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

Mauro Sérgio & Fábio Porto

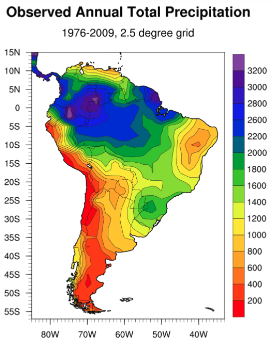
Agenda

Outline

Precipitation forecasting

- Precipitation forecasting is a **complex and critical task** in meteorology.
- Accurate prediction is challenging due to the **unbalanced distribution** of precipitation events.
- Rain prediction involves forecasting future weather conditions based on past data, which is typically structured as a **time series**.

Figure: Rainfall in South America.

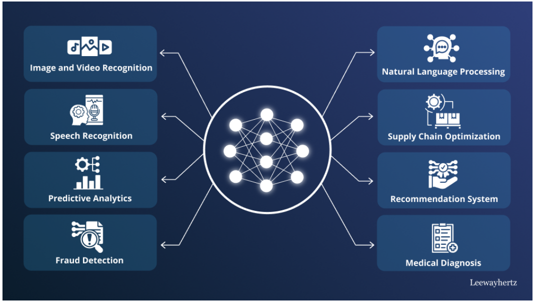


Source: `https://commons.wikimedia.org/wiki/File:South_America_Precipitation_Map_(data_from_1976-2009).gif`

Deep Learning (DL)

- Deep Learning (DL) has revolutionized various fields by enabling models to **automatically learn complex patterns** from large datasets.
- Deep learning can increase the accuracy of precipitation forecasts by capturing both **spatial and temporal dependencies** in the data, as well as taking advantage of neural network architectures, especially those based on the **Transformer**.

Figure: Deep Learning.



Source: `https://www.leewayhertz.com/what-is-deep-learning/`

Problem Formalization

In this work, we deal with **Tabular Multivariate Time Series Data and Multivariated models**.
With this we aim to: **Evaluate the performance of Linear and Transformer architectures** in predicting precipitation, contributing to the understanding of their effectiveness in regression tasks involving unbalanced datasets.

Outline

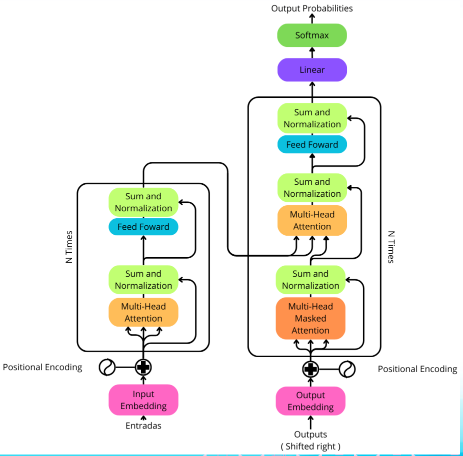
Transformer Models

XLV CILAMCE
11 – 14 November 2024
Maceió / AL / BRAZIL

Transformer models are a major advancement in deep learning, they use a self-attention mechanism to **process entire sequences at once**. Some Key features are:

- **Self-Attention Mechanism**: Helps the model focus on important parts of the input, capturing long-range dependencies.
- **Parallel Processing**: Allows faster training and more efficient handling of large datasets compared to recurrent models.

Figure: Transformer Architecture.

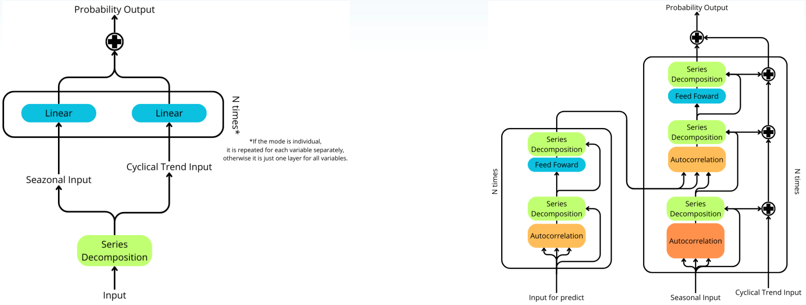


Learners

Table: Distinguish features between learners.

Learners	Distinctions	Architecture Type	Prediction Type
Autoformer	Decomposition layer, Auto-correlation, Time Delay Aggregation	Transformer	Temporal Multi-variate
DLinear	Linear model with Seasonal Decomposition	Linear	Multivariate
SARIMAX	Linear Seasonal model highly utilized in TSF	Linear	Multivariate

Models architecture comparison



(a) DLinear archtecture.

(b) Autoformer archtecture.

Figure: Models architectures.

Outline

Dataset (Rionowcast)

This dataset was obtained from the Rionowcast DataLake project and consists of weather station data collected from INMET weather stations in Rio de Janeiro city. It includes:

- Data from 2002 to 2023.

Dataset (Rionowcast)

This dataset was obtained from the Rionowcast DataLake project and consists of weather station data collected from INMET weather stations in Rio de Janeiro city. It includes:

- Data from 2002 to 2023.
- Time resolution: 1 hour.

Dataset (Rionowcast)

This dataset was obtained from the Rionowcast DataLake project and consists of weather station data collected from INMET weather stations in Rio de Janeiro city. It includes:

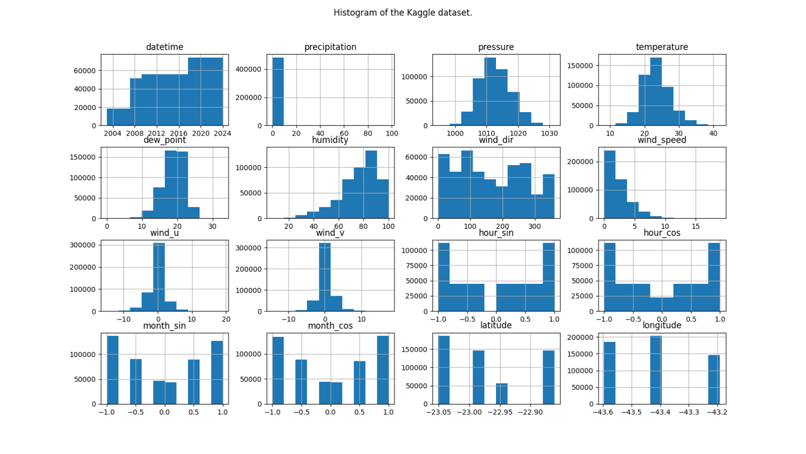
- Data from 2002 to 2023.
- Time resolution: 1 hour.
- Task: Precipitation regression.

Rionowcast Variables

Variable name	Description	Unity
datetime	Datetime of the measurement	
station_id	ID of the weather station	
precipitation	Amount of rainfall recorded by the stations aggregated by hour	mm
pressure	Atmospheric pressure	mB
temperature	Temperature	$^\circ C$
dew_point	Dew point temperature	$^\circ C$
humidity	Relative humidity	%
wind_dir	Clockwise wind direction	$^\circ$
wind_speed	Wind speed	m/s
wind_u	Cyclic U component from the wind	
wind_v	Cyclic V component from the wind	
datetime encoding	Sine and cosine components for hour (hour_sin, hour_cos) and month (month_sin, month_cos)	
stations	Station name	
latitude	Station Latitude	
longitude	Station Longitude	

Rionowcast Variables Histogram

Figure: Histogram of numerical variables.



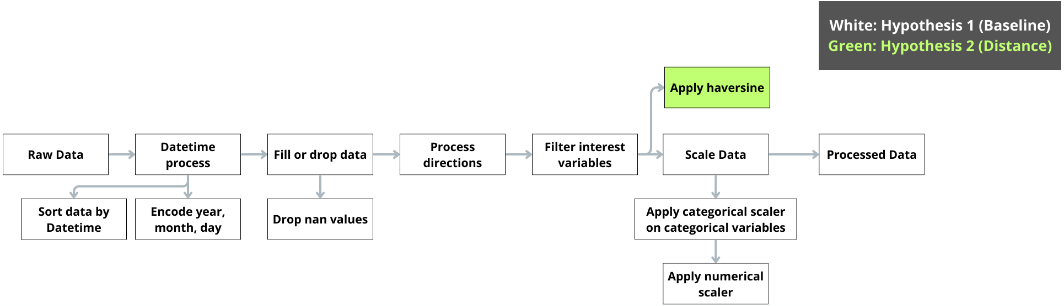
Datasets Considerations

In addition to the previously mentioned analyses, further evaluations were conducted to assess the quantity of **missing values** in the dataframe, as well as the **starting date of operations** for each station.

It is important to highlight that the locations **had different starting dates per station**, and there were **numerous missing values**.

Data Processing Pipeline

Figure: Data preprocessing pipeline.



Data Split

- The dataset was split as: 60% for training, 20% for validation and 20% for test **sequentially**.

Figure: Training Split.

Learner parameters

- Input Sequence: 96
- Output Sequence: 96
- Epochs: 50
- Optimizer: Adam with Learning Rate of 0.005.
- Callbacks: EarlyStopping, ReduceLROnPlateau, with patiences of 3 and 1 sequentially.

The learner training process was conducted on the DEXL laboratory machine Netuno, a Dell Precision T7820 With:

- $2\times$ CPU Intel Xeon Silver 4216 (16 CPU-core each)
- 126GB of RAM DDR4
- $2\times$ NVIDIA RTX A5000 with 24GB of VRAM GDDR6 with NVLink.

Metrics

Mean Squared Error:

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

Metrics

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.

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.

Outline

Experimental Setup

- Three models were selected for evaluation:
  - **DLinear**: Chosen for its simplicity and low computational requirements.
  - **Autoformer**: Selected as a Transformer-based extension of DLinear.
  - **SARIMAX**: Included as a widely used model for time series forecasting.
- Results are presented in the following format:
  - Actual and predicted values are segmented into four precipitation bins, with MSE calculated for each bin and presented in tabular form.

First Execution

Table: MSE per bins for first execution

Model	MSE	MSE 0-10mm	MSE 10-25mm	MSE 25-50mm	MSE 50mm+
DLinear	**1.5028**	**0,325**	**156,2437**	**863,8895**	**2806,2461**
Autoformer	1.9089	0,4345	207,7265	1053,6511	3390,4949
SARIMAX	2.1189	0,4903	243,2032	1147,6152	3566,6263

Second Execution (With Distances)

Table: MSE per bins for second execution

Model	MSE	MSE 0-10mm	MSE 10-25mm	MSE 25-50mm	MSE 50mm+
DLinear	**1,5028**	**0,325**	**156,2437**	**863,8895**	**2806,2461**
Autoformer	1,9110	0,4362	207,83	1053,8129	3392,0891
SARIMAX	2,1291	0,4991	243,5218	1148,3915	3568,1091

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 only**.

Implementation of Various Models

Table: Baseline Models and Transformer-based Models.

Model	MSE	MSE 0-10mm	MSE 10-25mm	MSE 25-50mm	MSE 50mm+
DLinear	**1.5028**	0.3250	156.2437	863.8895	2806.2461
SARIMAX	2.1189	0.4903	243.2032	1147.6152	3566.6263
Mamba	1.9717	0.3596	237.7751	1136.0001	3557.9128
TSMixer	2.3053	0.8463	203.8534	1053.0809	3353.6857
PrecipLSTM	**1051.6258**				
Transformer	1.9931	0.4018	232.5584	1124.3771	3538.4458
Autoformer	1.9089	0.4345	207.7265	1053.6511	3390.4949
Informer	2.0064	0.4200	231.4138	1121.9215	3532.7129
Mambaformer	1.9693	0.3599	237.1345	1134.4648	3554.9453

Execution Time of Various Models

Table: Training time by model.

Models	Total Execution Time	Number of Epochs	Time per Epoch
DLinear	**10 minutes**	15	43 seconds
Mamba	20 minutes	12	1 minute
TSMixer	29 minutes	15	2 minutes
PrecipLSTM	**33 hours 41 minutes**	8	4 hours 30 minutes
Transformer	21 minutes	4	5 minutes
Autoformer	1 hour 35 minutes	17	7 minutes
Informer	28 minutes	5	5 minutes
Mambaformer	53 minutes	14	3 minutes

Outline

Conslusions

In this work, we:

- Applied Transformer-based and Baseline state of the art Deep Learning models to the challenge of precipitation prediction.
- Assessed various data sources and evaluated the **reliability of meteorological stations** in supporting accurate predictions.

Conslusions

- The models faced difficulties in accurately predicting extreme precipitation events.
- Modifying the data representation had a low effect on performance, reinforcing that addressing the unbalanced data distribution remains crucial.
- The primary challenge in rainfall prediction lies in the unbalanced nature of the data, rather than in the architecture or data representation itself.

Future Work

- Explore data augmentation techniques for time series and refine loss functions to better address data imbalance.
- Integrate additional data sources (e.g., radar, weather balloons, ERA5), incorporating atmospheric variables and altitude information.
- Investigate the impact of different temporal scales to enhance predictive performance.
- Evaluate alternative approaches, such as classification or extreme event detection.
- Implement specialized models for spatiotemporal time series forecasting.

Acknowledgments

Laboratório Nacional de Computação Científica
UNIDADE DE PESQUISA DO MCTI

DEXL
DATA EXTREME LAB

RIO NOWCAST

CAPES

BR PETROBRAS

Thank You for the Attention

Contact:
mauro@posgrad.lncc.br