Comparing Transformers and Linear models for precipitation forecast in Rio de Janeiro

Mauro S. S. Moura[1], Fabio A. M. Porto [1]

[1]*Data Extreme Laboratory, National Laboratory of Cientific Computing*
Venue Getulio Vargas 333, 25651-075, Petrópolis, Brazil
mauro@posgrad.lncc.br, fporto@lncc.br

Abstract. Precipitation significantly influences geographic regions and human activities, especially in areas like Rio de Janeiro, known for its unstable weather conditions. Accurate forecasting of extreme precipitation events is essential for mitigating adverse impacts. This study assesses the predictive effectiveness of linear models and transformer-based approaches, using Artificial Neural Networks (ANNs), specifically the Autoformer model, which is recognized for its state-of-the-art performance in time series forecasting. Given the computational demands of transformer models, simpler alternatives like DLinear are explored for efficiency. The dataset, obtained from INMET, includes historical records from 2002 to 2023, comprising 18 variables and 425,733 non-null values, including 34,386 instances of precipitation. Despite the challenge of data imbalance, particularly with extreme events exceeding 25 mm, the dataset was used in its unbalanced form for regression analysis. Data was split into training (60%), validation (20%), and test (20%) subsets, with time series segmented into windows of 96 for training, validation, and testing. Both Autoformer and DLinear were trained with identical parameters, and the mean squared error (MSE) was used as the primary metric. This study aims to improve the understanding of model performance in precipitation forecasting, providing insights into the optimal architecture for unbalanced datasets.

Keywords: Deep Learning; Transformers; Precipitation Forecasting; Extreme Events;

1 Introduction

In 2017, the Transformer architecture [1] emerged as a groundbreaking neural network model characterized by its encoder-decoder structure and attention mechanisms, rapidly becoming a standard in natural language processing [2]. This success spurred interest in applying Transformer-based models to time series prediction, yielding significant advancements [3]. Notably, the Autoformer model [4] achieved state-of-the-art performance in this domain. However, due to the substantial computational resources required by Transformer models, simpler alternatives such as DLinear [5] have been proposed to deliver comparable or superior results with reduced complexity.

Precipitation forecasting is a critical meteorological task that impacts various sectors globally, particularly in predicting extreme rainfall events, which are essential for mitigating adverse effects [6]. While Transformer-based models have demonstrated efficacy in simpler forecasting tasks such as traffic and temperature prediction, their performance in precipitation forecasting has been relatively inferior [7].

This study aims to evaluate the applicability of Transformer-based models, specifically Autoformer, in comparison to the linear DLinear model for precipitation prediction. Utilizing spatiotemporal rainfall data from Rio de Janeiro, sourced from the INMET meteorological system, the dataset spans historical records from 2002 to 2023. It includes 18 meteorological variables and over 425,000 non-null entries, with 34,386 instances of precipitation exceeding 0 mm. To address the inherent data imbalance, particularly regarding extreme precipitation events (greater than 25 mm), the dataset is treated as a regression problem. It is partitioned into training (60%), validation (20%), and test (20%) subsets, comprising 255,248, 84,955, and 84,957 time series, respectively, with a window size of 96.

The research evaluates the performance of both model architectures in predicting precipitation, contributing to the understanding of their effectiveness in regression tasks involving unbalanced datasets. The findings aim to provide insights into the optimal model choice for accurate precipitation forecasting.

2 Theoretical Background

2.1 Deep Learning with time series regression

A time series is a sequence of data points arranged chronologically, typically recorded at uniform intervals. Examples of time series data include daily stock prices and hourly temperature readings. Time series data generally exhibit specific characteristics such as trend, seasonality, and noise [8]. The trend refers to the long-term progression or overall direction of the data over time. Seasonality encompasses regular patterns or cycles that repeat at fixed intervals, such as daily, monthly, or yearly patterns. Noise represents random variations or irregularities in the data that do not follow any discernible pattern and cannot be attributed to trends or seasonality.

In this work, we utilize data from weather stations to forecast precipitation as a regression problem, since our approach is to obtain the actual precipitation values in millimeters. These stations collect a range of meteorological variables crucial for accurate forecasting, including temperature, humidity, wind speed and direction, pressure, precipitation, and solar radiation. By analyzing these variables over time, weather models can identify patterns and make predictions about future weather conditions. While radar and satellite data are often used to enhance forecast accuracy, this study focuses solely on weather station data to evaluate its sufficiency for accurate precipitation prediction.

To address time series problems, various computational models are commonly used. These models range from statistical approaches, such as AutoRegressive Integrated Moving Average (ARIMA) [9], Exponential Smoothing, Seasonal Decomposition of Time Series (STL), and Seasonal ARIMA (SARIMA) [9], to machine learning models, including Linear Regression, Decision Trees, and Random Forests. Additionally, deep learning models like Recurrent Neural Networks (RNNs) [10], Long Short-Term Memory (LSTM) Networks [11], Gated Recurrent Units (GRUs) [12], Convolutional Neural Networks (CNNs) [13], and Transformer Models [1] are frequently employed.

These computational models are widely utilized to predict precipitation, a crucial task for various applications such as agriculture, water resource management, and disaster preparedness. Deep learning models, particularly Transformer models, can extract spatial features from meteorological data, providing a more comprehensive understanding of precipitation patterns.

2.2 Models

Three models were used in this study: Seasonal Autoregressive Integrated Moving Average with Exogenous Variables (SARIMAX), Autoformer and DLinear. More details about each model will be provided in this section.

2.3 SARIMAX

AutoRegressive Integrated Moving Average (ARIMA) is a statistical model utilized for the analysis and forecasting of univariate time series data, integrating three fundamental components: the autoregressive component, which captures the linear dependence between an observation and its past lags; integration, referring to the differencing of observations to achieve stationarity by removing trends and seasonality; and the moving average component, which models the dependence between an observation and the forecast errors from previous lags. Extending the ARIMA framework, the SARIMAX model accommodates both seasonal patterns and external predictor variables, thereby enhancing its capability to model time series data exhibiting seasonal fluctuations and influenced by external factors. This comprehensive approach allows SARIMAX to effectively capture complex temporal structures and external influences, making it widely applicable in diverse fields such as economics, finance, engineering, and environmental science for generating more accurate and reliable forecasts.

DLinear

The DLinear model is a linear model proposed by Zeng et al. [5]. It includes a seasonal decomposition layer, inspired by Autoformer Wu et al. [4] and FEDformer Zhou et al. [14], which extracts seasonal features from the data. This means the input data is decomposed into a seasonal component and a cyclical trend, as defined by eq. (1). Following this decomposition, there is a linear layer for each input, and finally, the outputs of these linear layers are summed to get the final prediction. When the data exhibits a clear trend, DLinear explicitly accounts for it, thereby improving the performance of a basic linear model Zeng et al. [5]. The model operates in two modes: individual and non-individual. In individual mode, a linear layer is assigned to each variable, in both cases the seasonal decomposition is calculated using full input data.

$$X_{trend} = AvgPool(Padding(x)),$$
$$X_{sazonal} = X - X_{trend}. \tag{1}$$

The DLinear implementation can be found on github[1].

Autoformer

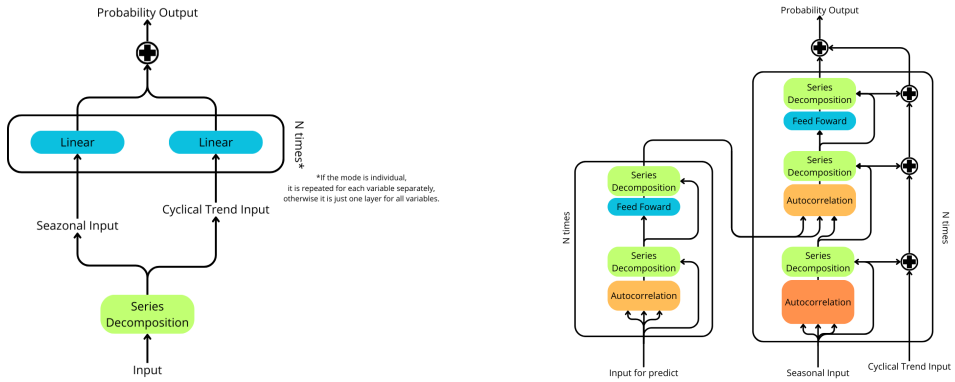
The Autoformer model proposed by Wu et al. [4] is specifically designed for time series forecasting, leveraging the strengths of the Transformer architecture. It is known for its capability to capture long-range dependencies through self-attention mechanisms. The key innovation in Autoformer lies in its ability to decompose time series data into trend and seasonal components, as shown in eq. (1), and its novel Auto-Correlation layer. The architecture consists of three main components:

- Decomposition Block: This component decomposes the time series data into trend and seasonal components.
- Encoder-Decoder Structure: Similar to traditional Transformer models, the encoder processes the historical time series data, while the decoder generates future time series values based on the encoded information.
- Auto-Correlation Mechanism: This mechanism replaces the conventional self-attention mechanism used in Transformers. It is designed to efficiently capture long-term dependencies by focusing on periodic patterns in the data. Unlike the traditional self-attention mechanism, which computes pairwise attention scores for all time steps, the auto-correlation mechanism identifies and leverages periodic correlations in the data. This approach significantly reduces computational complexity while enhancing the model's ability to capture long-range dependencies.

The Autoformer implementation can be found on github [2].

Model Comparison

Comparing the architectures of DLinear (Fig. 1a) and Autoformer (Fig. 1b), it is evident that the DLinear model is significantly simpler than Autoformer. In contrast, the Autoformer model leverages the encoder-decoder characteristics typical of transformers, which may indicate that the model is capable of detecting more complex patterns in the data than the linear model.



(a) DLinear archtecture. (b) Autoformer archtecture.

Figure 1. Models architectures.

3 Methodology

The models were implemented using Python 3.8 with the PyTorch framework. The repository containing the code used in this study is available on GitHub[3]. The dataset can be accessed through the INMET website.

[1] https://github.com/cure-lab/LTSF-Linear
[2] https://github.com/thuml/Autoformer
[3] https://github.com/mauro-moura/precipitation-forecast

The dataset was obtained via the Rionowcast DataLake project [15], which gathers weather station data from the Instituto Nacional de Meteorologia (INMET)[4]. INMET operates several weather stations in the city of Rio de Janeiro. The data collected by these stations are available through various INMET platforms, including INMET Tempo and INMET Mapas de Estações. The dataset used in this study covers the period from 2002 to 2023 and includes four stations located in Forte de Copacabana, Vila Militar, Jacarepaguá, and Marambaia. The dataset contains meteorological variables such as temperature, humidity, wind speed, wind direction, and precipitation, with the latter being the target variable.

In this initial approach, rows containing any missing values were dropped from the dataset. A histogram displaying the distribution of the numerical variables and their occurrences is shown in Fig. 2.

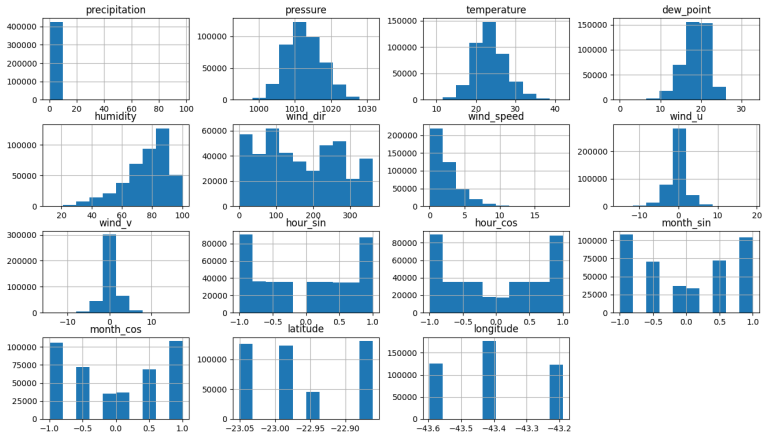


Figure 2. Histogram of numerical variables.

The data were preprocessed in two distinct ways. First, a minimally intrusive method was applied, using a *LabelEncoder* for categorical variables and a *MinMaxScaler*, with a range of 0.1 to 0.9, on all columns, including the categorical variables transformed by the *LabelEncoder scikit-learn*. The scaler range was empirically defined and based on studies demonstrating that this range can yield improved results for time series forecasting [16]. Categorical columns, such as *station_id* and *station_name*, were also included in this scaling process.

Second, to explore a potentially better spatial representation, the latitude and longitude information of each station was used to calculate the distance from an arbitrary central point at (0,0), which does not correspond to any actual station but serves as a reference point. This calculation was performed using the Haversine function [17], as expressed in eq. (2):

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right), \tag{2}$$

where r is the radius of the sphere (Earth, in this case), and ϕ_1, ϕ_2 are the latitudes of points 1 and 2, respectively (in radians), while λ_1, λ_2 are the longitudes of points 1 and 2, respectively (in radians).

After preprocessing, several variables were removed, and the data were ordered chronologically, with one row per location for each sequential date.

All models were trained under the same conditions for 20 epochs. The dataset was split into training (60%), validation (20%), and test (20%) sets. The parameters were selected empirically, and based on the literature, hyperparameter tuning was not prioritized. Instead, initial runs were used to determine the parameter set, and PyTorch's default function values were employed when no significant differences were observed. The model was trained with a batch size of 32, an input sequence length of 96, an output sequence length of 96, and a learning rate of 0.005.

Two callbacks were employed during model training: Early Stopping and ReduceLROnPlateau. Early Stopping halts the training process if no improvement is observed over a predefined number of epochs, with the patience parameter set to 3 for this experiment. Similarly, ReduceLROnPlateau decreases the learning rate if no improvement is detected, with a patience setting of 1. Model performance was evaluated using the Mean Squared Error (MSE), defined in eq. (3).

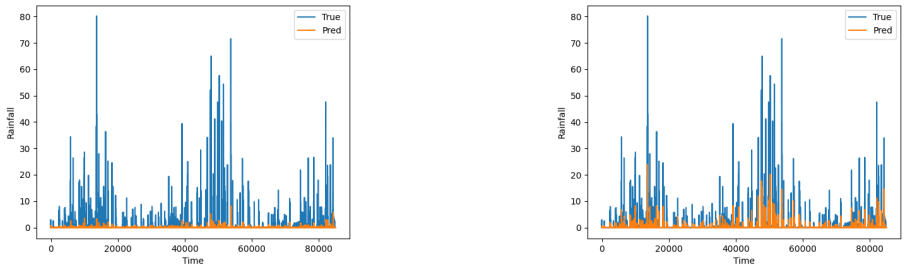
[4] `https://mapas.inmet.gov.br/`

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

The MSE metric accounts for the mean of squared errors, which, although not ideal for unbalanced problems, is commonly used in the literature and is thus considered a suitable metric for this study. However, it is important to note that MSE as a loss function does not inherently address the issue of unbalanced data. Currently, there is no widely accepted loss function specifically designed for dealing with unbalanced regression tasks. As part of future work, this study aims to explore alternative metrics to better handle unbalanced data in regression problems.

4 Results and Discussion

Autoformer and DLinear were trained using a seed value of 2021 to ensure reproducibility. Additionally, the SARIMAX model was selected as a benchmark to compare the results against a commonly used method for forecasting. On the test set, the Autoformer model achieved a Mean Squared Error (MSE) of 1.9089, while the DLinear model achieved a lower MSE of 1.5028, and the SARIMAX model obtained an MSE of 2.1189. These results suggest that the linear DLinear model was able to capture correlations in the data as effectively as the more complex Transformer-based Autoformer model. The full time series predictions from both models are illustrated in Fig. 3.



(a) Autoformer test results.

(b) Dlinear test results.

Figure 3. Precipitation per date.

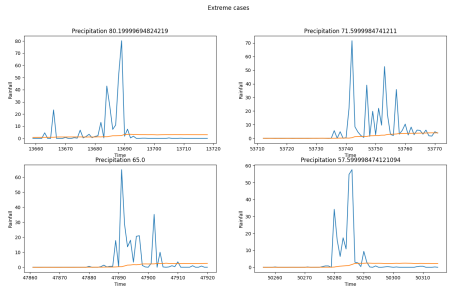
The DLinear model appears to have tracked the precipitation values more closely, although it did not capture the exact magnitude. Since this is a case with imbalanced data, we chose to evaluate the MSE by precipitation bins, divided into 0 to 10mm, 10 to 25mm, 25 to 50mm, and 50mm or more. The values for this simulation are presented in Table 1.

Table 1. MSE per bins for first execution

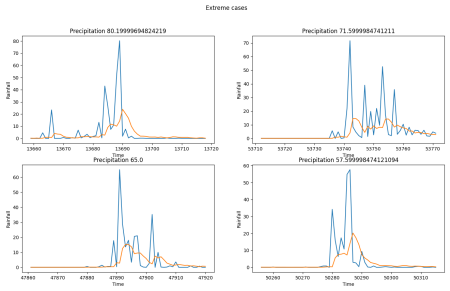
Model	MSE 0-10mm	MSE 10-25mm	MSE 25-50mm	MSE 50mm+
DLinear	**0,325**	**156,2437**	**863,8895**	**2806,2461**
Autoformer	0,4345	207,7265	1053,6511	3390,4949
SARIMAX	0,4903	243,2032	1147,6152	3566,6263

We opted to check some of the precipitation peaks in the test dataset. The highest precipitation found truth and prediction for each model in the initial experiments is found on Fig. 4.

This initial result indicates that DLinear produced better outcomes, coming closer to the peak values than Autoformer. Overall, Autoformer appears to smooth the results, which may be attributed to the attention mechanisms attempting to identify correlations with variables that are not genuinely correlated. Given that both models utilize a similar approach to the input data, this behavior could explain the discrepancy. Additionally, the SARIMAX method exhibited the poorest performance among the models tested.



(a) Autoformer highest peak.



(b) Dlinear highest peak.

Figure 4. Precipitation highest peak.

To improve the model, we experimented with another preprocessing technique. The second approach aimed to enhance the spatiotemporal representation by creating a substitute for latitude and longitude called ”distance,” calculated using the haversine function. In this test, SARIMAX achieved an MSE of 2,1291, DLinear achieved an MSE of 1.5028, while Autoformer had an MSE of 1.9110. The MSE per bin is presented in Table 2.

Table 2. MSE per bins for second execution

Model	MSE 0-10mm	MSE 10-25mm	MSE 25-50mm	MSE 50mm+
DLinear	**0,325**	**156,2437**	**863,8895**	**2806,2461**
Autoformer	0,4362	207,83	1053,8129	3392,0891
SARIMAX	0,4991	243,5218	1148,3915	3568,1091

We observe that, for DLinear, the results remain identical in both executions 1 and 2 using the same seed, suggesting that DLinear is primarily considering the target variable while disregarding the additional variables, indicating a lack of learning from the other inputs. Similarly, Autoformer does not appear to fully leverage the other variables either. The SARIMAX model also maintained the same result as before, showing no improvement with changes in data representation.

Lastly, we attempted to use Autoformer with a single variable to see if the results would improve, but it resulted in an MSE of 1.8741. The MSE per bin was 0.43073 for 0-10mm, 203.6539 for 10-25mm, 1036.6064 for 25-50mm, and 3289.4680 for 50mm+. In general, this approach had better results per bin than the others for Autoformer.

With these models and this type of data, the DLinear model was able to represent the results more accurately. By analyzing the graphs of the highest precipitation peaks, we observe that DLinear had a greater influence on the precipitation values, while Autoformer tended to smooth the data, leading to poorer predictions for extreme spatiotemporal events. The SARIMAX model, in comparison, produced the least accurate results, further indicating its limitations in handling this type of dataset.

This outcome may be attributed to the extreme nature of the data, as no balancing methods were applied, either through data augmentation or by using specific metrics tailored to unbalanced data. Based on these observations, we conclude that the DLinear model generally produced better results. However, the data from weather stations alone was not sufficient for highly accurate precipitation prediction, highlighting the need for additional data sources or advanced techniques to improve model performance.

5 Conclusions

In conclusion, it is important to acknowledge that the results obtained are specific to the models and dataset utilized in this study. To ensure the reproducibility of the experiment, it is necessary to implement additional models with different parameter configurations. Moreover, given the inherent difficulty of precipitation forecasting, exploring alternative prediction approaches, integrating supplementary data sources, and refining the preprocessing of input data may yield results where more complex models can fully demonstrate their potential.

For future work, we will focus on training additional models, including those utilizing spatiotemporal inputs rather than only multivariate ones. Furthermore, we plan to perform time series cross-validation to assess whether the models improve with larger datasets. Testing the models on data from different geographic regions will also be critical in evaluating their adaptability.

Alongside these experiments, we aim to delve deeper into the correlation between variables to identify any that may be detracting from the models' predictive accuracy, as observed in the initial results. While this precipitation forecasting problem has been approached as a regression task, it is feasible to explore alternative methods, such as classification and extreme event detection. Another challenge that must be addressed is the issue of data imbalance, which poses difficulties in regression tasks. However, several potential solutions, such as data augmentation, can be implemented to mitigate this problem.

Acknowledgements. The authors would like to thank the National Council for Scientific and Technological Development (CNPQ) for funding the research and the support of the RioNowcast project.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, vol. 30, 2017.

[2] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, vol. , 2022.

[3] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, vol. 32, 2019.

[4] H. Wu, J. Xu, J. Wang, and M. Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, vol. 34, pp. 22419–22430, 2021.

[5] A. Zeng, M. Chen, L. Zhang, and Q. Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

[6] M. V. d. J. Pristo, C. P. Dereczynski, P. R. d. Souza, and W. F. Menezes. Climatologia de chuvas intensas no município do rio de janeiro. *Revista Brasileira de Meteorologia*, vol. 33, n. 4, pp. 615–630, 2018.

[7] R. Castro, Y. M. Souto, E. Ogasawara, F. Porto, and E. Bezerra. Stconvs2s: Spatiotemporal convolutional sequence to sequence network for weather forecasting. *Neurocomputing*, vol. 426, pp. 285–298, 2021.

[8] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer, 4th edition, 2017.

[9] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[10] J. L. Elman. Finding structure in time. *Cognitive science*, vol. 14, n. 2, pp. 179–211, 1990.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, vol. 9, n. 8, pp. 1735–1780, 1997.

[12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, vol. , 2014.

[13] S. Becker, Y. Le Cun, and others. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 connectionist models summer school*, pp. 29–37, 1988.

[14] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

[15] F. Porto, M. Ferro, E. Ogasawara, T. Moeda, de C. D. T. Barros, A. C. Silva, R. Zorrilla, R. S. Pereira, R. N. Castro, J. V. Silva, and others. Machine learning approaches to extreme weather events forecast in urban areas: Challenges and initial results. *Supercomputing Frontiers and Innovations*, vol. 9, n. 1, pp. 49–73, 2022.

[16] P. G. S. d. SANTOS and others. *Previsão de variáveis ambientais na Amazônia com uso de redes neurais artificiais do tipo Long Short-Term Memory.* PhD thesis, Universidade Federal do Oeste do Pará, 2021.

[17] K. Gade. A non-singular horizontal position representation. *The journal of navigation*, vol. 63, n. 3, pp. 395–417, 2010.