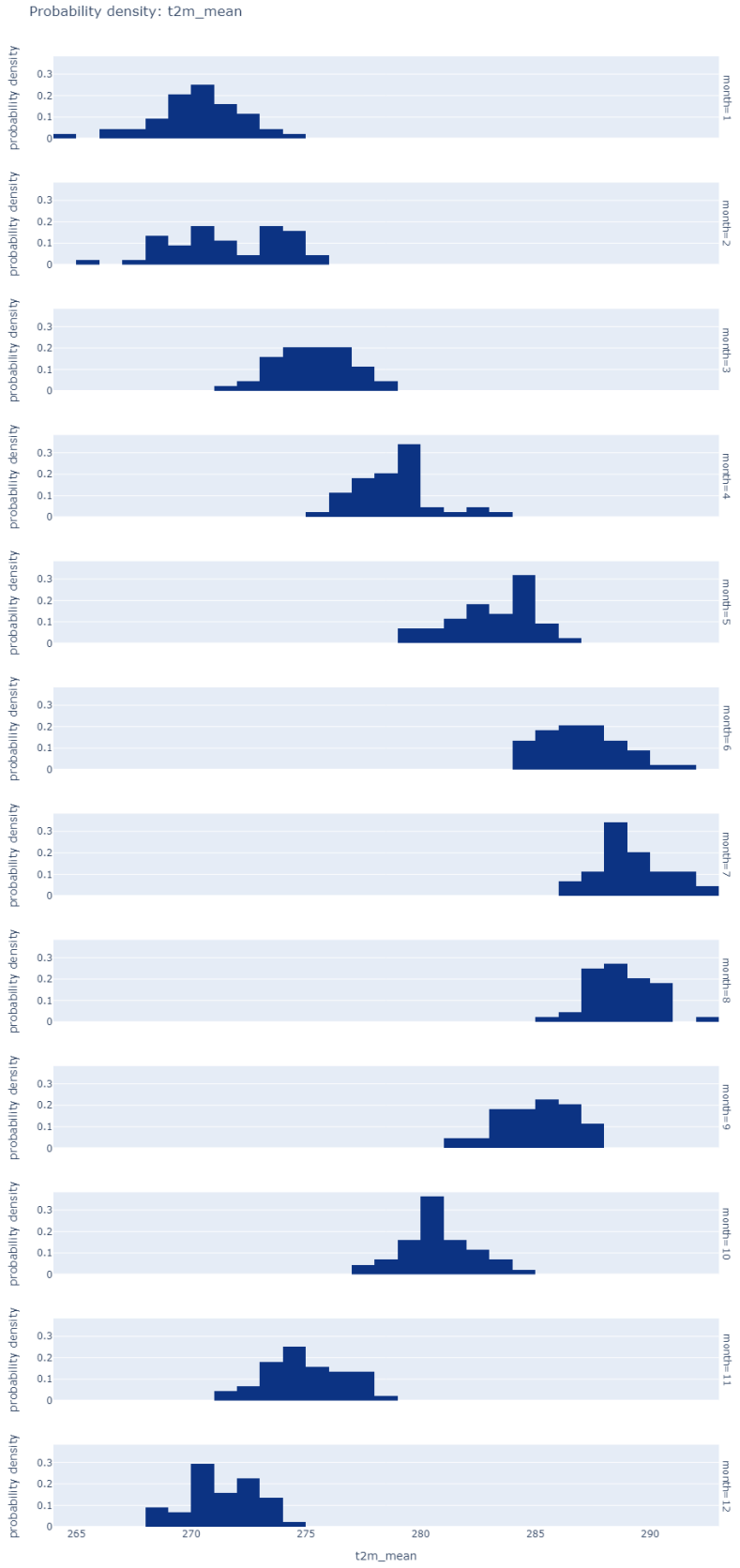

Figure 12 t2m_mean distribution

When plotting all datapoints as a box plot over a year, a clear curve is visibale, represensting the change of tempereature with each season or month. Febraury seems to be a bit of an outlier, having a wider spread .25 and .75 quantile. This most likely stems form the rise in temperature at the end of winter, which falls into February.

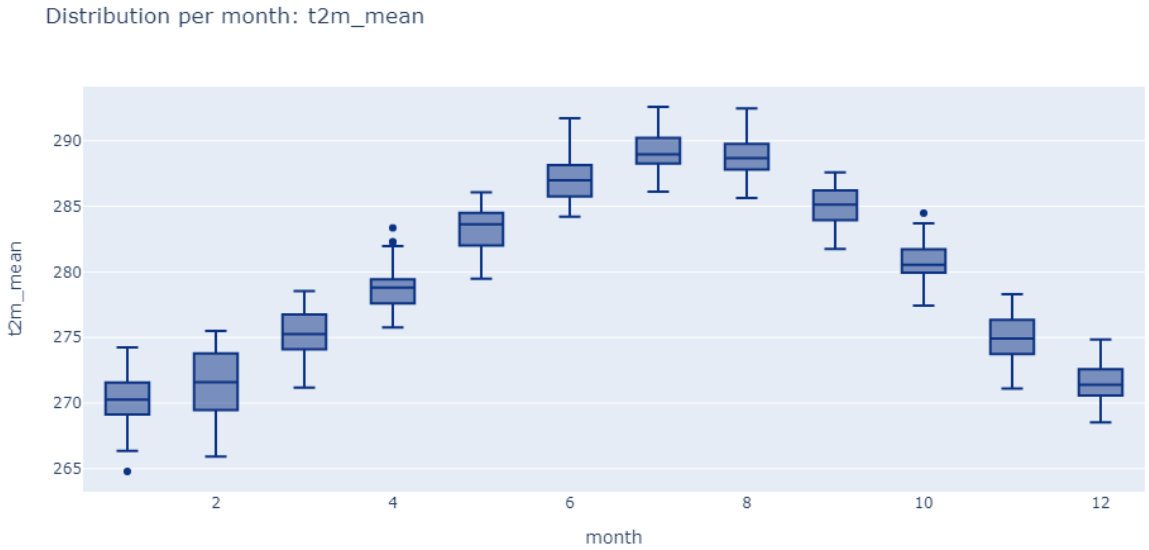

Figure 13 t2m_mean per month

Within the set time frame, the mean temperature curve contains some years with higher or lower peaks. If these are correlated with any of the available features will be seen later, when comparing the ENSO and MJO, as well as the polar vortex to the target array.

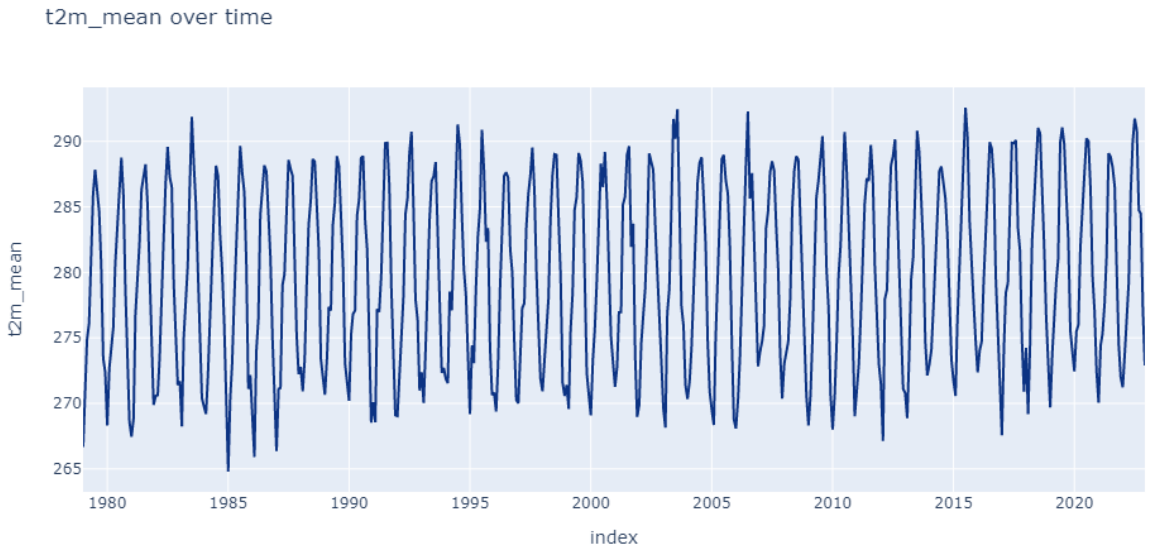

Figure 14 t2m_mean over time

Ignoring the concerning implications, when plotting the monthly mean over the years, a small but continuous upward trend can be spotted, hinting at rising average temperatures. This must be kept in mind when creating the validation and test set for the machine learning models.

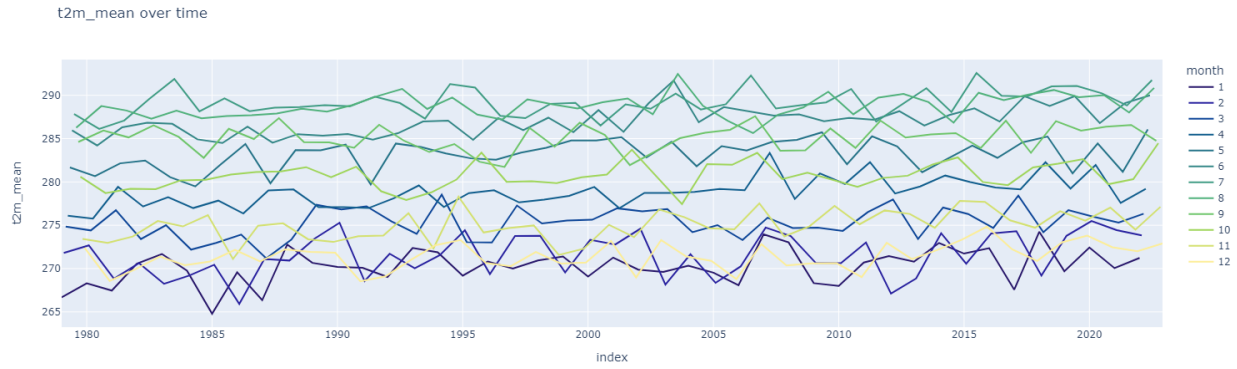

Figure 15 t2m_mean per month over time

As previously stated, the main two variables for predicting the t2m will be the ENSO and the MJO. Therefore, it is important to see if any surface level correlations can be spotted for outliers. If so, this could pose an additional feature for the machine learning models and aid in predicting anomalies. The anomalies or outliers will be analysed and identified with two methods:

- Fixed threshold of -1 and 1 (exclusively for the ENSO)
- Smoothed z-score algorithm with different parameters

First up is the ENSO. A plot over the whole time series reveals, that there are several peaks and drops. Because the ENSO has a limit of 3 and -3, it will always oscillate between these two values. When applying the z-score algorithm with the below mentioned parameters, the peaks can be detected somewhat accurately. But in this case, a simple threshold of 1 and -1 seems to do the trick better.

Z-score parameters: lag =24, threshold = 3.5, influence = 1/24

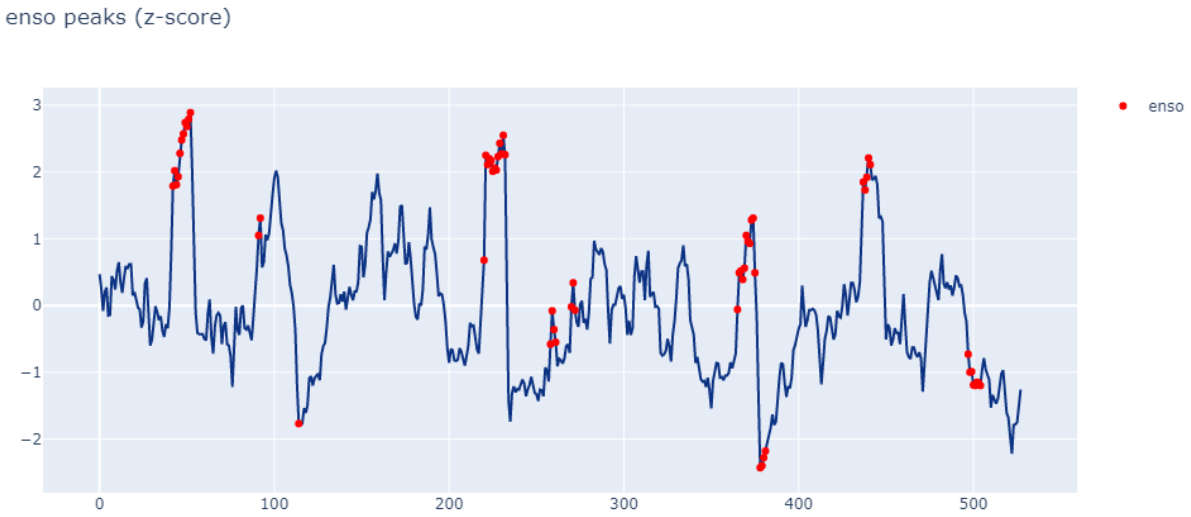

Figure 16 enso peaks (z-score)

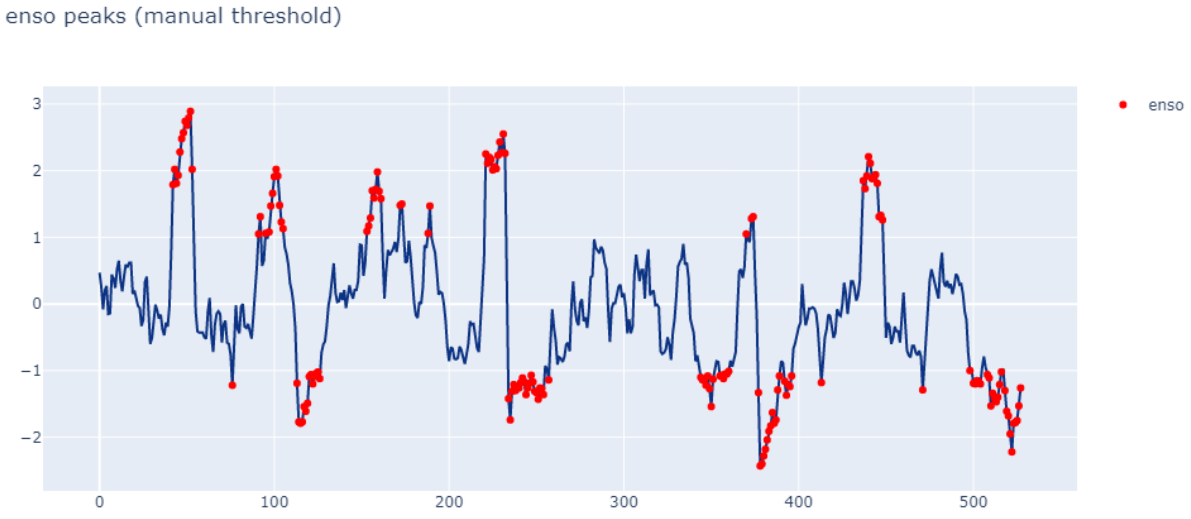

Figure 17 enso peaks (manual threshold)

Secondly, the same algorithm gets applied to the MJO data. Here, all higher positive peaks can be identified. A method with a manual threshold would not makes sense in this case, because of the nature of the oscillation.

Z-score parameters: lag =24, threshold = 2.5, influence = 1/24

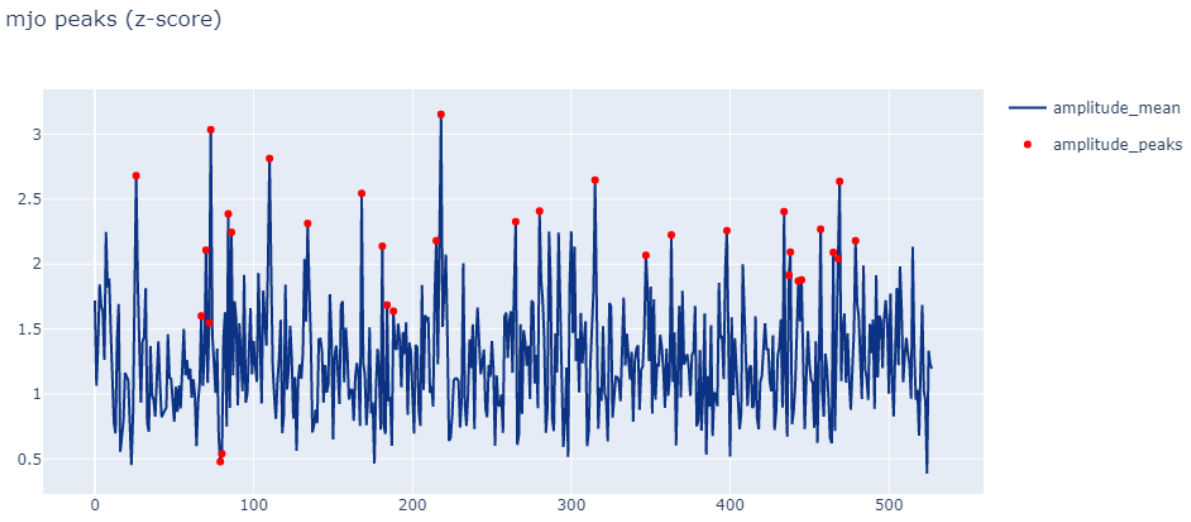

Figure 18 mjo peaks (z-score)

Although the effects of the ENSO and MJO are well documented and explored, this cannot be seen in the data at hand. When layering the peaks on top of the temperature data as a time series, and onto the probability density of each month, no clear pattern can be spotted. The main reason for this is probably the aggregation of the data. It is also probable, that the data sample for each month is too small to identify a trend or direct correlation.

Because the focus of this thesis is on the first two features, a closer analysis of the polar vortex data is out of scope. The same method could be applied to identify peaks and correlations in the data.

4.3 Feature engineering and data preparation (main_2)

To transform the date time indexes, the sine and cosine functions are applied on the time scale of a year. Meaning, one cycle of the function, refers to the passing of exactly one year. The values are added to existing data frame, expanding it by 2 additional features.

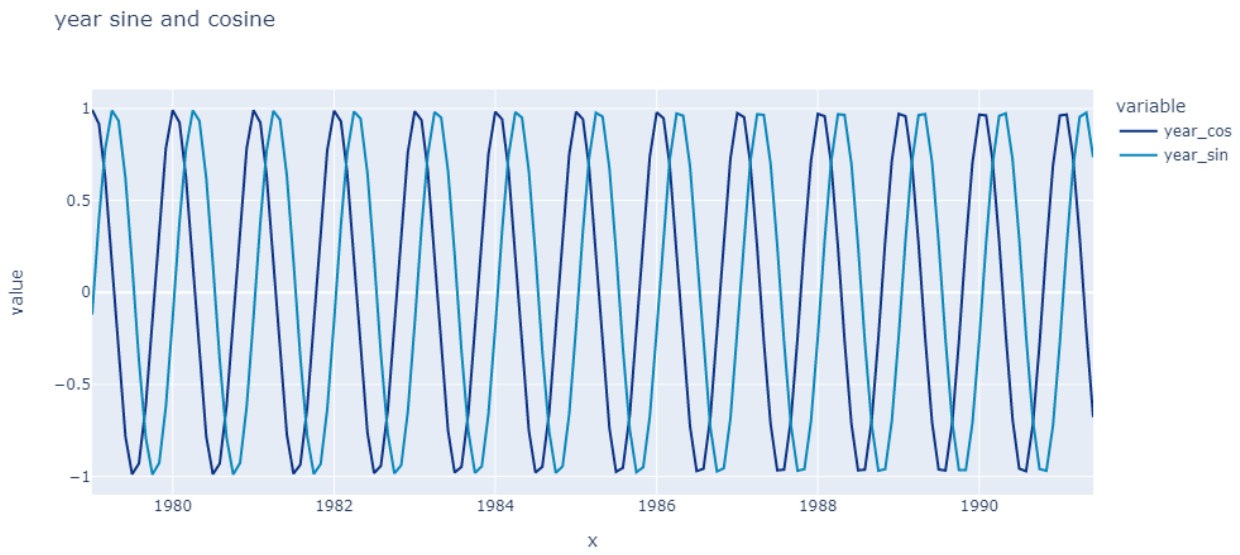

Figure 19 year sine and cosine

Furthermore, a standard and unweighted rolling mean with a time span of 4 is applied.

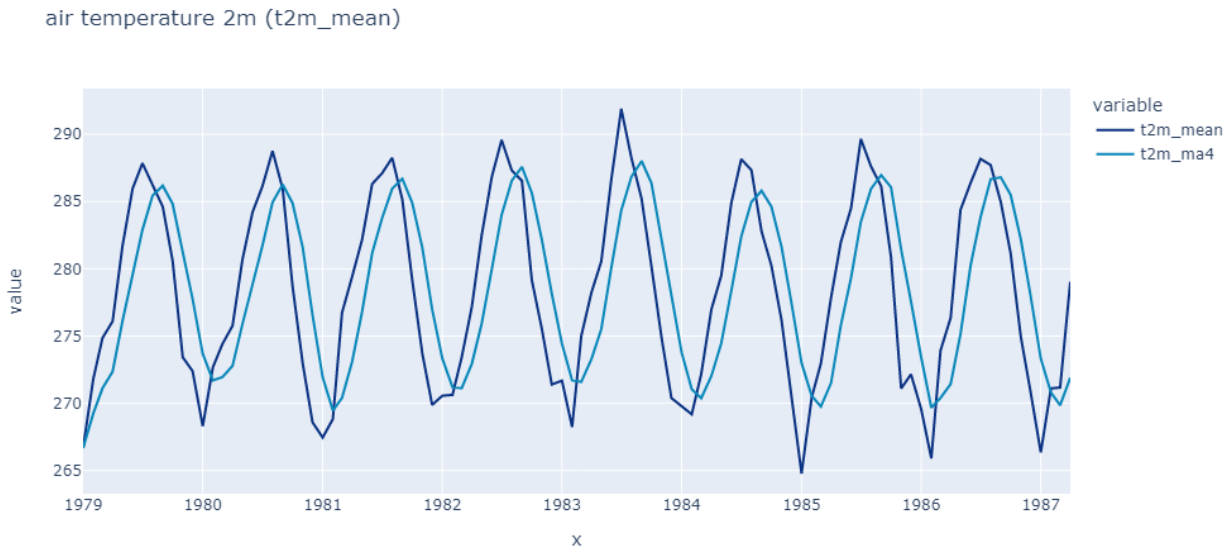

Figure 20 rolling mean on air temperature

To generate the classes for the machine learning models and classification of each month, the median mean air temperature is calculated and grouped by month. Meaning for each month, 22 datapoints will be classified as above and below average temperature. This results in a 50 / 50 split of the complete data set. The classes are labelled as 0 for below and 1 for above average.

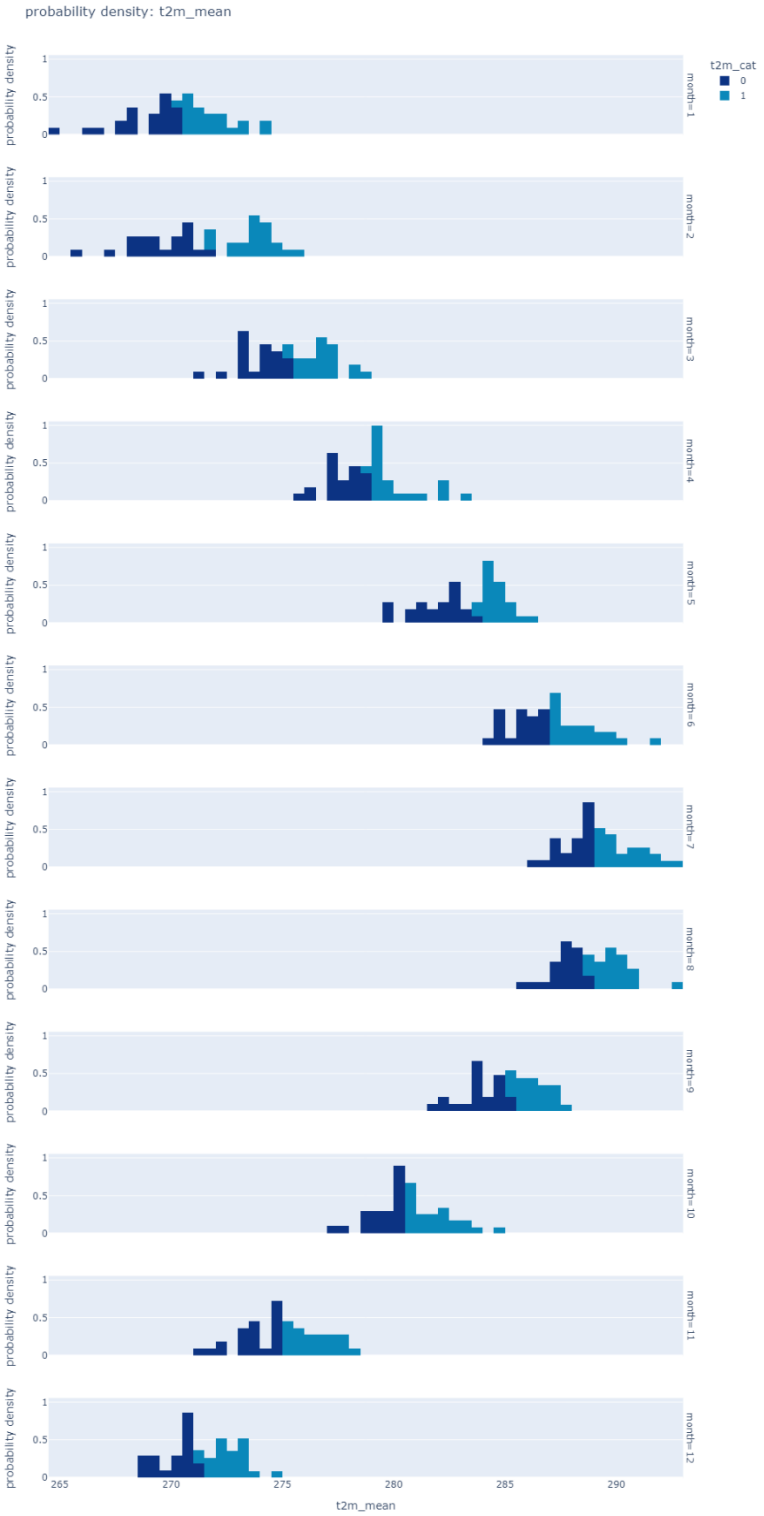

Figure 21 t2m mean classes

Because the sklearn library does not provide a built-in shift function for the target vector y, the offset is done manually by one time step. Important to note is, that the offset will result in a none value within the y vector. Thus, the last datapoint must be dropped.

Within the distribution of the two categories within each year, the same pattern emerges as with the temperature curve on the uncategorized data. The later in the time series a month is, the more likely it is that the mean temperature is above the median.

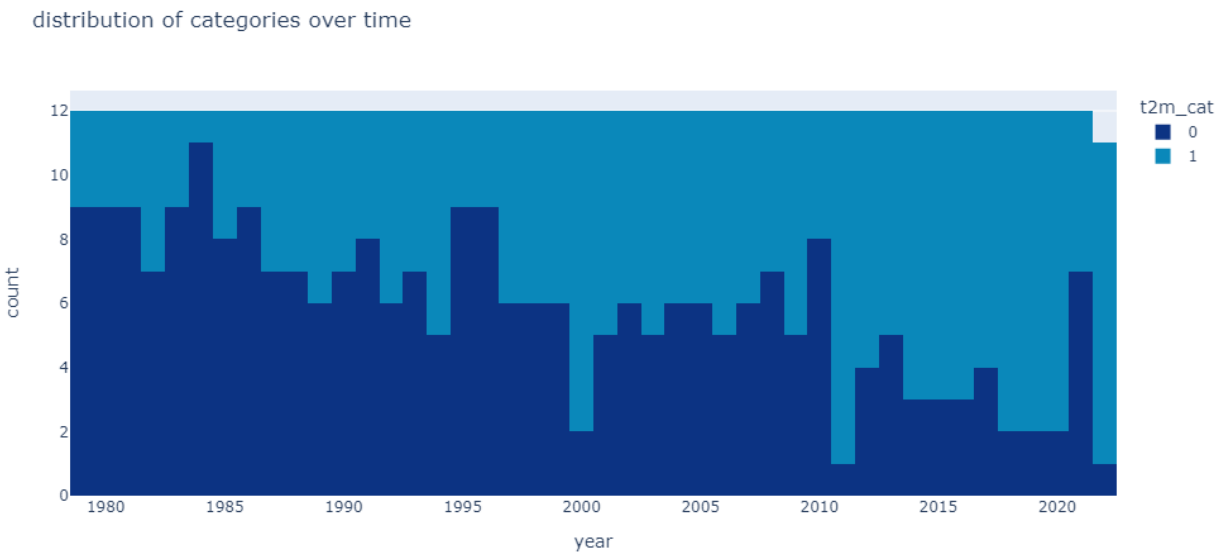

Figure 22 distribution of categories over time

After the data preparation and cleaning, the data frame contains the following columns and is saved as .csv file for later use.

index	1979-1	1979-2	1979-3	1979-4	1979-5	1979-6	1979-7	1979-8	1979-9	1979-10
year	1979	1979	1979	1979	1979	1979	1979	1979	1979	1979
month	1	2	3	4	5	6	7	8	9	10
enso	0.5	0.3	-0.1	0.2	0.3	-0.2	-0.1	0.4	0.4	0.2
pv_u_mean	-7.7	-10.1	1.0	-2.4	-2.1	-2.4	-3.0	-1.8	3.2	5.8
pv_u_std	27.3	23.4	11.5	4.5	2.1	2.4	2.1	2.4	4.6	7.0
pv_v_mean	-4.8	-19.5	13.4	-2.1	-0.8	-0.3	1.2	0.1	0.3	-1.2
pv_v_std	12.6	10.3	15.7	4.3	2.2	2.4	2.1	2.5	3.1	4.3
t2m_mean	266.7	271.8	274.8	276.1	281.7	286.0	287.8	286.3	284.6	280.6
t2m_std	4.6	3.1	2.9	3.6	5.6	3.9	4.2	4.3	4.8	3.5
rmm1_mean	-0.8	0.5	0.6	0.0	-0.1	-0.1	-0.7	-0.5	0.2	-0.1
rmm2_mean	1.1	-0.7	-0.3	-0.2	0.4	-0.7	0.1	0.6	-0.6	0.9
phase_mean	5.8	3.3	4.0	3.9	5.7	3.0	4.5	4.6	2.5	4.8
amplitude_mean	1.7	1.1	1.4	1.8	1.7	1.6	1.3	2.2	1.8	1.9
rmm1_std	0.8	0.5	1.2	1.4	1.3	1.3	0.9	1.8	1.7	1.5
rmm2_std	1.0	0.6	0.6	1.2	1.1	0.8	0.6	1.2	0.8	1.0
phase_std	2.4	1.0	1.8	2.3	1.9	1.6	3.0	2.7	1.5	2.6
amplitude_std	0.6	0.5	0.6	0.2	0.5	0.4	0.4	0.3	0.7	0.7
phase_mode	7	3	4	3	4	2	1	1	1	6
rmm1_last	0.1	0.9	0.3	-1.1	-2.0	0.5	1.2	-2.1	3.1	-0.7
rmm2_last	-0.6	-0.7	-1.2	-1.2	0.1	1.1	0.2	-1.0	1.5	-0.9
phase_last	3	4	3	2	8	6	5	1	5	2
amplitude_last	0.6	1.1	1.2	1.6	2.0	1.2	1.2	2.3	3.4	1.2
year_sin	-0.1	0.4	0.8	1.0	0.9	0.6	0.2	-0.4	-0.8	-1.0
year_cos	1.0	0.9	0.6	0.1	-0.4	-0.8	-1.0	-0.9	-0.6	-0.1
t2m_ma4	266.7	269.2	271.1	272.4	276.1	279.6	282.9	285.4	286.2	284.8
t2m_cat	0	1	0	0	0	0	0	0	0	1
t2m_cat_offset	1	0	0	0	0	0	0	0	1	0

Table 2 main data frame with engineered features

4.4 Model selection and model creation

4.4.1 Data splitting and standardizing

Bevor splitting the data into the train and validation set, the test set is removed, until the model is optimized, to see how the classifier generalizes on unseen data. Because the data is shifting into more above average data points, the later in the time series they are, it is not recommended to just take the first or last n data points. The split of the test set is done by setting the number of years which should be contained in the test set. Afterwards, the years are picked, so that they are distributed evenly among the dataset. This ensure, that there are different years in the test data set with various distributions of the data, while retaining an almost equal amount of above and below categorized data points in each set.

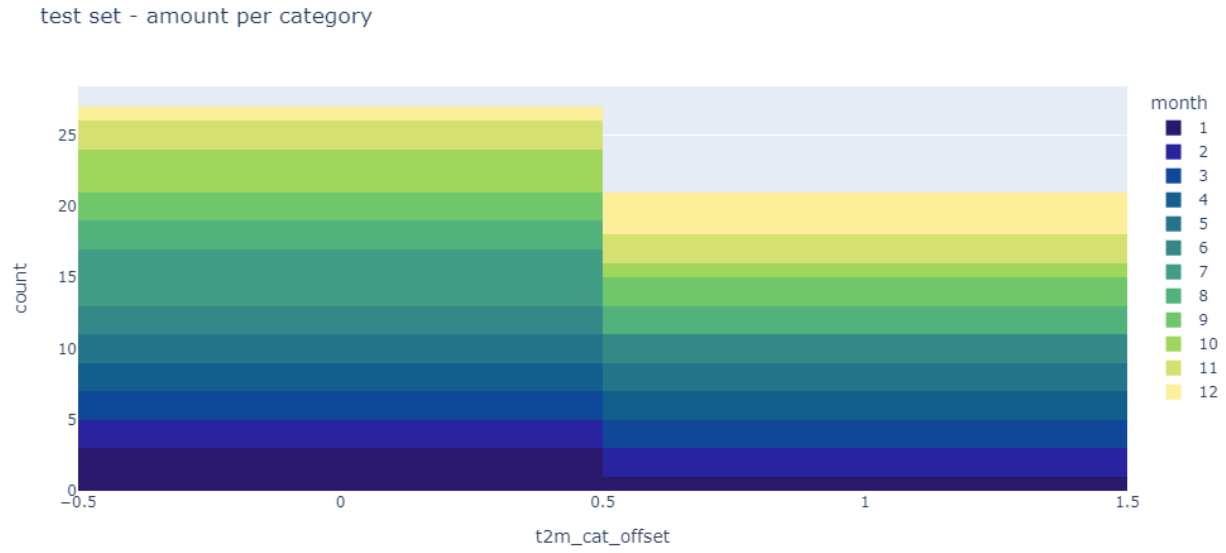

Figure 23 test set category distribution

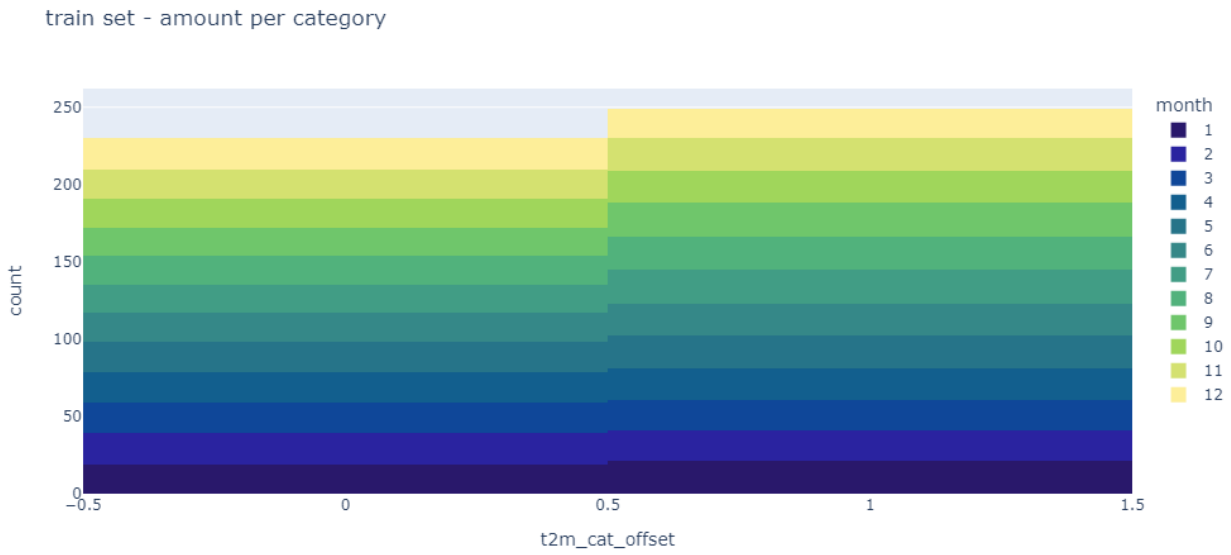

Figure 24 train set category distribution

The splitting of the validation and training set is done in two different ways, depending on the model, the machine learning library and if the model is only used for hyperparameter tuning or not. Three of the four machine learning models are created with sklearn library. The hyper parameter tuning is done with cross validation and partly normal validation. The folds of the cross validation are generated using the time series split method. This keeps the data in order and does not shuffle it randomly, which is important not to do when working with time series data.

The normal validation for optimization as well as training the model to get the test set scores is done by a custom splitter, taking into account the before mentioned distribution, depending on time. It works the same as the test set splitter.

Training the two neural networks (the multi-layer perceptron and recurrent neural network) require the data to be standardized. The standardizing on the training and validation set is done as described in chapter **Error! Reference source not found.**. The test set is also normalized when acquiring the model's s core, but with the mean and standard deviation from the train and validation set. This is common practice. [39]

Important to note is that when applying cross validation, the test score is the same as the validation score and is used interchangeably. This is due to the used sklearn library, which returns the validation accuracy in cross validation as test score.

4.4.2 Random forest (main_3)

The optimization of the random forest classifier is done in two separate steps. Firstly, all possible permutations for the number of estimators and the maximum depth is optimized as much as possible. To achieve this, two separate arrays are generated, with each element being calculated as 2^n, where n ranges from 0 to 12. Each array represents the n estimators and max depth respectively. The cross validation is used with 4 folds.

The results of this search are sobering. Either the model seems to overfit on the training data, and performs okay on the validation set, or the model does not overfit, and the accuracy of the model is as good as a random guess on the validation set.

	n_folds	n_estim	n_depth	test_score	train_score
62	4	64	8	0.5868	0.9978
86	4	256	8	0.5842	0.9978
64	4	64	32	0.5816	1.0000
65	4	64	64	0.5816	1.0000
66	4	64	128	0.5816	1.0000
67	4	64	256	0.5816	1.0000
68	4	64	512	0.5816	1.0000
69	4	64	1024	0.5816	1.0000

	n_folds	n_estim	n_depth	test_score	train_score
0	4	2	2	0.4605	0.6765
36	4	16	2	0.4737	0.6824
13	4	4	4	0.4763	0.7790
1	4	2	4	0.4895	0.7413
12	4	4	2	0.4921	0.6976
35	4	8	4096	0.4921	0.9530
34	4	8	2048	0.4921	0.9530
33	4	8	1024	0.4921	0.9530

Table 3 random forest grid search results

Considering the above-mentioned results, it is apparent, that the factor causing the overfitting is mainly the maximum depth (n_depth). But when cutting this parameter down too much, the model's performance suffers heavily. Looking at the fitting graph for this attribute, the described phenomenon can be spotted easily.

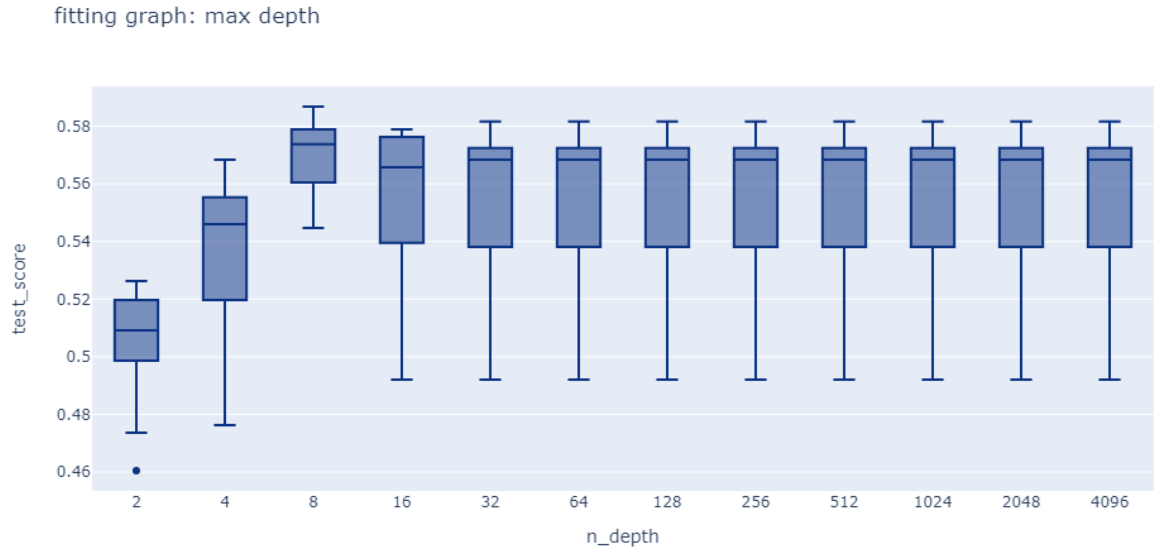

Figure 25 random forest classifier fitting graph

A grid search is run on the top result with 64 estimators and a maximum depth of 8, with the main goal of trying to prevent overfitting, while retaining an acceptable accuracy. The parameters and values run for the grid search are as follows:

Tested parameters:	min_samples_split: 2, 4, 6 min_samples_leaf: 1, 2, 4 max_features: auto, sqrt, log2 bootstrap: True, False
Optimal parameters:	n_estimators: 64 max_depth: 8 min_samples_split: 2 min_samples_leaf: 4 max_features: auto bootstrap: False

The results of the grid search, at first glance, seem to improve the model. With the above stated parameters, the random forest classifier obtains training accuracy of 0.98 and a validation accuracy of 0.73. When looking at the performance on the test set, it only yields an accuracy of 0.42. Considering the confusion matrix, it seems that the predictions are more or less random on the test set.

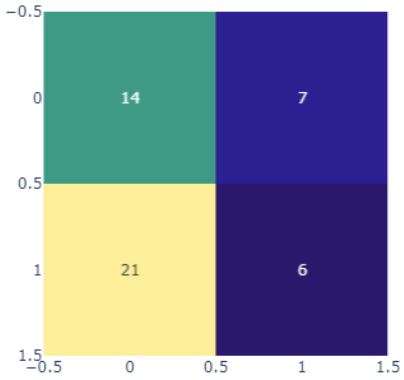

Figure 26 random forest confusion matrix / test set

Although the model does not generalize well, some insight still can be gained when plotting the feature importance of the best performing model. By far the most important feature is the year. Meaning the model made the correlation, stated in the data exploration. The rest of the features are similar in importance, hovering at around 0.02 to 0.05, compared to the year's 0.11 importance. Therefore, it is safe to say that the model relies heavily on the year as an indicator.

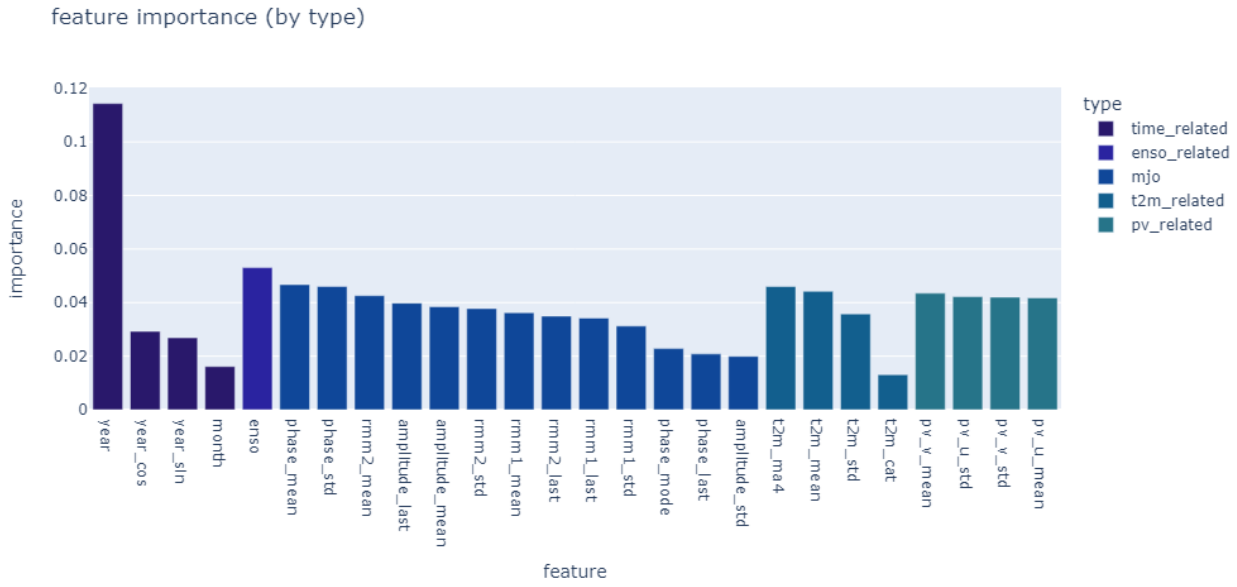

Figure 27 random forest feature importance

4.4.3 Multi-layer perceptron (main_4)

Before starting the optimization, the data is standardized as described in the previous chapter. One of the main parameters to optimize for the multi-layer perceptron model is the architecture, or more precisely: the hidden layer size. To generate the hidden layer sizes and be able to compare them easily with the cross validation, a custom function is used. It creates layers, based on the number of input features of the model, generating several nodes which are a multiple or a divisional of the number of input nodes. Hidden layer sizes of up to 4 layers are tested and its possible permutation under the mentioned restrictions. Cross validation is applied to verify the models.

When looking at the top results, it seems that the model does not overfit as strongly as the random forest does. But when comparing the top and bottom results, no clear pattern can be spotted among the different architecture.

	n_folds	test_score	train_score	arch
3306	4	0.6421	0.8742	[4, 3, 78, 13]
6470	4	0.6316	0.7767	[9, 5, 13, 2]
3292	4	0.6237	0.8700	[4, 3, 52, 4]
5208	4	0.6211	0.9198	[6, 2, 78, 52]
7161	4	0.6211	0.7391	[13, 2, 9, 3]
8554	4	0.6211	0.8450	[26, 6, 6, 6]
4031	4	0.6211	0.6323	[4, 78, 4, 3]
4720	4	0.6211	0.7608	[5, 13, 3, 2]
4650	4	0.6184	0.7156	[5, 9, 6, 2]
483	4	0.6184	0.8845	[5, 26, 5]

	n_folds	test_score	train_score	arch
150	4	0.3711	0.5315	[2, 6, 2]
1320	4	0.3711	0.5288	[2, 4, 3, 2]
6241	4	0.3763	0.5855	[9, 3, 5, 3]
9110	4	0.3816	0.5958	[52, 2, 2, 2]
1771	4	0.3842	0.5442	[2, 13, 13, 3]
2100	4	0.3895	0.5592	[2, 78, 78, 2]
4071	4	0.3895	0.7611	[4, 78, 13, 3]
2083	4	0.3895	0.5470	[2, 78, 26, 5]
2070	4	0.3921	0.7425	[2, 78, 13, 2]
1391	4	0.3921	0.5956	[2, 4, 52, 3]

Table 4 multi-layer perceptron grid search results

When plotting the number of hidden layers against the validation accuracy, a downward trend in the median score is suggested. But this could also be due to the bigger sample size within the higher layer count.

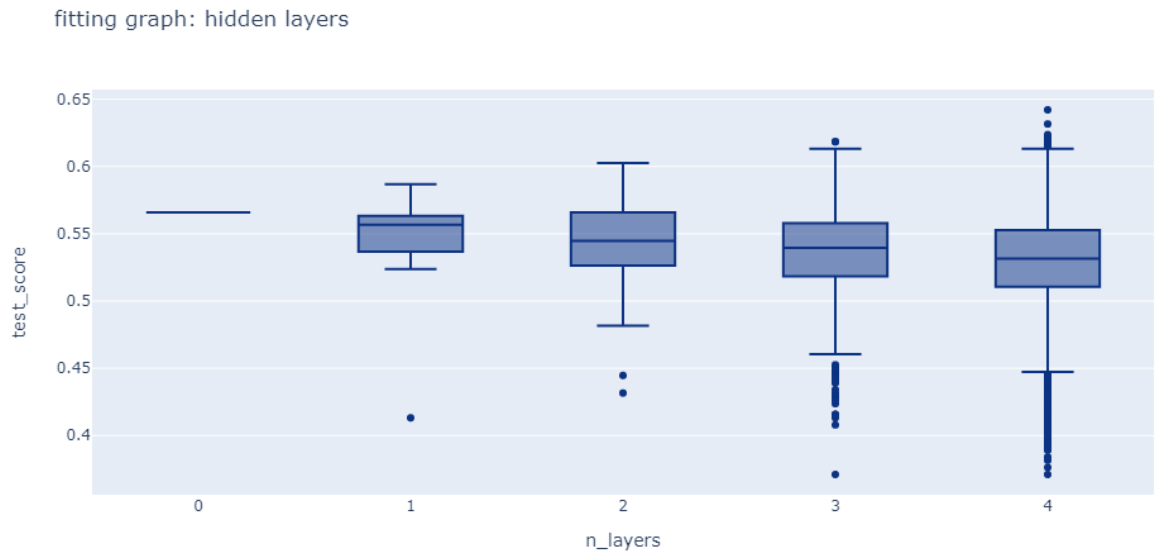

Figure 28 multi-layer perceptron fitting graph: layers

Trying to find a pattern and gain some understanding of the behaviour and correlations of the models, the mean number of nodes per layer are calculated and plotted against the validation accuracy. The more nodes in each layer, the better the performance on the training set. The same cannot be said about the validation accuracy, which does not seem to follow a particular pattern. The cone shape of the plot is mostly due to the number of available samples.

Figure 29 multi-layer perceptron fitting graph

Since the models already performed poorly with default values and heavy optimization tries, a grid search is not conducted on the multi-layer perceptron. The top performing model parameters are as follows:

Optimal parameters: hidden_layer_size: [4, 3, 78, 13]
activation: relu
solver: adam
alpha: 0.1
max_iter: 500

The resulting model yields a training accuracy of 0.67, a validation accuracy of 0.70 and a test accuracy of 0.44. But when inspecting the confusion matrix, it is apparent that the model predicts the same category (below average) for all entries in the test set. Even when testing other top models, the same result occurs on the testing set.

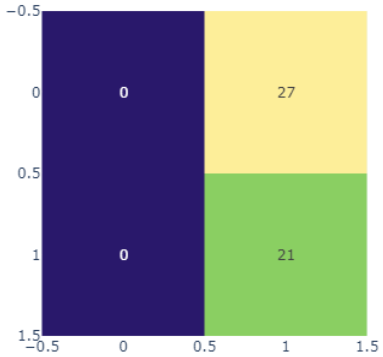

Figure 30 multi-layer perceptron confusion matrix

4.4.4 Recurrent neural network (main_5)

Being that the recurrent neural network is based on the keras library, which itself is a high-level API for tensorflow, the optimisation process is written up a bit different from the rest of the classifiers. Due to

the complexity of the network, the training takes longer than the multi-layer perceptron. Thus, only a smaller number of hidden layer sizes are tested. Two types of architectures are relevant for this part of the optimisation:

- Cone shaped architectures, which start out with many neurons and gradually decrease the number of nodes with each following layer.
- Linear architectures, which have the same number of nodes on each layer.

Once again, the data is standardized. The customization of the recurrent neural networks allows for numerous changes and permutations. Most of the parameters are therefore fixed from the beginning, as it would take up too much time to optimize each of them.

One problem, which became apparent quickly, is the low amount of data points. Numerous of the models only produce predictions for one or the other category. Only a small fraction of them do not create same category predictions. But the performance is only as good as a random guess.

	model_type	train_score	valid_score	arch
8	RNNC_SEQ	0.52	0.48	[13, 13, 13]
12	RNNC_SEQ	0.49	0.47	[13, 13, 13, 13]
6	RNNC_SEQ	0.47	0.42	[52, 52]
18	RNNC_SEQ	0.50	0.39	[52, 52, 52, 52, 52]
15	RNNC_SEQ	0.48	0.37	[512, 256, 128, 64]
16	RNNC_SEQ	0.47	0.30	[13, 13, 13, 13, 13]

Table 5 recurrent neural network top results

The model parameters of the best model are:

Optimal parameters:
Model type: sequential
Input layer: lambda
hidden layers: 3 x bidirectional lstm, 13 cells each
activation: selu
loss: binary crossentropy
optimizer: sgd, learning rate = 0.1
shuffle: false
epochs: 10

When applying top 3 models onto the test set, the same problem arises again. The neural networks predicted the same category repeatedly.

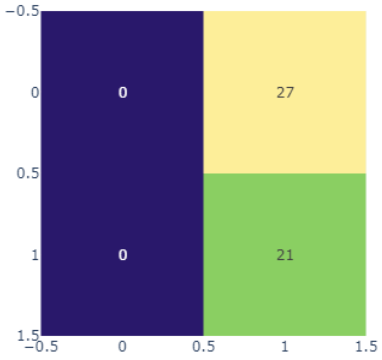

Figure 31 recurrent neural network confusion matrix

4.4.5 Nu support vector machine (main_6)

For the nu support vector machine, a manually coded grid search is run which focuses on the main parameters: kernel, degree, and nu. The degree parameter only will be taken into account, if the kernel is set to polynomial. The values for the tested parameters are as follows:

Tested parameters: kernel: linear, rbf, sigmoid, polynomial
degree: 1,2,3,4,5
nu: 0.1, 0.2, …, 0.9

There is a possibility, that some of the higher polynomial kernels will not generalize well, henceforth four optimal models are created, one for each kernel. Each will yield its optimal parameters. The grid search reveals some interesting insight. It seems, that all top models from the different kernels perform over 10% better on the validation set, compared to the training set. This could hint at some irregularity within the data the model has picked up on. Apart from that, the model performances look better than the neural network models and the random forest. The search result suggests that the model does not overfit in contrast to the random forest classifier.

	n_folds	kernel	degree	nu	test_score	train_score
46	0	poly	5	0.4	0.774648	0.649510
63	0	poly	2	0.8	0.746479	0.647059
7	0	linear	0	0.8	0.746479	0.642157
22	0	sigmoid	0	0.5	0.746479	0.637255
21	0	sigmoid	0	0.4	0.746479	0.627451
6	0	linear	0	0.7	0.732394	0.644608
8	0	linear	0	0.9	0.732394	0.637255
17	0	rbf	0	0.9	0.718310	0.639706
23	0	sigmoid	0	0.6	0.718310	0.625000
52	0	poly	1	0.6	0.718310	0.625000
57	0	poly	1	0.7	0.718310	0.595588

Table 6 nu support vector machine grid search results

A pattern can be spotted, when plotting the nu against the validation, regardless of kernel or polynomial degree. The median validation accuracy seems to increase the higher the nu is. But it is suspicious, that the models with a nu of 0.9 seem to perform all equally well. A possible explanation could be, that the model creates same guesses in combination with a badly distributed validation and training set. But this is not the case, because these datapoints are cleared from the results table.

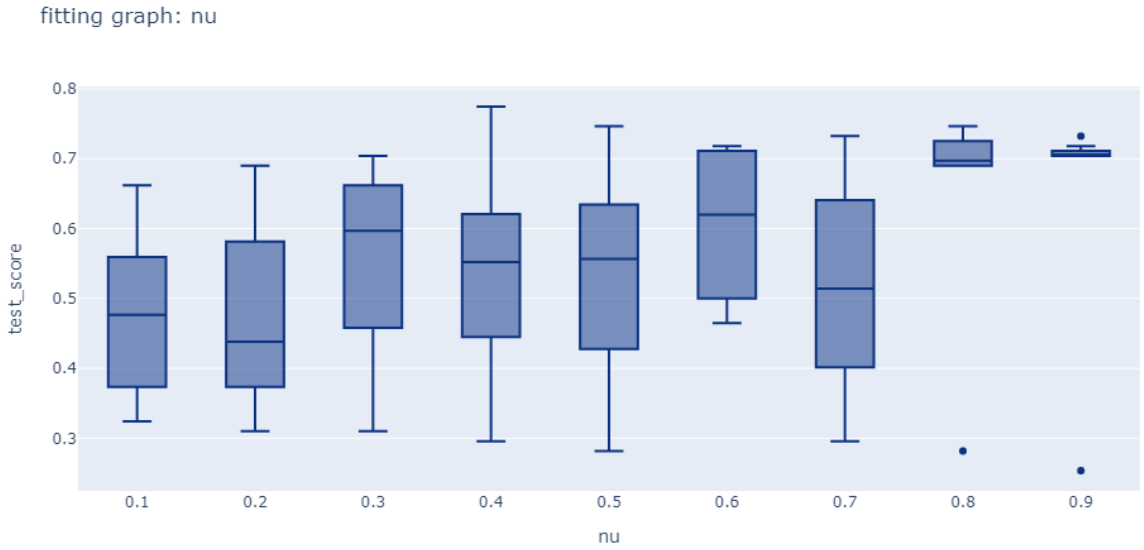

Figure 32 nu support vector machine fitting graph: nu

Another indication that the nu support vector machine classifier generalises better than any of the previous models, is the comparison of the training and validation score. It is apparent, that these two increase or decrease in a linear fashion to one another. Note that on the scatter plot (Figure 32) the size of the dot represents the nu.

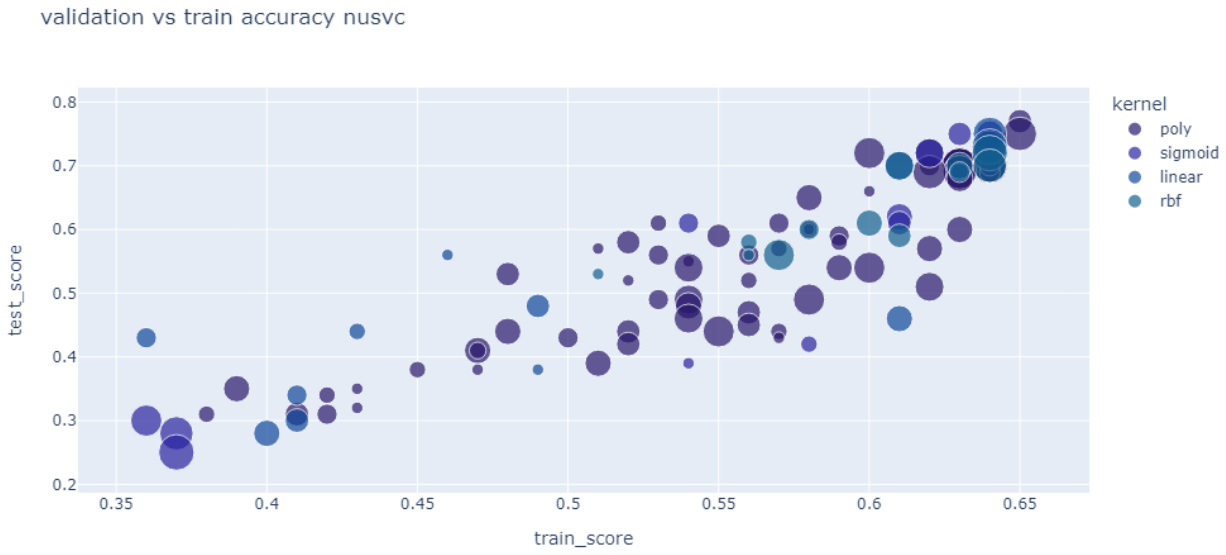

Figure 33 nu support vector machine validation vs train score

The model parameters for each of the kernels are as follows:

Optimal parameters:

kernel: linear
nu: 0.8

kernel: rbf
nu: 0.9

kernel: sigmoid
nu: 0.5

kernel: polynomial
nu: 0.4
degree: 5

The performance on the test set, are once again, sobering. None of the above listed models exceeds an accuracy of 0.52 on the test. Henceforth, none of them generalises well on the test set.

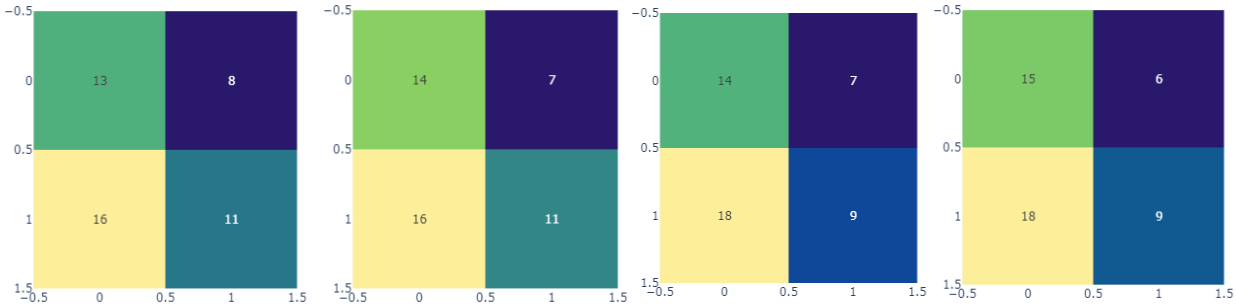

Figure 34 confusion matrix nsvm (from left to right): polynomial, sigmoid, linear, rbf

5 Conclusion

5.1 General findings

Against expectation, the artificial neural network performed worse than the random forest model and nu support vector machine. Although, the literatures research suggests, that neural networks are very well fitted for time series predictions. Two of the contributing factors to these circumstances are probably:

- Small sample size with only around 530 total data points
- The complexity of the models compared to the sample size

Another contributing factor for the overarching bad performance of all models could be the data aggregation. It is possible, that the data was aggregate down too much, for it to hold any information on clear patterns and correlation. This was also the finding during the data exploration on the MJO and ENSO while layering them on top of each month's data.

5.2 Discussion and research questions

Taking reference to the research questions, proposed in chapter **Error! Reference source not found.**, t he following can be stated.

What data is available and suitable for the forecasting of the next month's average temperature and how does the data need to be transformed?

- The available data, when aggregated as described, does not yield enough information to compile a comprehensible model which is able to create accurate forecasts of the mean temperature of the next month. Some outliers and peaks can be spotted in the ENSO and MJO time series data, but these do not seem to correlate or cause any temperature anomalies. This could be due to the high data aggregation to monthly data points.

How can the heating demand for the next month be forecasted, based on the temperature, categorized by below or above average temperatures (classification model)?

- The heating demand, based on outside temperature, can currently not be forecasted by any of the constructed machine learning models, which include: random forest, multi-layer perceptron, recurrent neural network, nu support vector machine.

What is the forecast accuracy and how can it be improved further (influence factors)?

- The forecast accuracy of all the models barely exceeds 0.5. But this is most likely due to the test set size. If more testing data would be provided, the models are likely to converge to an accuracy of 0.5.

Henceforth, the task of prediction temperature remains a hard, if not impossible to reach goal. At least for longer time frames than about 2 weeks. Even if the forecast is based on a simple classification on historical data.

5.3 Further steps and bachelor thesis

Because this project poses as a pre-project for the another's bachelor thesis, the following further steps could be considered:

- As previously stated, the resulting number of data points when aggregated down, seems to be not enough for the models, and especially the neural networks to converge. Therefore, the data needs to be aggregated less. Two possible ways would be a daily aggregation and creating multi-step forecasting models or create rolling months within intervals of calendar weeks.

- Adding more features would also be a way to improve the models. There are two options which could turn out to be interesting:
    - Additional temperature and wind speed data of the surrounding countries, in form of a square grid. These two factors could be used to predict weather and temperature fronts from every direction.
    - More detailed polar vortex data in form of grid, starting at the north pole and continuing to central Europe. The described effects of the polar vortex on weather could be deduced more precisely.

- Testing additional models and comparing them to the tested ones. A suitable neural network, as stated in chapter 2.2.1, could be a convolutional neural network. But if the results and assumptions are true, more data still has to be provided for training and validation the new neural network.

- Trying to reduce the model complexity, by cutting down on input features could be an interesting take. But this would only be a viable strategy, if enough datapoints are available.

- Further investigate the nu support vector machine model:
    - As seen, the accuracy-spread on the model with a nu of 0.9 is very thin. This could hint at a possible significant prediction model.
    - Increasing the polynomial degree to a value higher than 5 and compare the results.

- Further development of the recurrent neural network:
    - Implement a shifting window function into the model. This would not only feed one data point at a time, but also the last n data points.

6 Literature

[1] Federal Department of Foreign Affairs FDFA, "Energy – Facts and Figures," 07 02 2023. [Online]. Available: https://www.eda.admin.ch/aboutswitzerland/en/home/wirtschaft/energie/energie---fakten-und-zahlen.html. [Accessed 21 02 2023].

[2] World Economic Forum, "These charts show Europe's reliance on gas before the war in Ukraine," World Economic Forum, 10 11 2022. [Online]. Available: https://www.weforum.org/agenda/2022/11/europe-gas-shortage-russia/. [Accessed 21 02 2023].

[3] Gaznat, "Supply," [Online]. Available: https://www.gaznat.ch/en-39-supply.html. [Accessed 05 03 2023].

[4] Swiss federal office of energy, "Swiss energy consumption up 6.3% in 2021," Swiss federal office of energy, 23 06 2022. [Online]. Available: https://www.bfe.admin.ch/bfe/en/home/news-and-media/press-releases/mm-test.msg-id-89418.html. [Accessed 21 02 2023].

[5] Swissinfo SWI, "Some 58% of Swiss buildings heated with fossil fuels," 06 10 2022. [Online]. Available: https://www.swissinfo.ch/eng/business/some-58--of-swiss-buildings-heated-with-fossil-fuels/47958720. [Accessed 21 02 2023].

[6] Federal Office fro National Exonomic, "Natural Gas," 30 01 2023. [Online]. Available: https://www.bwl.admin.ch/bwl/en/home/themen/energie/erdgas.html. [Accessed 05 03 2023].

[7] K. Lylykangas, "Shape Factor as an Indicator of Heating," Internationales Holzbau-Forum, Helsinki, Finland, 2009.

[8] scijinks, "How Reliable Are Weather Forecasts?," [Online]. Available: https://scijinks.gov/forecast-reliability/. [Accessed 21 02 2023].

[9] Raleigh NC, Weather Forecast Office, "The Forecast Process: Observing and Analysis," National weather service, [Online]. Available: https://www.weather.gov/rah/virtualtourfcstobsanalysis. [Accessed 25 02 2023].

[10] Metoffice, "Polar Vortex," [Online]. Available: https://www.metoffice.gov.uk/weather/learn-about/weather/atmosphere/polar-vortex. [Accessed 22 02 2023].

[11] D. J. Cohen, "Arctic Oscillation and Polar Vortex Analysis and Forecasts," Versik, 13 02 2023. [Online]. Available: https://www.aer.com/science-research/climate-weather/arctic-oscillation/. [Accessed 22 02 2023].

[12] European Centre for Medium-Range Weather Forecasts , "ERA5," [Online]. Available: https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era5. [Accessed 22 02 2023].

[13] National center for enviromental information, "El Niño/Southern Oscillation (ENSO)," [Online]. Available: https://www.ncei.noaa.gov/access/monitoring/enso/technical-discussion . [Accessed 22 02 2023].

[14] Climate.gov, "El Niño & La Niña (El Niño-Southern Oscillation)," 09 03 2022. [Online]. Available: https://www.climate.gov/enso. [Accessed 27 03 2023].

[15] J. Gottschalck, "What is the MJO, and why do we care?," Climate.gov, 31 12 2014. [Online]. Available: https://www.climate.gov/news-features/blogs/enso/what-mjo-and-why-do-we-care. [Accessed 22 02 2023].

[16] S. Z. Bryan Lim, "Time-series forecasting with deep learning: a survey," *The royal society,* vol. 379, no. 2194, 2021.

[17] L. B. K. A. M. F. N. G. M. L. J. R. E. R. U. S. U. S. M. T. T. P. Barnett, "Forecasting global ENSO-related climate anomalies," *Tellus,* vol. 46, no. 4, pp. 381-397, 1994.

[18] M. W. K. W. F. V. N. S. H. L. H. H. D. W. K. S. M. N. C. P. M. F. a. W. H. J. Gottschalck, "A Framework for Assessing Operational Madden–Julian Oscillation Forecasts," *Arican Meteorological Society,* vol. 91, no. 9, pp. 1247-1258, 2010.

[19] B. S. B. G. R. H. Donald M. Lafleur, "Some Climatological Aspects of the Madden–Julian Oscillation (MJO)," *Journal of Climate,* vol. 28, no. 15, p. 6039–6053, 2015.

[20] G. T. H. &. K. D. Papacharalampous, “Predictability of monthly temperature and precipitation using automatic time series forecasting methods,” *Acta Geophys,* vol. 66, no. 1, pp. 807-831, 2018.

[21] S. M. B. S. J. K. H. V. Trang Thi Kieu Tran, “A Review of Neural Networks for Air Temperature Forecasting,” *Water,* vol. 13, no. 1294, pp. 1-15, 2021.

[22] Z. M. Mohsen Hayati, “Application of Artificial Neural Networks for,” *World Academy of Science, Engineering and Technology,* vol. 28, no. 1, pp. 275-279, 2007.

[23] K. Beckmann, “Energy Demand for Heating,” ClimateXChange, Edinburgh, 2015.

[24] P. M. A. B. G. G. Selin Yilmaz, “Hourly demad profiles for space heating and electricity,” University of Geneva, EMPA, ETH Zurich, 2020.

[25] S. Gupta, “Data Science Process: A Beginner’s Guide in Plain English,” Springboard, 16 5 2022. [Online]. Available: https://www.springboard.com/blog/data-science/data-science-process/. [Accessed 2 10 2022].

[26] scikit learn, “Support Vector Machines,” [Online]. Available: https://scikit-learn.org/stable/modules/svm.html. [Accessed 29 03 2023].

[27] Z. Brodtman, “The Importance and Reasoning behind Activation Functions,” Towards Data Science, 15 11 2021. [Online]. Available: https://towardsdatascience.com/the-importance-and-reasoning-behind-activation-functions-4dc00e74db41. [Accessed 30 03 2023].

[28] T. Böhm, “A first Introduction to SELUs and why you should start using them as your Activation Functions,” Towards Data Science, 28 08 2018. [Online]. Available: https://towardsdatascience.com/gentle-introduction-to-selus-b19943068cd9. [Accessed 30 03 2023].

[29] papers with code, “Scaled Exponential Linear Unit,” [Online]. Available: https://paperswithcode.com/method/selu. [Accessed 30 03 2023].

[30] J. Fernando, “Moving Average (MA): Purpose, Uses, Formula, and Examples,” Investopedia, 09 01 2023. [Online]. Available: https://www.investopedia.com/terms/m/movingaverage.asp. [Accessed 26 02 2023].

[31] A. Loo, “Exponential Moving Average (EMA),” Corporate Finance Institute CFI, 16 01 2023. [Online]. Available: https://corporatefinanceinstitute.com/resources/capital-markets/exponential-moving-average-ema/. [Accessed 26 02 2023].

[32] J. Fernando, “Relative Strength Index (RSI) Indicator Explained With Formula,” Investopedia, 15 07 2022. [Online]. Available: https://www.investopedia.com/terms/r/rsi.asp. [Accessed 26 02 2023].

[33] P.-L. Bescond, “Cyclical features encoding, it’s about time!,” Towards Data Science, 08 06 2020. [Online]. Available: https://towardsdatascience.com/cyclical-features-encoding-its-about-time-ce23581845ca. [Accessed 26 02 2023].

[34] baeldung, “Normalizing Inputs for an Artificial Neural Network,” 16 11 2022. [Online]. Available: https://www.baeldung.com/cs/normalizing-inputs-artificial-neural-network. [Accessed 26 02 2023].

[35] J. Brakel, “Robust peak detection algorithm using z-scores,” Stack Overflow, 2014. [Online]. Available: https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data/43512887#43512887. [Accessed 29 03 2023].

[36] R. Vickery, “8 Metrics to Measure Classification Performance,” Towards Data Science, 07 12 2021. [Online]. Available: https://towardsdatascience.com/8-metrics-to-measure-classification-performance-984d9d7fd7aa. [Accessed 27 03 2023].

[37] T. Yiu, “Understanding Random Forest,” Towards Data Science, 12 06 2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2. [Accessed 02 04 2023].

[38] Study Machine Learning, “Support Vector Machine (SVM),” [Online]. Available: https://studymachinelearning.com/support-vector-machine-svm/. [Accessed 02 04 2023].

[39] C. Bento, “Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis,” Towards Data Science, 21 09 2021. [Online]. Available: https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141. [Accessed 02 04 2023].

[40] B. Whitfield, “A Guide to Recurrent Neural Networks: Understanding RNN and LSTM Networks,” built in, 28 02 2023. [Online]. Available: https://builtin.com/data-science/recurrent-neural-networks-and-lstm. [Accessed 02 04 2023].

[41] “Copernicus,” [Online]. Available: https://www.copernicus.eu/en. [Accessed 02 04 2023].

[42] “Multivariate ENSO Index Version 2 (MEI.v2),” Physical Sciences Laboratory, [Online]. Available: https://psl.noaa.gov/enso/mei/. [Accessed 02 04 2023].

[43] Australian Government, “Madden-Julian Oscillation (MJO),” [Online]. Available: http://www.bom.gov.au/climate/mjo/. [Accessed 02 04 2023].

[44] ه. جمال, “Atmospheric pressure and the factors affecting it and the distribution of its systems through this report,” Arabia weather, 27 10 2020. [Online]. Available: https://www.arabiaweather.com/en/content/atmospheric-pressure-and-the-factors-affecting-it-and-the-distribution-of-its-systems-through-this-report. [Accessed 29 03 2023].

[45] madur, “Calculator: elevation → pressure,” madur, [Online]. Available: https://www.madur.com/index.php?page=/altitude. [Accessed 29 03 2023].

[46] B. Giba, “When, Why, And How You Should Standardize Your Data,” Machinelearning compass, 06 07 2021. [Online]. Available: https://machinelearningcompass.com/dataset_optimization/standardization/. [Accessed 30 03 2023].

[47] J. Brownlee, “A Tour of Machine Learning Algorithms,” Machine Learning Mastery, 12 8 2019. [Online]. Available: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/. [Accessed 2 10 2022].

7 Table of figures

Figure 1 Cover Image, source https://www.umweltbundesamt.de/en/topics/health/environmental-impact-on-people/special-exposure-situations/emissions-from-wood-coal-burning-stoves-in#use-of-wood-and-coal-stoves-in-residential-areas 1

Figure 2 Power consumption by source 4

Figure 3 Heating energy consumption by source 5

Figure 4 ENSO, source: https://www.climate.gov/enso 6

Figure 5 MJO, source: https://www.climate.gov/news-features/blogs/enso/what-mjo-and-why-do-we-care 7

Figure 6 Polar vortex, source: https://www.latimes.com/california/story/2020-03-28/if-a-warm-u-s-winter-was-a-preview-of-global-warming-what-part-did-a-polar-vortex-play 7

Figure 7 confusion matrix, source : https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 13

Figure 8 Switzerland outline for temperature measurements 15

Figure 9 Geolocation of polar vortex data 16

Figure 10 MJO data sample, source: http://www.bom.gov.au/climate/mjo/ 16

Figure 11 Correlation matrix 18

Figure 12 t2m_mean distribution 19

Figure 13 t2m_mean per month 20

Figure 14 t2m_mean over time 20

Figure 15 t2m_mean per month over time 21

Figure 16 enso peaks (z-score) 21

Figure 17 enso peaks (manual threshold) 22

Figure 18 mjo peaks (z-score) 22

Figure 19 year sine and cosine 23
Figure 20 rolling mean on air temperature 23
Figure 21 t2m mean classes 24
Figure 22 distribution of categories over time 25
Figure 23 test set category distribution 27
Figure 24 train set category distribution 27
Figure 25 random forest classifier fitting graph 29
Figure 26 random forest confusion matrix / test set 30
Figure 27 random forest feature importance 30
Figure 28 multi-layer perceptron fitting graph: layers 31
Figure 29 multi-layer perceptron fitting graph 32
Figure 30 multi-layer perceptron confusion matrix 32
Figure 31 recurrent neural network confusion matrix 33
Figure 32 nu support vector machine fitting graph: nu 35
Figure 33 nu support vector machine validation vs train score 35
Figure 34 confusion matrix nsvm (from left to right): polynomial, sigmoid, linear, rbf 36

8 Table of tables

Table 1 merged data frame 17
Table 2 main data frame with engineered features 26
Table 3 random forest grid search results 28
Table 4 multi-layer perceptron grid search results 31
Table 5 recurrent neural network top results 33
Table 6 nu support vector machine grid search results 34

9 Declaration of Authorship

I hereby certify that I composed this work completely unaided, and without the use of any other sources or resources other than those specified in the bibliography or the source code. All text sections not of my authorship are cited as quotations, and accompanied by an exact reference to their origin.

Place, date: _____

Signature, Joël Tauss: _____