



Seasonal average temperature forecast with the AutoGluonTS modern autoML tool

Diego Kiedanski, Pablo Rodriguez-Bocca, Gerardo Rubino

► To cite this version:

Diego Kiedanski, Pablo Rodriguez-Bocca, Gerardo Rubino. Seasonal average temperature forecast with the AutoGluonTS modern autoML tool. MACHine Learning for EArth ObservatioN, European Conference on Machine Learning, Sep 2023, Torino, Italy. pp.1-10. hal-04384070

HAL Id: hal-04384070

<https://hal.science/hal-04384070v1>

Submitted on 10 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Seasonal average temperature forecast with the AutoGluonTS modern autoML tool [★]

Diego Kiedanski¹, Pablo Rodríguez-Bocca¹, and Gerardo Rubino²

¹ Facultad de Ingeniería, Univ. de la República, Montevideo, Uruguay
`{diego.kiedanski, prbocca}@fing.edu.uy`

² INRIA, Rennes, France
`gerardo.rubino@inria.fr`

Abstract. Forecasting the temperature at a given place in a given future epoch is of evident huge interest. This is extremely hard unless future means “next days”. In this paper, we focus on the problem of forecasting if a given future period (say, next summer) will be hotter than usually. More precisely, the objective is to say if the average daily temperature will be above normal, normal or below normal, where normal is defined on past observed values. For this task, most of the production tools are of the dynamic type (that is, physical, usually huge systems of partial differential equations) or statistical, or a combination of both. In the paper, we explore the use of modern Machine Learning tools and show that they can compete with those dedicated and always heavy and complex procedures, in spite of the fact that our results come from standard tools and hardware. We illustrate the results with data coming from a project where these questions are studied, concerning a large area in the Southern part of South America.

Keywords: Seasonal Average Temperature · Forecasting · AutoGluon-Time Series.

1 Introduction

In this paper, we focus on a particularly difficult time-series analysis problem: the forecasting of seasonal average temperatures in the following 90 days. The problem is very hard to solve. For this reason, the climatologists usually avoid forecasting the average temperature directly. Instead, they forecast a three-level classification of this average temperature: *above normal*, *normal*, and *below normal*, where normality is defined as typical conditions for the current season (see below).

To solve this problem, several dedicated models have been developed. Following [1], we summarize below the most important ones for our study. In the paper, we apply instead a set of modern Machine Learning (ML) techniques to the problem, in order to evaluate their performances. Machine Learning (ML)

[★] This work was supported by the “ClimateDL” project belonging to the Climat Am-Sud program, with program code 22-CLIMAT-02.

has made significant progress over the past decade. However, as they get more sophisticated, it is increasingly difficult, even for experts, to incorporate all the recent best practices into their modeling activities. In that context, automated Machine Learning (autoML) frameworks offer a powerful alternative, since they take care alone of the whole configuration process. Numerous autoML frameworks have recently emerged to solve time-series problems [2]. In this paper, we will use the autoML framework called AutoGluon-Time Series (AutoGluonTS) [3]. It contains several tools, ranging from off-the shelf boosted trees to customized neural networks. These models are assembled into a final one, that a priori should outperform any of them taken individually.

The rest of this paper is organized as follows. This section finalizes with the description of main related work. Section 2 presents the dataset and the methodology developed to create our model. Our experimental setup and results are presented in Section 3. Finally, Section 4 provides some concluding remarks.

Related work. We briefly describe here the tools that we selected as relevant for this analysis, and then the metrics used for comparing the different techniques.

A main reference for seasonal weather forecasting is the NMME (North American Multi-Model Ensemble) tool ³, a large platform combining several also large models, among the main ones in North America. They are all based on numerical simulation methods, also called dynamical models, or Numerical Weather Prediction (NWP) models. On the statistical side, we have the CPT (Climat Predictability Tool) model using many variables (wind, temperature, precipitations, etc.). It comes from Columbia University [4]. The SMN-CPT.NMME variant combines previous tool with NWP models used inside NMME. We also have CLIMAX [5], based on the outputs of most of the modules composing NMME, combined using regressions to return a single forecast. Let us also mention the IRI (International Research Institute for Climate and Society), an important research center at Columbia University [6]. It produces climate well-known prognostics of different kinds, one of them by combining the outputs of the components of NMME. There is also a different way of combining the outputs of several models, using a “consensus” in a set of human experts. It is used by some consumers of forecasts in the area [1]. The experts analyze those outputs, use their experience and skills, and produce a new forecast, often better than the individual original ones.

For the seasonal average temperature three-level classification forecasting problem, our main reference [1] uses two metrics to evaluate the performance of the models [7]: (1) HIT SCORE (for deterministic forecasts), the number of hits divided by the number of forecasts issued (a hit being a correct class classification), and (2) the well-known AUC, the area under the Relative Operating Characteristics (ROC) curve. While the Hit score is the suggested metric to use when you have a single forecast map for a quarter or season, the AUC is suggested for a series of forecast maps, as in our case. We will present our results using the AUC metric (see [7] for a deep explanation of these and other met-

³ <https://www.cpc.ncep.noaa.gov/products/NMME/>

rics). Table 1 shows the AUC metric results for our problem using the models described above [1] for the period January 2018 .. June 2021.

Table 1: AUC for the three classes for the mean seasonal temperature forecasted by the models presented above. From January 2018 until June 2021. Source: [1].

Model	<i>above normal</i>	<i>normal</i>	<i>below normal</i>
Consensus	0.59	0.56	0.53
SMN-CPT	0.53	0.61	0.51
SMN-CPT.NMME	0.55	0.62	0.53
CLIMAX	0.56	0.55	0.50
IRI	0.52	0.50	0.49
NMME	0.57	0.54	0.50

2 Materials and Methods

Let us formalize our forecasting problem and define the metrics used. Later, we will present our dataset and the feature engineering developed. Finally, we will describe the AutoGluonTS tool and our experimental methodology.

In our seasonal average temperature classification problem, we deal with the mean temperature in the following 90 days, and we forecast the fact that it is *above normal*, *normal*, or *below normal*. To decide what is “normal” at a specific date, we basically look at the same calendar day (month–day, that is, Jan 1, Jan 2, . . . , Dec 31) in the 30 years-length fixed interval 1981..2010, and compute the terciles of that set of values. Then, normal temperature means being between them. Normality is thus defined as the “typical” temperature for current season observed during the considered reference period 1981..2010.

More precisely, we have a set $\mathcal{P}$ of N geographical points where we want to forecast the average temperature class in the next 90 days. These points are weather stations, where we have historical data in discrete time steps, $t \in \mathbb{N}$. In our case, as we will show later in Subsection 2.1, we have the daily measure of the maximum temperature $\mathbf{y}_{\max,t}$ and minimal temperature $\mathbf{y}_{\min,t}$, in Celsius (plus the accumulated precipitations in mm), for which we will use respectively, the notation $\mathbf{y}_{\max,t}$, $\mathbf{y}_{\min,t}$, for the day $t \in \mathbb{N}$. Fix then any of the stations in $\mathcal{P}$, and consider the problem of forecasting the class of the mean temperature in the next 90 days. We estimate the daily average temperature just as the arithmetic mean $\bar{\mathbf{y}}_t = (\mathbf{y}_{\min,t} + \mathbf{y}_{\max,t})/2$, for all $t \in \mathbb{N}$. The seasonal average temperature in the last 90 days (past 90 days until now) is defined as follows: for all epoch t , $\bar{\mathbf{y}}_{90,t} = 90^{-1} \sum_{i=t-89}^t \bar{\mathbf{y}}_i$.

Next, we need to compute the quantiles of the data associated with every used epoch t . Instead of looking at the calendar day (month–day) of the reference 1981..2010 period only, we add to these 30 numbers the mean temperatures of 1 and 2 days before and after that (month–day), in order of better capturing the typical conditions on those years. We then compute the terciles of the resulting

set of $30 \cdot 5 = 150$ numbers, that we call here $\mathcal{D}_t$. Let us denote them as $\tau_{1,t}$ and $\tau_{2,t}$. So, epoch t is in class *below normal* if $\bar{y}_{90,t} < \tau_{1,t}$, in class *normal* if $\tau_{1,t} \leq \bar{y}_{90,t} < \tau_{2,t}$ and in class *above normal* if $\tau_{2,t} \leq \bar{y}_{90,t}$. This must be done for all geographical points and calendar days. Then, we define the class of any segment $\{t-89, t-88, \dots, t\}$, for all epoch t , as *above normal*, *normal*, *below normal* by means of the terciles $\tau_{1,t}$ and $\tau_{2,t}$. The vector of the classes of the segment $\{t-89, t-88, \dots, t\}$ in all the N geographical positions (the N stations) is denoted by $\mathbf{y}_t \in \{\text{above normal}, \text{normal}, \text{below normal}\}^N$.

Now, assume that you are in the training phase. We know the data until time t included. It consists of the vectors of the seasonal average temperature class $\mathbf{y}_t \in \{\text{above normal}, \text{normal}, \text{below normal}\}^N$, and other features, denoted by $\mathbf{X}_t \in \mathbb{R}^{N \times F}$, N geographical points with F features each (accumulated precipitations, atmospheric pressure, etc.). Therefore, we can define our predictive problem as the search for a function $f()$ that predicts the target variable S days ahead in the future, as a function of the historical data plus features values on the T last days, as follows:

$$\mathbf{y}_{(t-T):t}, \mathbf{X}_{(t-T):t} \xrightarrow{f} \mathbf{y}_{t+S},$$

where $\mathbf{y}_{(t-T):t} \in \mathbb{R}^{N \times T}$, $\mathbf{X}_{(t-T):t} \in \mathbb{R}^{N \times F \times T}$ and $\mathbf{y}_{t+S} \in \mathbb{R}^N$. In our case, $S = 90$ in order to predict the seasonal average temperature class in the immediately following 90 days. The size of the training data depends on the model.

We will use standard quality metrics to assess the quality of forecasts. In ML, the F1-score (F1) is commonly used to measure the performance of a binary classifier. A procedure to generalize it to multi-class problems is called One versus Rest multi-class strategy, and can be applied to F1 and other metrics. We can proceed in two ways: we can either take the mean of the performances (the F1s) of each class (this is called *macro-averaging*) or taking all the classes simultaneously (*micro-averaging*). In the same way, as in our main reference work [1], we can compute the Receiver Operating Characteristic (ROC) curve per class, and compute the Area Under the Curve (AUC). Here, we will use F1-macro, F1-micro, and AUC per class.

2.1 Southern South-America Dataset

The main dataset used in our project comes from 137 weather stations in Southern South America. It includes max and min daily temperatures and accumulated precipitations from 1977 to 2018. Data is provided by national meteorological services from Argentina, Uruguay, Chile, Brazil and Paraguay. The data came already pre-processed by Solange Suli and Verónica Dankiewicz at the Departamento de Ciencias de la Atmósfera y los Océanos, Facultad de Ciencias Exactas y Naturales, UBA⁴. Errors and outliers were fixed. Weather stations with high level of missing data were discarded (69 out of 206).

The dataset was split into two subsets: training and testing. The training includes the first 40 years, from 1977 to 2016; the year 2017 is used for testing

⁴ <https://exactas.uba.ar/institucional/concursos/auxiliares/departamento-de-ciencias-de-la-atmosfera-y-los-oceanos/>

and model evaluation. The year 2018 has a lot of missing data (19.4%), therefore it is discarded. Globally, the final dataset has 4.8% of missing temperatures with respect to the original data. This missing data appears differently depending on the geographical point and date.

As part of our dataset, we consider the following 34 features:

Main measured daily features (5): max, min and mean temperature, average of the means in the last 90 days, and accumulated precipitations.

Static weather station features (3): lon, lat, and altitude (in m).

Temporal features (2): day of the year transformed into two orthogonal sinusoidal real values (*sin* and *cos*).

Standardized Precipitation Indices (SPI, 5 features): the SPI [8] are used to characterize meteorological drought on a range of timescales. On short timescales, the SPI are closely related to soil moisture, while at longer timescales, they can be related to groundwater and reservoir storage. In our case, we compute several SPI indices: for 30, 90, 180, 270, and 360 days timescales, using daily precipitation input data.

Sea Surface Temperature (SST, 16 features). The Physical Sciences Laboratory (PSL) of the National Oceanic and Atmospheric Administration (NOAA) monthly publishes the sea surface temperature with a grid granularity of grades: 90×180 points (lat, lon). This dataset is known as “NOAA Extended Reconstructed SST V5”⁵. We use this dataset to compute a measure of anomalous month’ temperature, at each grid point, as the difference between the current temperature and the historical average one (from 1981 to 2010) at this grid point. Then, we separate the data by ocean basins: Pacific (long from 125 to 290°E, lat from -70 to 60°N), Atlantic (long from 290 to 340°E, lat from -70 to 70°N), and Indian (long from 20 to 125°E, lat from -60 to 20°N). With these three time-series of anomalies over a map grid, we compute the main spatial variability patterns using the Empirical Orthogonal Function (EOF) analysis [9]. EOF analysis is also known as weighted Principal Components Analysis (wPCA), and it is a decomposition of a dataset in terms of orthogonal basis functions to study spatial patterns very common in climate study fields. We compute the first 20 EOFs per time-series, and we keep as features the minimal number of EOFs to explain the 50% of variance in the signal. We need 7, 6 and 3 EOFs for the Pacific, Atlantic and Indian ocean regions respectively. In Figure 1 we show the first three EOFs for the Pacific Ocean and its associated time series. The time series are used as time dependent features. Therefore, in the Pacific Ocean case, we have 7 new features. These features are relevant because they summarize important climate predictors as ENSO (El Niño-Southern Oscillation) in the first EOF, and the PDO (Pacific Decadal Oscillation) in the second EOF. The same is performed with the Atlantic and Indian Oceans.

Sea level pressure at Geo-potential Height Gradient of 500 hPa (HGT500, 3 features). ERA5 reanalysis⁶, from the European Center for

⁵ <https://psl.noaa.gov/data/gridded/data.noaa.ersst.v5.html>

⁶ Copernicus Climate Change Service (C3S), 2017. ERA5: Fifth Generation of ECMWF Atmospheric Reanalyses of the Global Climate. Copernicus Climate

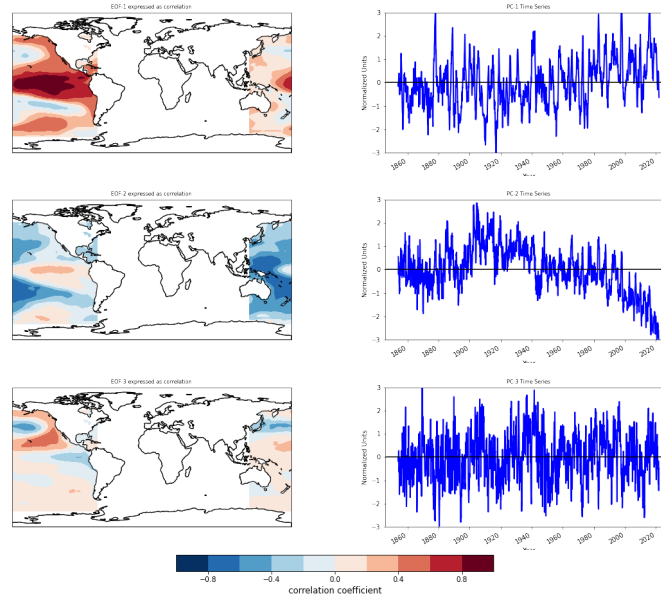


Fig. 1: First three EOFs for the Pacific Ocean and its associated time series.

Medium-Range Weather Forecasts (ECMWF), provides a highly accurate, hourly data, with a spatial resolution of $0.25^\circ \times 0.25^\circ$ and a vertical resolution of 37 pressure levels (1000 hPa to 1 hPa) from 1979 to present. From ERA5 we obtain the daily temperature at 500 hPa geo-potential height between the Atlantic Anticyclone and the Chaco Low (lon from 240 to 360°E, and lat from -70 to -10°S). With this dataset, we repeat the procedure explained for the Sea Surface Temperature (SST). We first compute the daily anomaly at each grid point, as the difference between the current and the historical average temperatures (from 1981 to 2010) at this grid point. With this time-series of anomalies over a map grid, we compute the main variability patterns using the Empirical Orthogonal Function (EOF). We compute the first 20 EOFs, and we keep as features the first three, which explains the 50% of variance in the signal.

2.2 Forecasting models in AutoGluonTS

AutoGluonTS uses several models on the same training data that share the same time segment. It automatically cleans data and select appropriate hyper-parameters, relevant models, etc. The models are of three types: local (they fit a single time-series, global (they can handle many time series simultaneously) and ensemble (to combine several forecasts into a single output).

Change Service Climate Data Store (CDS). (<https://cds.climate.copernicus.eu/>) (Accessed: 05 June 2023).

Local models use statistics only. They basically capture patterns (trends or seasonality). In this work, we used the following ones: the **Naïve** tool (it predicts the next value by copying the present one) and the **SeasonalNaïve** version, that uses the value observed last year, the same calendar day; the classical statistical **ARIMA** and variants (**SARIMA**, **SARIMAX**); the **ETS** and **AutoETS** models which compute weighted averages of previously observed values (as in ARIMA, etc.), but where the weights are exponentially decreasing with the time lags; another tool, **Theta**, uses a different decomposition of the series (the idea is to separate short and long term parts).

On the global models, we only used two tools (because of computation constraints). **AutoGluonTabular** [3] converts the problem into a tabular one with specific types of features: lags (based on the frequency of the training data), time features (for instance the calendar day), etc. Quantiles are computed assuming that the residuals are distributed according to Normal distributions with zero mean and estimated variance. The transformed problem is solved using an ensemble of XGBoost, CatBoost and LightGBM. The **DeepAR** tool [10] is a probabilistic forecasting model that uses a recurrent neural network to get probabilistic forecasts. It can handle numerous related time series. Finally, an ensemble model, **WeightedEnsemble**, works by combining the predictions of all other models.

2.3 Experimental Design

As mentioned, our dataset contains 34 features (defined in Section 2.1) for each day and for each geographical point, from 1977 to 2018. In order to avoid overfitting, AutoGluonTS automatically splits the training data into several folds, and apply traditional model-training vs. validation procedures. We use 2017 for testing, in order to stay close to our reference work [1]. Poggi et.al. evaluate expert predictor models from January 2018 to June 2021, three years and a half. This is not the only difference, we evaluate our work in 137 geographical points throughout the region, and the reference work only does so in 86 stations in Argentina. This is another reason that prevents a direct comparison with [1]. Moreover, the same authors periodically extend their work by increasing the testing period, and they observe relevant differences in the methods’ performance; therefore, a direct comparison with [1] is out of scope.

In addition to an advanced configuration, AutoGluonTS offers a simple configuration based in presets, where most hyper-parameters are chosen by the framework. In particular, we configure: 1) **preset=high_quality**: that includes local models (Naive, SeasonalNaive, ETS, AutoETS, Theta and ARIMA), global models (AutoGluonTabular and DeepAR), and optionally two deep learning models (TemporalFusionTransformer and SimpleFeedForward, not used in our case). This preset will enable hyper-parameter optimization for local statistical models; 2) **prediction_length=90**: all the fitted models forecast time series multiple steps into the future. As mentioned in Section 2, the number of these steps is the forecast horizon, 90 in our case; and 3) **eval_metric=sMAPE**: trained models are ranked based on their performance on an internal validation set,

which is constructed by holding out the last `prediction_length` timesteps of each time series in the training dataset. So, the rank is based in our prediction target, $\mathbf{y}_{t+90}$.

By default, AutoGluonTS makes its forecasts using the model with the best rank, usually the WeightedEnsemble model. To generate our prediction, we must compute the class (*above normal*, *normal*, or *below normal*) of the last predicted target (`prediction_length=90`), using the terciles of the reference period, as we explained in Section 2.

3 Results

We run the AutoGluonTS `high_quality` preset in a high performance server, with the following specifications: Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz, with 40 CPU cores and 96GB RAM.

A simple evaluation procedure could consist of fitting a single AutoGluonTS for all days in the training dataset, and evaluating its performance at each day in the testing dataset. Considering that our testing dataset is quite large (12 months), and that this simple procedure does not use the last historical data to train where we are advanced in the testing data, we prefer a more sophisticated approach: At the beginning of each testing month, we train a new model with the historical information until the date, and evaluate the performance in the next three months. Therefore, for our testing dataset, we train 10 different models (we cannot use the last two months to train because we want to evaluate 3 months into the future), and evaluate them in the following 3 months.

Each model needs approximately 50 min to train in our hardware, so, after approximately 9 hours, we had our 10 models trained. Figure 2 shows the performance on the first 9 months, evaluated in the Weather Station Encruzilhada-do-Sul, Brazil (lon: -52.51, lat: -30.53, id: 83964). The red/yellow/green areas shows the terciles for the classes *above normal*/*normal*/*below normal*. The black line is the actual value, y_{t+90} (seasonal mean temperature in the following 90 days) and the blue line is the model prediction $\hat{y}_{t+90}$ for each t in the following three months. As expected, the model slowly loose quality when is evaluated farther from the training data. A model is better when actual and predicted lines share the same class area in many days and in many geographical points.

To globally measure the quality, we used the F1-macro, F1-micro, and AUC metrics. Figure 3a shows the One-vs-Rest multi-class ROC curve, where micro-averaged and macro-averaged One-vs-Rest ROC AUC scores are 0.65 and 0.57 respectively. The AUC discriminated by classes, as shown in the related work, is as follows: 0.64 for *above normal*, 0.55 for *normal* and 0.53 for *below normal*. The quality obtained is similar to the obtained in the reference work (Table 1) by sophisticated state-of-the-art models, despite the fact that the evaluation period and geographical points differ. The error is not distributed uniformly across the dates (as showed in the example of the Figure 2). Globally, the same happens on the average to most geographical points. Table 3b shows the error of the models at each time period, that is, the decrease in performance at the end of each one.

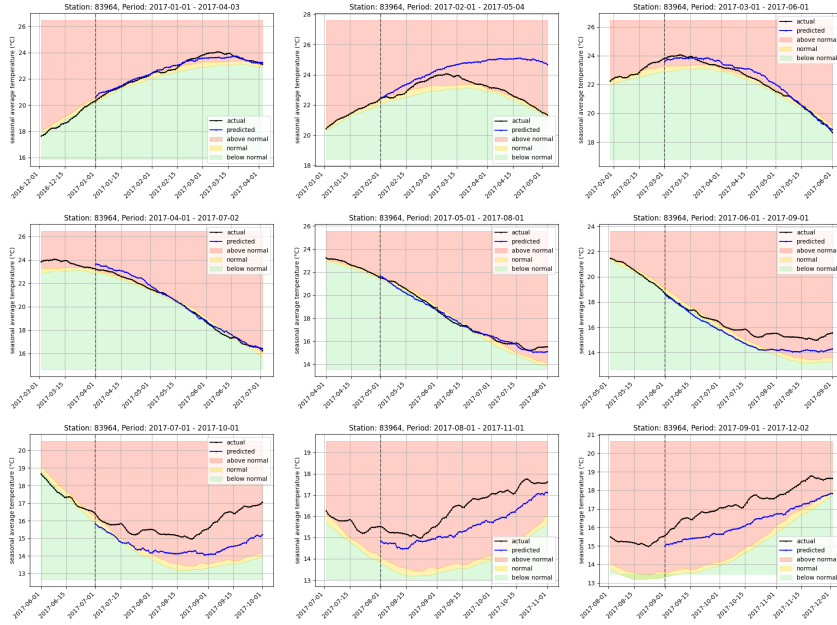


Fig. 2: Each figure shows the results for the first 9 months of 2017, in the weather station 83964: Encruzilhada-do-Sul, Brazil (lon: -52.51, lat: -30.53).

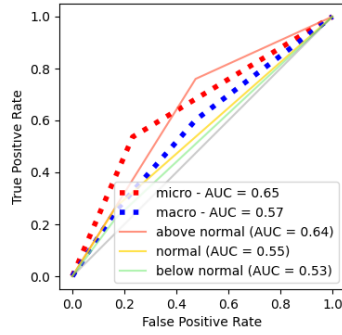
4 Conclusions

Forecasting seasonal temperatures is a great challenge, due to the irregular nature behind this physical parameter, and to the associated uncertainty. This work describes an approach using a powerful state-of-the-art automated ML (autoML) tool. Our main conclusion of the paper is that these modern techniques can compete with the sophisticated and heavy models used in production.

Concerning future work, we intend to study the impact of the forecasting period on the performances of these methods. We must analyze the sensitivity of their performance with respect to the position and size of the forecasting segment. We must also try to integrate to the learning process the fact that there are periods where forecasting temperatures is harder. We must also explore other forecasting procedures having in mind these seasonal aspects. Finally, the extreme difficulty of the problem seems to say that our variable set is not enough for a good performance. This must also be explored in deep.

References

1. M. M. Poggi, M. d. I. M. Skansi, and R. de Elía, “Metodología de verificación del pronóstico estacional por consenso,” *Servicio Meteorológico Nacional (Meteorological National Service)*, vol. 2021, p. 110, 2021.



(a) ROC curve.

Period start	Period end	F1-macro	F1-micro
2017-01-01	2017-04-03	0.42	0.53
2017-02-01	2017-05-04	0.27	0.61
2017-03-01	2017-06-01	0.45	0.58
2017-04-01	2017-07-02	0.34	0.60
2017-05-01	2017-08-01	0.29	0.64
2017-06-01	2017-09-01	0.41	0.74
2017-07-01	2017-10-01	0.48	0.74
2017-08-01	2017-11-01	0.34	0.63
2017-09-01	2017-12-02	0.17	0.18
2017-10-01	2018-01-01	0.10	0.13

(b) Error discriminated by date

Fig. 3: Global quality measures: (a) Receiver Operating Characteristic to One-vs-Rest multi-class. Micro-averaged and Macro-averaged One-vs-Rest ROC AUC scores are 0.65 and 0.57 respectively; (b) F1-macro and F1-micro for each particular period.

2. A. Alsharef, K. Aggarwal, Sonia, M. Kumar, and A. Mishra, "Review of ML and AutoML solutions to forecast time-series data," *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 5297–5311, 2022.
3. Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li and Alexander Smola, "AutoGluon-tabular: Robust and accurate autoML for structured data," arXiv:2003.06505, 2020.
4. Simon J. Mason, Michael K. Tippett, Lulin Song, Ángel G. Muñoz, "Climate predictability tool version 18.1.2," <https://doi.org/10.7916/39fx-at72>, 2023, Columbia University Academic Commons.
5. M. Osman, C. Coelho, and C. Vera, "Calibration and combination of seasonal precipitation forecasts over south america using ensemble regression," *Clim Dyn*, vol. 57, pp. 2889–2904, 2021.
6. D. Wilks, "Extending logistic regression to provide full-probability-distribution MOS forecasts. meteorological applications," *Meteorological Applications*, vol. 2009, pp. 361–368, 2009.
7. S. Mason, "Guidance on verification of operational seasonal climate forecasts," World Meteorological Organization, Commission for Climatology XIV Technical Report., Tech. Rep., 2018.
8. T. B. McKee, N. J. Doesken, and J. Kleist, "Drought monitoring with multiple time scales, 9th conference, applied climatology," in *Conference on applied Climatology, Applied climatology, 9th Conference, Applied climatology*. The Society;, 1995, pp. 233–236. [Online]. Available: <https://www.tib.eu/de/suchen/id/BLCP%3ACN008169111>
9. Z. Zhang and J. C. Moore, "Chapter 6 - Empirical Orthogonal Functions," in *Mathematical and Physical Fundamentals of Climate Change*, Z. Zhang and J. C. Moore, Eds. Boston: Elsevier, 2015, pp. 161–197. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128000663000061>
10. D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169207019301888>