Improving Weather Forecasting via SSW Characterization

Marcel Bühler
Department of Computer Science
ETH Zurich
buehlmar@ethz.ch

Dimitrios Koutsoukos
Department of Computer Science
ETH Zurich
dkoutsou@ethz.ch

Orhun Özbek
Department of Computer Science
ETH Zurich
oezbeko@ethz.ch

Abstract

When Europeans experience a heavy temperature drop in winter, chances are that a Sudden Stratospheric Warming (SSW) has happened. This phenomenon is characterized by a strong temperature increase in the stratosphere, possibly up to 50 °C within a few days. Being able to predict these events can help meteorologists improve weather forecasting. In this work, we aim to classify and predict SSWs. To the best of our knowledge, this is the first successful attempt for the prediction of this phenomenon using Machine Learning approaches.

1 Motivation and problem statement

Sudden Stratospheric Warmings (SSWs) are events which involve a sudden temperature increase in the mid to upper stratosphere, combined with a reversal of the climatological westerly zonal-mean zonal winds associated with the stratospheric polar night jet [6].

Except for their effects on the stratosphere, SSWs are also important since the associated temperature and wind anomalies can descend downward into the troposphere on time scales of weeks to months. This can lead to extreme cold air outbreaks in parts of North America, northern Eurasia, and Siberia and strong warmings in Greenland, eastern Canada, and southern Eurasia [7].

What is considered an SSW event or not, is still an active area of research. Consequently, different definitions based on various criteria exist. For example, the CP07 and U65 definitions annotate the event by looking at the reversal of the wind at a specific latitude and pressure level. In contrast, the ZPOL and EOFZ definitions annotate these events by tracking geopotential anomalies at the stratosphere.

The ability to characterize SSWs has many advantages, as it will help meteorologists improve weather forecasting and will also give a more concrete definition of the phenomenon. Therefore, in this project we used Machine Learning (ML) methods to classify and predict SSWs using various definitions. By following a systematic approach, we achieved almost perfect classification accuracy for past events and decent results for the prediction task.

2 Related work

Despite their importance to climate research and meteorology, only few researchers have tried to characterize SSWs using ML algorithms. [5] applies supervised learning in order to classify SSWs.

Data Science Lab (2018), Zurich, Switzerland.

Their approach consists of two parts: First of all, they do feature engineering and consequently they use the calculated features as input to various classifiers. Their objective is a daily multi-class classification. Some of the features they used are an indicator of whether the first principal component of the averaged temperature exceeds some threshold, the long term mean of the temperature at 30 hPa and effects generated by other phenomena such as the quasi-biennial oscillation and the solar cycle with a temporal lag. Their objective consists of four classes:

1. Undisturbed: no SSW has occurred.
2. Final: the warmings observed are not results of an SSW.
3. Major: an SSW that caused a wind reversal has happened.
4. Minor: an SSW that did not cause a wind reversal has happened.

The classifiers used are Linear Discriminant Analysis (LDA), Soft Margin Support Vector Machines (SM-SVMs) and Multilayer Perceptrons (MLPs) with 2 hidden layers.

Apart from supervised learning, unsupervised learning has also been applied to investigate if there is a natural way to distinguish between SSWs. [9] use various polar stratospheric measurements, including temperature, zonal winds and geopotential at different latitudes and pressure levels and try to apply clustering algorithms to find out if there is a natural separation between data points that contain an SSW and data points that do not. In total, 154 measurements are clustered using the k-means algorithm. Many of these measurements are highly correlated with one another. The authors use silhouette values, which is a measure of the within-group and between-group differences, in order to determine the optimal number of clusters. Their experiments show that by using $k = 2$ clusters they can separate the measurements that contain an SSW from the ones that did not almost perfectly and with a high separation margin.

Another interesting study is conducted by [14], where the CAVE-ART algorithm is proposed to detect split-like events in the polar vortex and afterwards associate them with potentially major SSWs. The algorithms consists of using scaled polar vorticity (sPV) values and segmentation algorithms to determine intrinsic diagnostics of each vortex object. After that, the predictor-corrector algorithm [15] is used to track the vortex objects through time.

3 Experimental setup

Our work consists of two tasks:

1. Classification: Given time series data for the entire winter, determine whether an SSW happened or not (binary classification).
2. Prediction: Given time series data for some days in a winter, predict whether an SSW will happen in one of the next weeks.

Since it is unfeasible and illogical to train machine learning models with only real data (where the sample size is limited to just 58 winters, as we do not have observations before that), a synthetic dataset consisting of 2959 winters is generated using the model referred in [16] with different initial conditions for every 100 generated years. Using the simulated winters, we are able to train our ML models and derive some conclusions about the real data by using them as a small test set.

Given advice from our domain experts, both for prediction and classification, three variables are used: the zonal velocity of the wind (also known as U component) at 60 °N and 65 °N latitude at a pressure of 10hPa, combined with cosine-averaged temperature between 60 °N and 90 °N latitude at a pressure of 10hPa. The choice of the variables is also done to be in accordance with the definitions used and implemented for both of our tasks. Deep Learning (DL) and ML approaches are being used to tackle the tasks.

The main ML algorithms used are bagging and boosting techniques, such as Random Forest (RF) and XGBoost [8]. The input to these algorithms are handcrafted features such as histograms of the time series values or the fourier transform of the time series. Most of the handcrafted features are derived by using the python package tsfresh [4]. We also use autoencoders (simple and denoising ones) in order to investigate if there is a more natural encoding of our input data. The DL approach used is Convolutional Neural Networks (CNNs). All of these techniques will be covered in more detail in the next sections. A visualization of the pipeline can be observed in Figure [1]. For all our experiments we set a constant random seed for reproducibility.

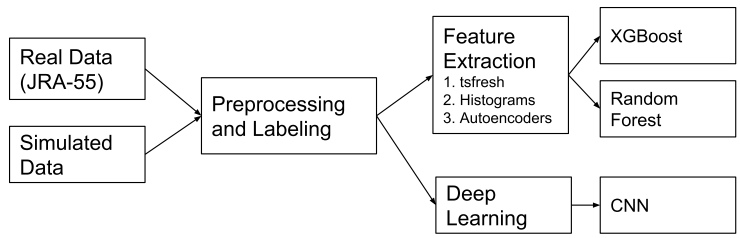

Figure 1: An overview of the pipeline for the classification and prediction task

3.1 Preprocessing of the simulated data

The simulated data are a series of NetCDF4 files. NetCDF [2] is a popular format for storing climate simulation data because of its object and array structure. Each file represents a calendar year and required heavy preprocessing to get the three variables needed as our input. At this point, we have to note that the winter period that is referred in this report is actually from the beginning of October to the end of April, because this is the time period where we can observe and characterize SSWs. We also have to note that in the simulated data all months contain exactly 30 days. Taking these information into account, our preprocessing steps can be summarized as follows:

1. We create one winter period by concatenating the corresponding periods from two calendar years.
2. We get the variables of interest at all longitudes and latitudes and average them though all longitudes.
3. For the wind variables, we interpolate the closest values that exist for the pressure of 10 hPA and then we interpolate again for the latitudes of interest ($60°$ and $65°$). We end up having two time series of 210 days with float values.
4. For the temperature variable we first interpolate for all the latitudes in order to get values starting from $60°$ and ending to $90°$ with a step of $10°$. Then we interpolate though the pressure levels to get the pressure level of 10 hPA. After we have interpolate the values for all the levels of interest (both pressure and latitude), we calculate a weighted-cosine average as our final value. We end up with a time series of 210 days of float values. For all the procedures above we use linear interpolation.
5. Our final input tensor has dimensions $2959 \times 3 \times 210$ (2959 data points with 3 measurements over the period of 210 days).

3.2 Preprocessing of the real data

Our real data come from the open dataset JRA-55 [13], which covers 55 years starting from 1958. The preprocessing procedure is very similar to the one used for the simulated data. The only difference is that in this dataset, every month is not 30 days, but instead has its real value. To alleviate for this issue, we remove as many days from April as it is necessary, so as to have time series with length of 210. This choice was made because no SSW events are recorded in April, but the inclusion of this month in our input data is necessary for characterizing events that have happened in the end of March.

3.3 Labeling of the data

Because of the lack of a uniform definition for what an SSW is, after guidance from our domain experts we implement the 3 most prominent definitions in order to label our data and train our ML models in a supervised manner. Table [1] includes a list of the definitions which are used for the labeling procedure [7]. The labeling is conducted on a daily basis, as for each definition we can specify the exact date that the SSW has happened. In Figures [2, 3, 4] we can observe the distribution of SSW events for the various definitions for the months of October to April. We can notice that the distribution of the simulated data resembles a normal one. As there are significantly fewer data points in the real data, the distributions do not align perfectly. However, we cross-checked the correctness of our labeling procedure for the real data with the events that have been recorded by climate scientists, and we notice that our labels match the real ones perfectly. Finally, every definition detects a different number of SSW events throughout the years as the criteria used are not the same.

Table 1: SSW definitions used to label simulated and real data

Definition	Description	SSW per Year
Zonal Wind Reversal at 60° N (CP07)	Events occur when the zonal-mean zonal winds at 10 hPa and 60° N fall below 0 ms^{-1} from Nov to Mar. Events must return to westerly (> 0 ms^{-1}) for at least 20 consecutive days between events. The winds must return to westerly for at least 10 consecutive days prior to 30 Apr (or an event is considered a final warming).	0.65
Zonal Wind Reversal at 65° N (U65)	Identical to CP07, except using zonal-mean zonal wind at 65° N.	0.84
Zonal Wind and Temperature Gradient Reversal (U&T)	Events occur when the zonal-mean zonal winds at 10 hPa and 60° N fall below 0 ms^{-1} from Nov to Mar. Events that do not also have a meridional temperature gradient reversal (defined as the zonal-mean temperatures averaged from 80° to 90° N minus the temperatures averaged from 60° to 70° N) within $\sim$ 10 days of the circulation reversal are excluded. Events must return to westerly (> 0 ms^{-1}) for at least 20 consecutive days between events. The winds must return to westerly for at least 10 consecutive days prior to 30 Apr (or an event is considered a final warming).	0.65

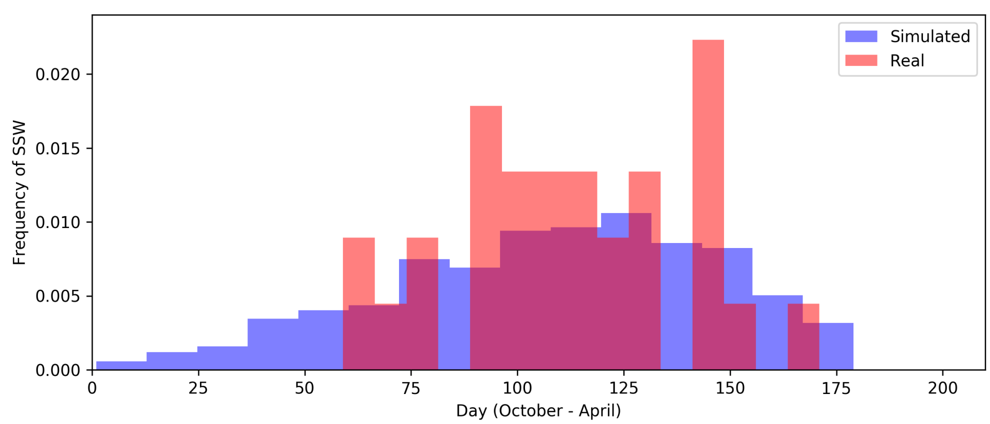

Figure 2: Distribution of SSWs for real and simulated data for the CP07 definition

3.4 Feature extraction

For the Random Forest (RF) and XGBoost models, we have to perform some feature engineering on the preprocessed data. For the RF model this procedure is composed of splitting the three variables of interest and computing histograms consisting of 20 bins on each one of them. After the 3 histograms are calculated, they are stacked horizontally in order to create a data point with dimension 60. For

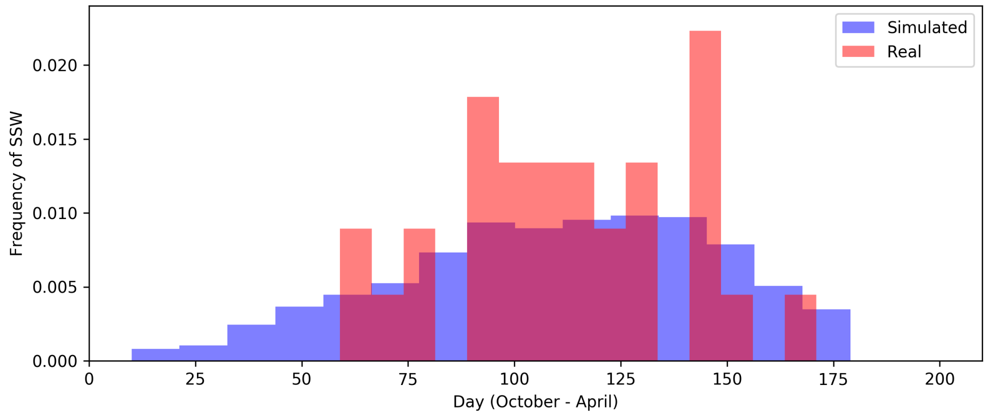

Figure 3: Distribution of SSWs for real and simulated data for the U&T definition

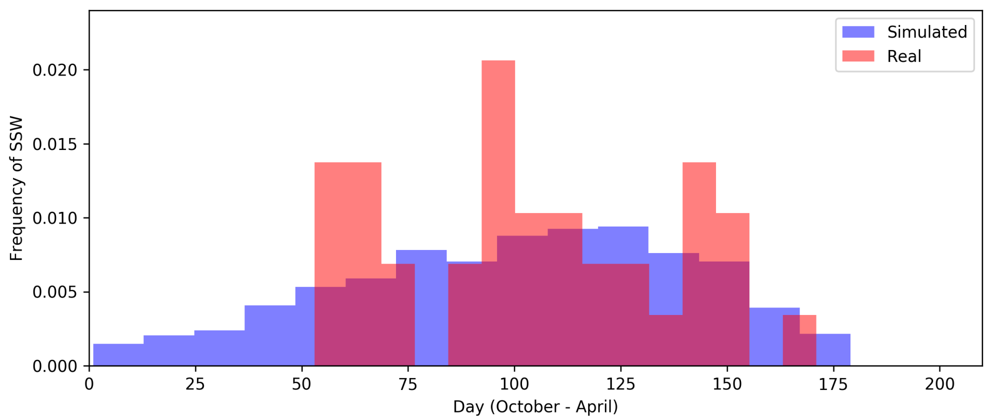

Figure 4: Distribution of SSWs for real and simulated data for the U65 definition

the XGBoost model, the tsfresh python package module is used. It calculates almost 800 time series features, such as the Fast Fourier Transform (FFT) and the mean of the time series. For a complete list of features, please go [here]. The same procedure as before for splitting and stacking the variables is used, leading to a dimensionality of around 2400 for each data point. For the prediction task, we also experiment with autoencoders for feature extraction. The autoencoders consist of two encoding and two decoding layers where each encoding layer has half the dimension of the previous layer and the decoding layers can be determined from the corresponding encoding layers. We also implement a denoising version, meaning that in the final encoding layer we randomly turn some of the units to zero. The activation functions used are ReLU and sigmoid. As before, we stack the three input variables horizontally to create one data point. Because of the different scales that the variables had, we experimented both with applying a scaling in the range of $[-1, 1]$ in each variable separately and with feeding the raw data into the autoencoder. For the implementation of the neural networks we used the PyTorch [3] package.

3.5 Evaluation

As mentioned before, the amount of real data available does not allow us to draw statistically sound conclusions about the effectiveness of our models. Therefore our evaluation procedure consisted of applying 5-fold Cross Validation into the simulated data. The evaluation metrics take into account if our data labels are balanced or unbalanced (e.g. Accuracy, Area Under the Receiver Operating

Characteristic Curve, F1 score). To get an idea of the performance for the real data, we trained our models using the whole simulation data and then used the real data as the test set.

4 Classification of SSWs

4.1 Models

In order to treat this task as a binary classification, we had to squash our 210 dimensional labels (one for each day of the winter period) for every year into a 1 dimensional point. The procedure to do that was rather straightforward. If there was an SSW anywhere in the winter period we categorized the whole winter into the positive class, whereas if there was no SSW throughout the whole winter we categorized it into the negative class. Consequently, we trained the following models:

1. An XGBoost Classifier with a 1000 estimators, 5 as the maximum tree depth for the base learners and L2 regularization term on weights equal to 0.1. The choice of the parameters was done after a GridSearch for the best score. The other parameters were left in their default values described [here]. The input to this classifier was the the tsfresh features that have been described before in the report.
2. A Random Forest Classifier with 10000 estimators. The other parameters were left in their default values described [here]. The input to this classifier was the the histogram features that have been described before in the report.
3. A CNN with max pooling (5816 parameters) with an exact architecture of a 1-D convolutional layer with 64 filters with multiple kernel sizes (one kernel has size 10 and the other one has size 15) followed by a max-over-time pooling layer and a fully connected layer with dropout with a probability of 0.4.
4. A CNN with two layers (65154 parameters) with an exact architecture of a 1-D convolutional layer with 32 filters with a kernel size of 15 followed by a 1-D convolutional layer with 64 filters with a kernel size of 20 and a fully connected layer with dropout with a probability of 0.3.

For the training of the CNNs , the ADAM [12] optimizer was used with a learning rate of $3 \cdot 10^{-4}$ and a batch size of 8. The models are trained for 100 epochs. Note that the CNNs were trained on the raw input data that were treated as 3 different channels leading to a dimension of 210×3 for each data point.

4.2 Results

Since the label distribution on the simulated data is almost equal in the positive and the negative class, accuracy is a suitable metric to evaluate our models. For the real data the label distribution is skewed towards the negative class. Our results using 3 metrics for the simulated data are presented in Figures [5, 6, 7] and for the real data in Table [2]. For both simulated and real data we use 3 metrics for the evaluation (Accuracy, AUROC and F1 Score). The results of the simulated data are presented as boxplots and the results on the real data are being rounded to two decimal places. We can make the following observations:

- For the simulated data XGBoost performs almost perfectly for all the available metrics and has very good results on the real data, taking into account that they are only 58 points, meaning that each misclassified point results in a drop of about 2% in the scores.
- Random Forest has the largest variance of every estimator, probably because the histogram features are very different between data folds and real and simulated data. Also its performance on the real data is very low.
- 1D-CNN with max pooling achieves very good results on the simulated data, but its performance on the real data is low probably because of the difference in the distribution.
- The simple 1D-CNN may not have the best results on the simulated data and also the second largest variance but it has comparable results with XGBoost on the real data.
- Taking the previous points into account, we can say that good performance on the simulated data does not necessarily imply good performance on the real data.

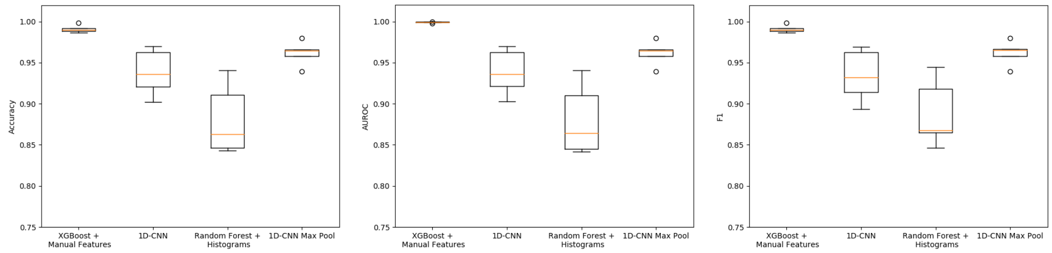

Figure 5: Classification results for the CP07 definition

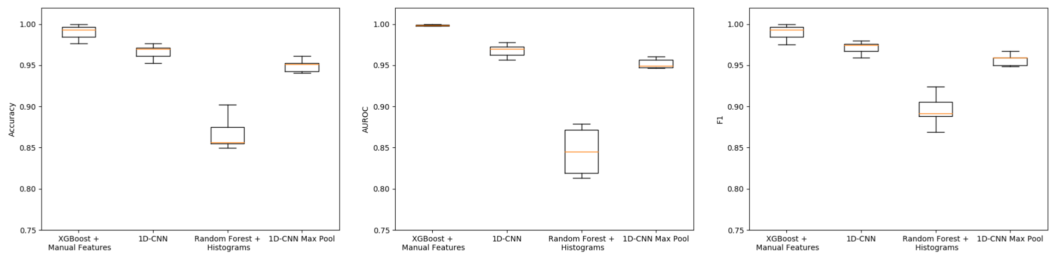

Figure 6: Classification results for the U65 definition

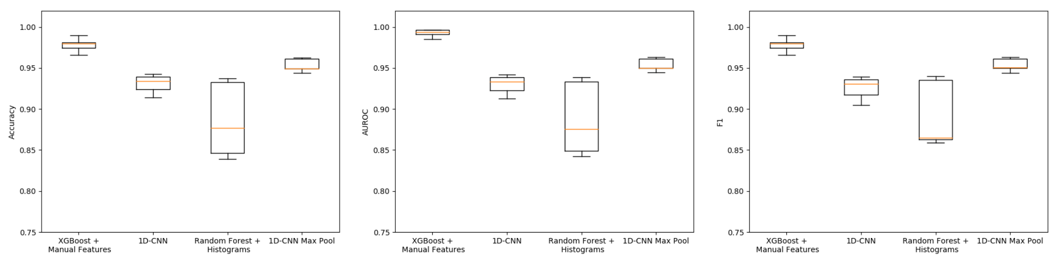

Figure 7: Classification results for the U&T definition

Because of the very good performance that XGBoost had on both the simulated and real data and the nature of the classifier we also investigated which are the three most important features for the classification for all three definitions. Based on the feature importances results, we can make the following comments:

- For all the three definitions, the classifier accurately grasps the introduced bias, as its most important features are the ones used to characterize an SSW.
- The only thing that is not taken into account in the U&T definition is the meridional temperature gradient reversal, but this is not in the initial variables that have been used for the calculation of the features.

5 Prediction of SSWs

5.1 Introduction

For the prediction task, we had to make some modifications compared to the classification task. After consultation from our domain experts, we focused on predicting SSWs for the months of January and February, where the concentration of the phenomenon is higher. Therefore, we cut our time series for the three variables at the end of December and January respectively and we created labels for the full month ahead of the cutting point for each week independently. We labeled as the positive class if an SSW has happened in the specific week and as the negative class otherwise. We also experimented on how far to look into the past, meaning before the cutoff point. We aggregate the time series starting from one week and up until a month before the cutoff point. Our final grid of experiments consists of a supervised binary classification task for each of the weeks. The size of the grid is 4 prediction

Definition	Classifier	Metric	Score
CP07	Random Forest with histograms	Accuracy	0.59
		F1	0.71
		AUROC	0.57
	1D-CNN	Accuracy	0.84
		F1	0.86
		AUROC	0.84
	1D-CNN with max pooling	Accuracy	0.48
		F1	0.61
		AUROC	0.47
	XGBoost with tsfresh features	Accuracy	0.88
		F1	0.88
		AUROC	0.88
U65	Random Forest with histograms	Accuracy	0.69
		F1	0.80
		AUROC	0.57
	1D-CNN	Accuracy	0.91
		F1	0.90
		AUROC	0.86
	1D-CNN with max pooling	Accuracy	0.66
		F1	0.75
		AUROC	0.59
	XGBoost with tsfresh features	Accuracy	0.88
		F1	0.88
		AUROC	0.83
U&T	Random Forest with histograms	Accuracy	0.59
		F1	0.71
		AUROC	0.57
	1D-CNN	Accuracy	0.84
		F1	0.87
		AUROC	0.84
	1D-CNN with max pooling	Accuracy	0.5
		F1	0.59
		AUROC	0.49
	XGBoost with tsfresh features	Accuracy	0.83
		F1	0.82
		AUROC	0.82

Table 2: Results of all the classifiers for all the definitions for the real data

weeks $\times$ 3 definitions $\times$ 2 cutoff points $\times$ 4 feature intervals $\times$ 2 data types (real and simulated) and is equal to 192. We are doing the evaluation on the simulated and real data with the same procedure as the one described in the classification task. For the simulated data we only report the mean of our scores. The variance in each of the runs is very small, with a maximum value of 0.04

5.2 Models

The same models that have been described in the classification task have been trained and evaluated on the prediction task. The only additional model was the XGBoost classifier trained with autoencoders features instead of tsfresh features. The autoencoders' structure has been described in the preprocessing section. The optimizer used was Stochastic Gradient Descent (SGD) with a learning rate of 0.01. As the dataset was highly unbalanced towards the negative class, we oversampled our training set. The test set was not oversampled as this would lead to false results. For the oversampling we used the ADASYN [10] algorithm with implementation provided by the imbalanced-learn [1] python package. The default parameters were used. We also evaluated using the F1 and AUROC scores as accuracy is an unsuitable metric for imbalanced datasets.

Definition	Feature
CP07	Location of the first minimum of the wind at 60° latitude
	Number of x-axis crossings of the wind at 60° latitude
	Minimum of the wind at 60° latitude
U65	Number of x-axis crossings of the wind at 65° latitude
	Location of the first minimum of the wind at 65°
	Minimum of the wind at 65° latitude
U&T	Location of the first minimum of the wind at 60° latitude
	Number of x-axis crossings of the wind at 60° latitude
	Minimum of the wind at 60° latitude

Table 3: Feature importances for the XGBoost Classifier for all three definitions

5.3 Results

Unfortunately, besides the XGBoost classifier with tsfresh features, all the other classifiers and combinations that we tried, performed poorly. They either predicted a random class or only the dominant one besides having an oversampled training set. We present the results of the prediction task in Figures 8-19. We can make the following comments:

- Predicting an SSW for a period of more than a week is not feasible with our current methods and features. Indeed, as the phenomenon is sudden it would be very hard to achieve good results for a longer prediction interval.
- The scarcity of SSWs, as well as the size of the real data, does not allow us to draw useful conclusions as to how much a correctly classified data point will increase the corresponding score. However, we can see that classifiers that perform well on the simulated data tend to perform well on the real data as well.
- Our best scores come for the period of February. We think that this is due to to the higher frequency of SSWs compared to the period in January.
- Looking further into the past more does not help with the prediction, but on the contrary decreases the score of our classifier. This may be because some features are aggregations for longer time periods. Running the classifiers for separate features calculated on each week would yield more insight. Unfortunately, we did not have time for this.
- The most important feature for the U&T and CP07 definitions were the minimum of the wind at 60° latitude, whereas for the U65 was the minimum of the wind at 65° latitude. Some other important features were the angles of the Fast Fourier Transform coefficients.

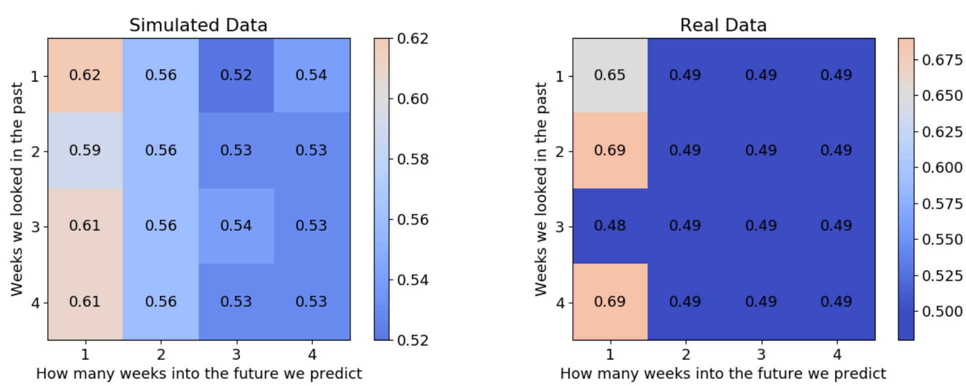

Figure 8: F1 scores for the CP07 definition for predicting if an SSW will happen in January

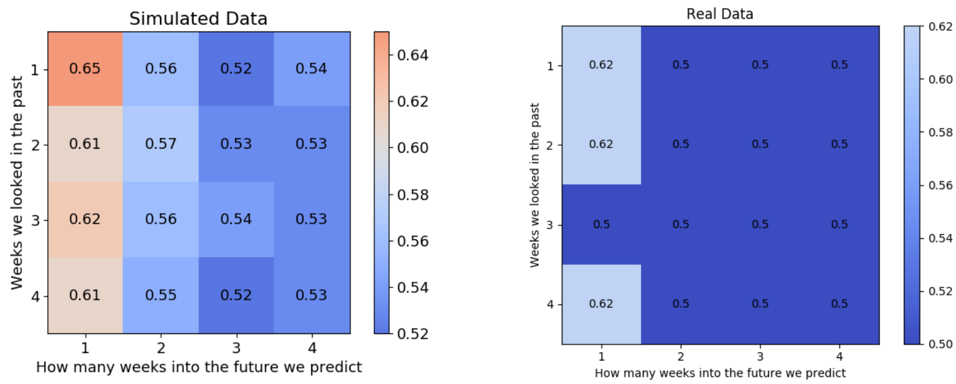

Figure 9: AUROC scores for the CP07 definition for predicting if an SSW will happen in January

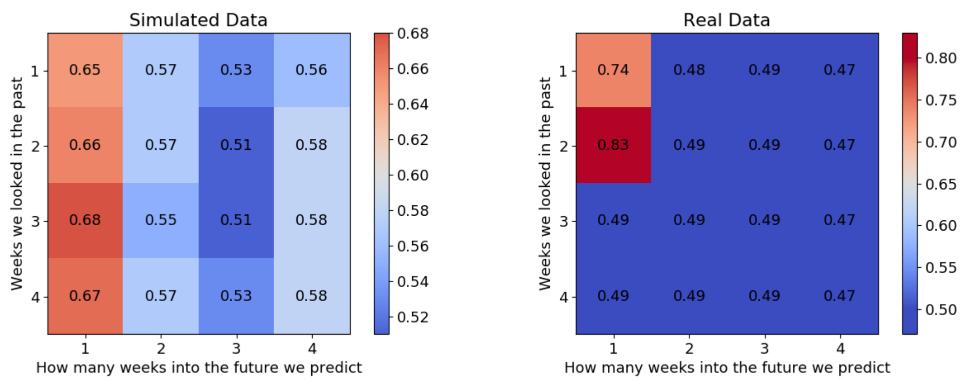

Figure 10: F1 scores for the CP07 definition for predicting if an SSW will happen in February

6 Further work and conclusion

In this work, we got involved with the classification and the prediction of SSWs using three definitions. After trying various approaches in both feature engineering and the use of ML models, we achieved almost perfect classification results and decent prediction results for the horizon of one week. It is important to note that even deterministic dynamical systems cannot predict an SSW for a horizon of more than a few days [11]. It would be interesting to investigate if the calculated time series features over smaller periods of time would still decrease the score of the XGBoost classifier when we look more into the past for the prediction task. Another interesting direction for the prediction task would be to experiment with encoder-decoder Long Short Term Memory (LSTM) architectures. We hope that our approach will make a step towards a better prediction of SSWs over a longer period in the future and support meteorologists in weather forecasting. All our code and instructions on how to reproduce our experiments are [here] (please note that you have to request access from the [DS3Lab] at ETHZ in order to view the code).

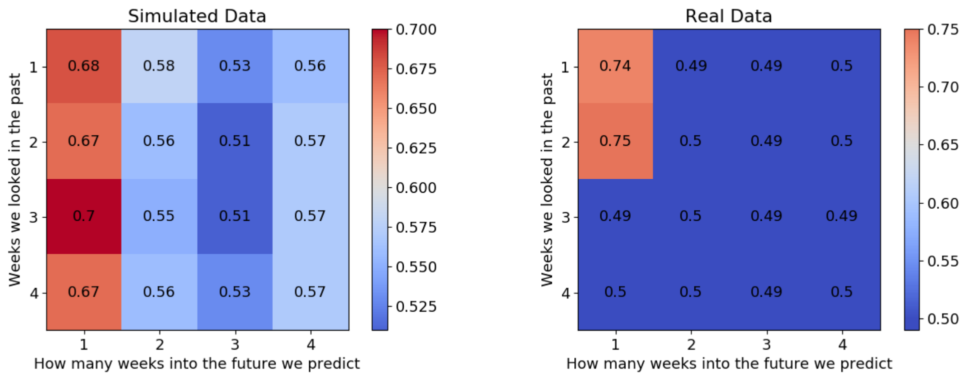

Figure 11: AUROC scores for the CP07 definition for predicting if an SSW will happen in February

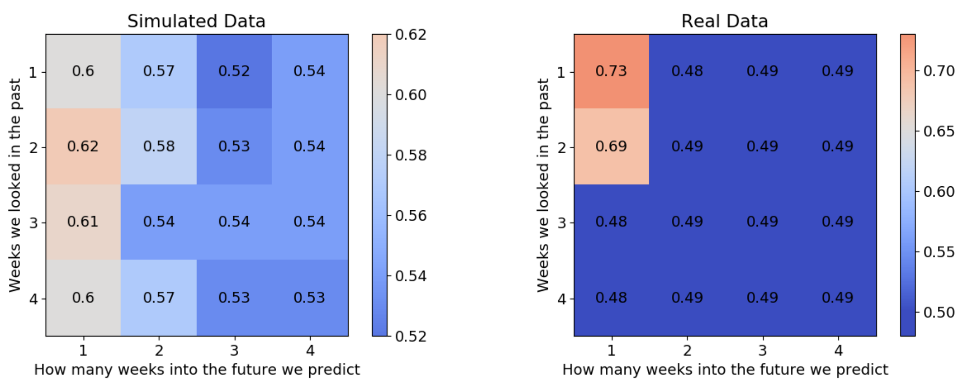

Figure 12: F1 scores for the U&T definition for predicting if an SSW will happen in January

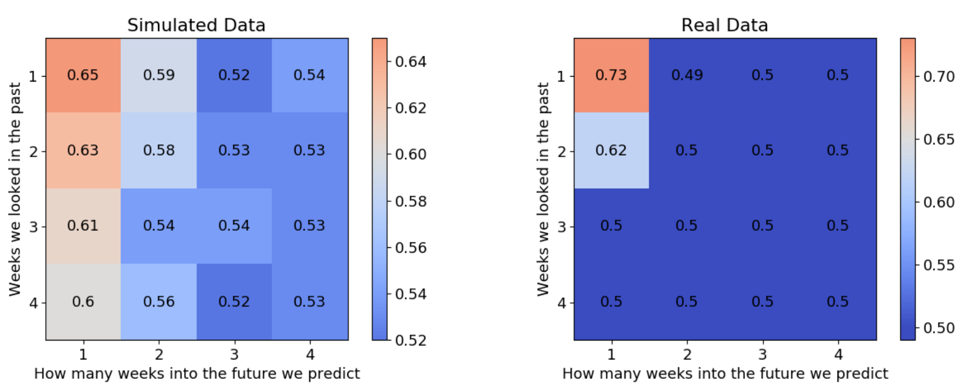

Figure 13: AUROC scores for the U&T definition for predicting if an SSW will happen in January

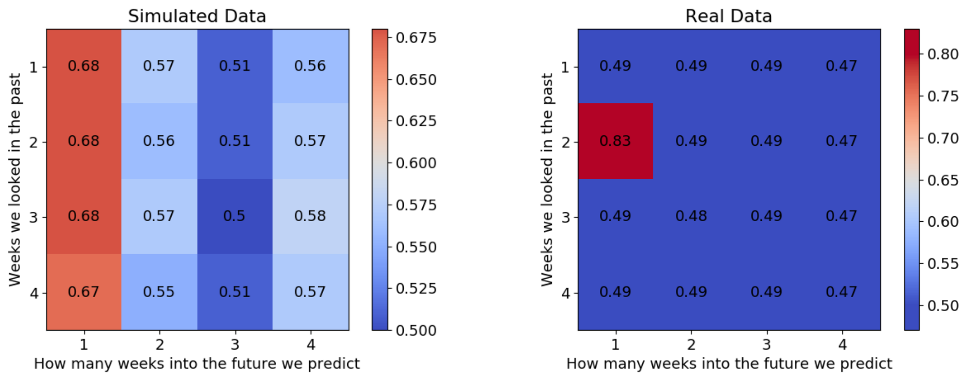

Figure 14: F1 scores for the U&T definition for predicting if an SSW will happen in February

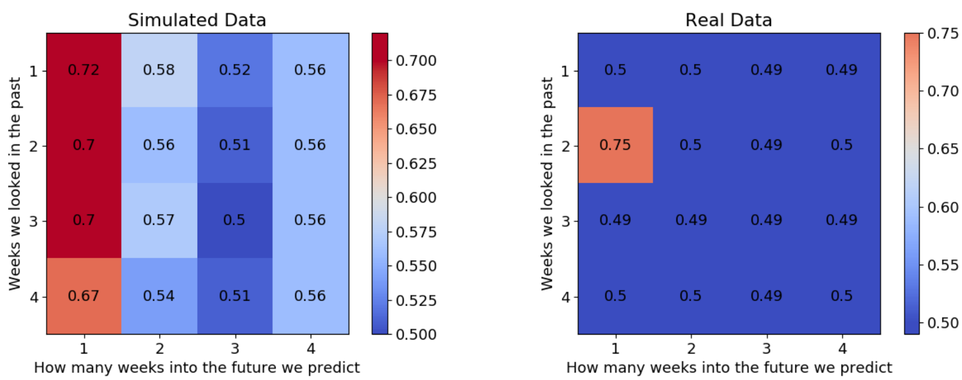

Figure 15: AUROC scores for the U&T definition for predicting if an SSW will happen in February

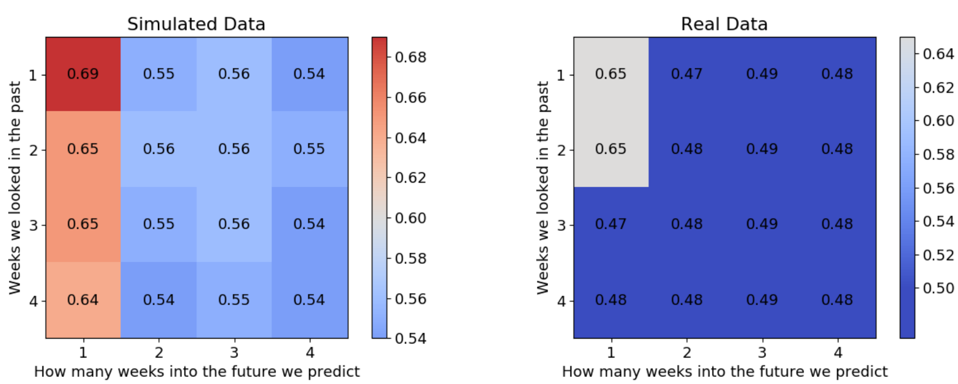

Figure 16: F1 scores for the U65 definition for predicting if an SSW will happen in January

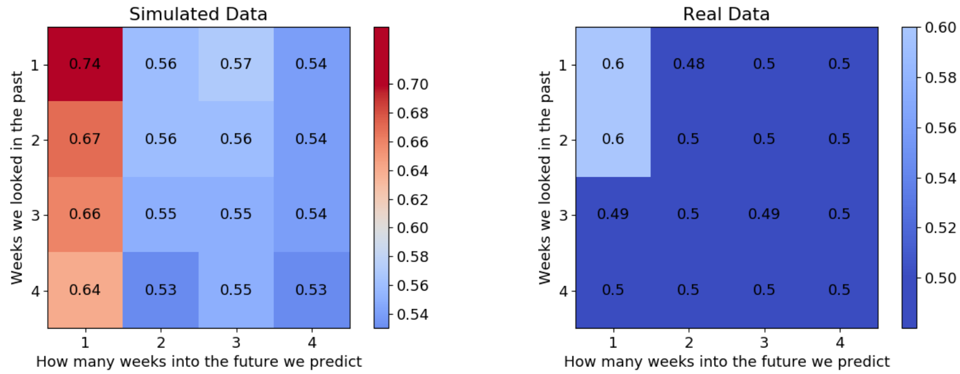

Figure 17: AUROC scores for the U65 definition for predicting if an SSW will happen in January

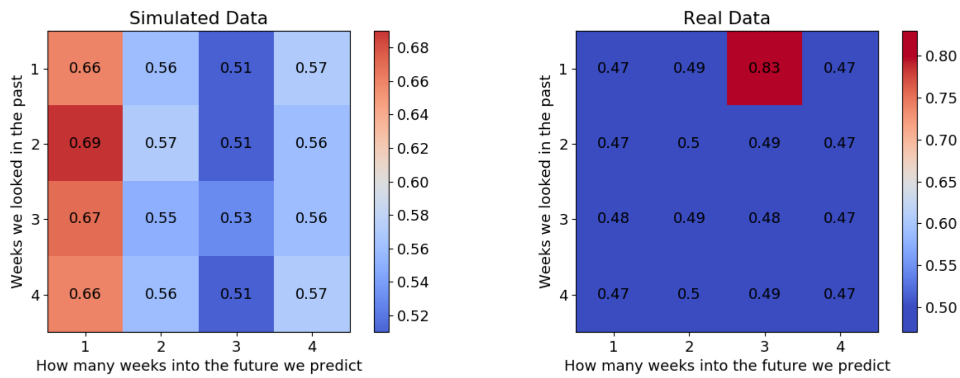

Figure 18: F1 scores for the U65 definition for predicting if an SSW will happen in February

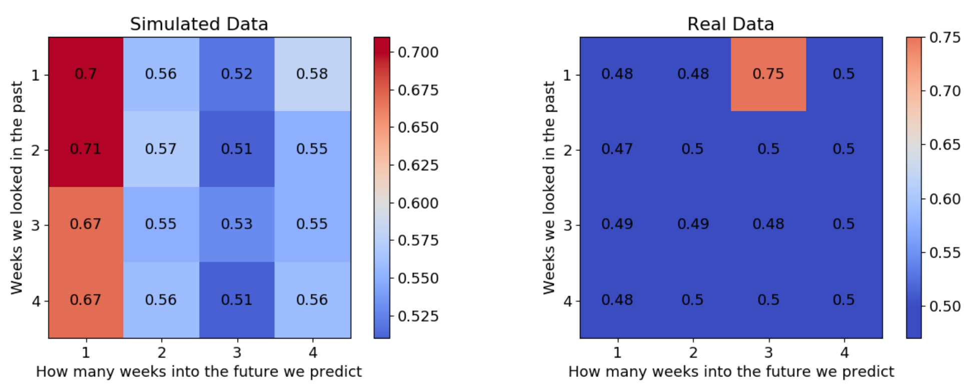

Figure 19: AUROC scores for the U65 definition for predicting if an SSW will happen in February

References

[1] imblearn. `https://imbalanced-learn.org/en/stable/`. Accessed: 2018-12-17.

[2] Netcdf. `https://www.unidata.ucar.edu/software/netcdf/docs/netcdf_introduction.html`. Accessed: 2018-12-15.

[3] pytorch. `https://pytorch.org/`. Accessed: 2018-12-16.

[4] tsfresh. `https://tsfresh.readthedocs.io/`. Accessed: 2018-12-15.

[5] Christian Blume, Katja Matthes, and Illia Horenko. Supervised learning approaches to classify sudden stratospheric warming events. *Journal of the Atmospheric Sciences*, 69(6):1824–1840, 2012.

[6] Amy H Butler and Edwin P Gerber. Optimizing the definition of a sudden stratospheric warming. *Journal of Climate*, 31(6):2337–2344, 2018.

[7] Amy H Butler, Dian J Seidel, Steven C Hardiman, Neal Butchart, Thomas Birner, and Aaron Match. Defining sudden stratospheric warmings. *Bulletin of the American Meteorological Society*, 96(11):1913–1928, 2015.

[8] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[9] K Coughlin and Lesley Janet Gray. A continuum of sudden stratospheric warmings. *Journal of the Atmospheric Sciences*, 66(2):531–540, 2009.

[10] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1322–1328. IEEE, 2008.

[11] Alexey Yu Karpechko. Predictability of sudden stratospheric warmings in the ecmwf extended-range forecast system. *Monthly Weather Review*, 146(4):1063–1075, 2018.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Shinya Kobayashi, Yukinari Ota, Yayoi Harada, Ayataka Ebita, Masami Moriya, Hirokatsu Onoda, Kazutoshi Onogi, Hirotaka Kamahori, Chiaki Kobayashi, Hirokazu Endo, et al. The jra-55 reanalysis: General specifications and basic characteristics. *Journal of the Meteorological Society of Japan. Ser. II*, 93(1):5–48, 2015.

[14] Zachary D Lawrence and Gloria L Manney. Characterizing stratospheric polar vortex variability with computer vision techniques. *Journal of Geophysical Research: Atmospheres*, 123(3):1510–1535, 2018.

[15] Chris Muelder and Kwan-Liu Ma. Interactive feature extraction and tracking by utilizing region coherency. In *Visualization Symposium, 2009. PacificVis' 09. IEEE Pacific*, pages 17–24. IEEE, 2009.

[16] Geoffrey K Vallis, Greg Colyer, Ruth Geen, Edwin Gerber, Martin Jucker, Penelope Maher, Alexander Paterson, Marianne Pietschnig, James Penn, and Stephen I Thomson. Isca, v1. 0: a framework for the global modelling of the atmospheres of earth and other planets at varying levels of complexity. 2018.