Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Machine learning approaches for short-term weather forecasting from a local weather station

Carolina Estrella Machado

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Vinícius Ruela Pereira Borges

Brasília
2023

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Machine learning approaches for short-term weather forecasting from a local weather station

Carolina Estrella Machado

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Vinícius Ruela Pereira Borges (Orientador)
CIC/UnB

Prof. Dr. Jan Mendonca Correa
CIC/UnB

Prof. Dr. Lucas Moreira
EST/UnB

Dr. Kleber Renato da Paixão Ataíde
INMET

Prof. Dr. Marcelo Grandi Mandelli
Coordenadora do Bacharelado em Ciência da Computação

Brasília, 19 de dezembro de 2023

Dedicatória

Dedico este trabalho à meus familiares, amigos e meu namorado que me apoiaram durante essa longa caminhada.

Agradecimentos

Gostaria de agradecer primeiramente ao CNPq por ter financiado esta pesquisa, processos números 144451/2021-2 e 135189/2022-5.

Agradeço ao meu orientador Prof. Dr. Vinícius Ruela Pereira Borges por todo o apoio durante todo o processo de construção deste trabalho, desde quando se iniciou como uma pesquisa de Iniciação Ciêntífica. Obrigada pelo imenso incentivo.

Agradeço aos demais professores do departamento de Ciência da Computação que auxiliaram de forma direta ou indireta no desenvolvimento deste projeto.

Agradeço mais uma vez pelo apoio pessoal que tive da minha família, amigos e namorado. Obrigada por permancerem ao meu lado ao longo desse processo.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Machine learning approaches for short-term weather forecasting from a local weather station

Carolina E. Machado, and Vinícius R. P. Borges, *Member, IEEE,*

***Abstract*—Weather forecasting is a relevant task that affect the human activities, including agriculture, transportation, economy, environment, tourism and entertainment. The weather conditions are predicted based on complex mathematical models that require considerable computer resources and data from satellites, sensors and atmospheric simulations. The successful application of machine learning techniques for weather forecasting worldwide motivated us to explore these approaches on weather data from the Instituto Nacional de Meteorologia (INMET). The goal of this research is to compare different machine learning techniques for both weather time-series forecasting and rain prediction using data from a weather station located in Brasília, Distrito Federal. The weather forecasting is dealt as a regression task, while the rain prediction is dealt as a classification task. The proposed methodology consists of several steps including data collection, preprocessing, sampling via hold-out, hyperparameter optimization, regression and classification experiments, evaluation and discussion. As for the classification task, the Support Vector Machines (SVM) presented better results of f1-score when compared to other methods. The Bidirectional long short-term memory (BiLSTM) presented better results for the regression task when compared to other deep learning techniques.**

***Index Terms*—Weather forecasting, long short-term memory, bidirectional long short-term memory, support vector machine, random forest, transformers, regression, binary classification**

I. Introduction

The analysis of weather conditions and weather forecasting play a fundamental role in several fields such as agriculture [1] [2], aviation [3], public health [4], urban flood prediction [5] and environment preservation [6]. Traditionally, weather forecasting is performed based on the results of numerical models, devised to simulate the physical processes that take place in the atmosphere. These models consider as input a series of weather parameters collected from radars and weather stations in land, as well as remote sensing images from artificial satellites [7]. As a result, the meteorologists can monitor the weather conditions in real-time and predict weather-related events according to the international standards [8].

The run of numerical simulations require considerable computational power since they process simultaneously high amounts of complex data. Although supercomputers are employed in those tasks, this process is time-consuming and its reliability depends on the considered range, so short-term forecasts present higher accuracies when compared to the longer ones [9]. This scenario is appropriate to explore alternate and complementary approaches based on machine learning and computer vision for weather forecasting. Furthermore, several national meteorological organizations worldwide make their meteorological data available, which comprises satellite images, weather stations and radars.

The focus of this research is processing data collected from weather stations, since they are generated on a regular basis and easy to obtain in the underlying platforms of the weather agencies. Moreover, weather data can be collected from the weather stations in a tabular structure, in which the rows represent a measure of weather condition at a specific time, while the attributes correspond to the atmospheric parameters, such as humidity, temperature, wind speed, radiation etc. This structure is straightforward to be processed by computational techniques when compared to the images obtained from the satellite images, which demand additional feature engineering due to the high dimensionality of the acquired images.

An important aspect of weather data is the temporal nature of weather data, since a current measure obtained from a weather station depends on past measurements. In this sense, weather forecasting is often addressed as a multivariate time series [10] [11], in which deep learning methods appear as successful candidates for regression tasks [12]. Additionally, they can be used to simplify classification and regression tasks, when compared to common correction methods [13].

Recent literature has published many papers in the subject of machine and deep learning methods for short-term weather forecasting and rain prediction. Zhang et al. [14] proposed a Tiny-RainNet combination of convolutional neural networks (CNNs) with bidirectional long short-term memory (BiLSTM). Salman et al. [15] used single layer long short memory model (LSTM) and multi layers LSTM model to explore the use of intermediate weather variable related to accuracy prediction of weather forecasting. Chaudhary et al. [16] explored several machine learning and deep learning techniques for predicting rainfall for the next day, in which Random Forest achieved the best results.

The Instituto Nacional de Meteorologia (INMET) [1], the Brazilian National Institute of Meteorology, owns several weather stations distributed in the territory of Brazil and has the biggest network of stations of Latin America. The Institute's Meteorological Data Collection and Distribution System collects atmospheric characteristics of surface meteorological stations in real time as it is equipped with upper air sounding stations (radiosonde), which are manually operated. The weather data here considered come from which obtain the weather conditions for every hour.

Carolina E. Machado and Vinícius R. P. Borges are with the Departamento de Ciência da Computação from Universidade de Brasília, Brasília, DF, Brazil e-mail: carolina.machado@aluno.unb.br, viniciusrpb@unb.br.

[1] https://portal.inmet.gov.br/

Thus, we propose to explore the use of machine learning and deep learning techniques for weather forecasting using data from the weather station from Brasília, Distrito Federal. For that purpose, we deal weather forecasting as a regression task, since the weather data is a multivariate time series. The goal is to predict the incoming weather conditions (temperature, radiation, pressure, rain, etc) based on a number of past weather conditions. For that purpose, we evaluate and compare the performance of state-of-the-art deep learning techniques for a short-term weather forecasting, which includes Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM) and Transformers. Furthermore, classical machine learning approaches, such as Logistic Regression, Random Forest and Support Vector Machine (SVM), as well as Multilayer Perceptron (MLP) are employed for a binary classification task for rain forecasting, which indicates the event of rain prediction.

This paper is structured as follows. Section II reports some recent researches on literature concerning weather data analysis. Section III details the proposed methods and its constituting steps. Section IV describes the experimental results that were conducted on a dataset of a Brazilian weather station and using quantitative evaluation metrics. Section V discusses the obtained results. Section VI concludes this paper and discusses possibilities for future work.

II. Related work

For this section, a series of recent researches on the field of rain prediction and weather forecasting will be discussed. These tasks were used as keywords in order to find related works related to the proposed goals in this research. We selected some researches that explored deep learning techniques in time series forecasting problem on weather data.

Zhang et al [17] proposed an improvement of accuracy for forecasting models. The long short-term memory (LSTM) was used to build the corrected model along with the K-means clustering method, which was used to divide the samples. The experiment uses eight types of meteorological factors as inputs and the difference between the actual rainfall and the model-forecast rainfall as output. The relation between the model forecast and the actual rainfall were learned, allowing the effective correction of model-forecast rainfall after feature extraction and continuous debugging training of the LSTM model.

Salman et al [15] proposed a model for forecasting univariate weather variable and uses single layer Long Short Memory Model (LSTM) model and multi-layers LSTM model to explore the effect of intermediate weather variable related to accuracy prediction. As intermediates data this research used temperature, pressure, humidity and dew point and visibility as predicted data. Results show that the multi layer LSTM presented better values for validation accuracy than single layer LSTM, achieving 0.860 compared to 0.7243 for pressure variable. For conclusion, this research shows that the combination of predicted variable and intermediate variables can optimize forecasting accuracy in time series data model.

Zhang et al [13] proposed a model in order to improve the weather forecasting accuracy. The K-Means clustering method was used to separate the sample in four groups and each one was modelled by Long Short-Term Memory (LSTM). For this experiment, real-time rainfall data from the automated stations in East China and the near-coastal areas of China from the Central Meteorological Observatory of Shanghai were used as reference values. The evaluation parameter used after the model correction was the Root Mean Squared Error (RMSE) and threat scores (TS). After clustering, the RMSE decreased by 0.4. The TS of results was also improved by correcting each rainfall type.

Weyn et al [18] used a Convolutional Neural Network to build a weather prediction model. The CNN model is able to forecast changes of weather that are significantly notable when compared to the capability of the fundamental dynamical equation. The model proposed is able to forecast realistic atmospheric states at lead times of 14 days. In conclusion, as the authors propose a simple model, it was not capable of outperforming the operational weather model, but machine learning is an important tool for weather forecasting, in particular the CNNs.

We detail the proposed methodology for exploring the machine learning and deep learning techniques on binary classification and regression tasks using the weather data from an INMET station.

III. Proposed Methodology

The proposed methodology encompasses the following tasks:

- **Rain prediction:** a binary classification process that receives a single measurement of atmospheric parameters and predicts if there is a rain event or not;
- **Weather forecasting:** a regression task that takes into account the number of past atmospheric measurements and predicts all atmospheric conditions for the next hour.

We detail next the proposed methodology. Subsection III-A presents the dataset characteristics. Subsection III-B details the preprocessing step before training the model. Subsection III-C describes the classification models considered in this research. Subsection III-D presents the regression models.

The flowchart in Figure 1 illustrates the required steps for both rain predict and weather forecasting models. First, a preprocessing was required to remove irrelevant and redundant attributes as well as instances with missing values. After obtaining the preprocessed dataset, a sampling via Holdout is performed so that the data is divided in training (60%), validation (20%) and test (20%) sets. The training set is employed to provide knowledge regarding the weather patterns over time to the underlying models. The validation set is used for the hyperparameter optimization of the models. The test set is then used to evaluate the optimized models for the aforementioned tasks using the appropriate metrics.

A. Dataset

The dataset chosen for this research was obtained from the INMET website. The data was collected from an automatic weather station located in Brasília from 02/10/2021 to 08/10/2023, totalizing 21888 instances. Table I shows the

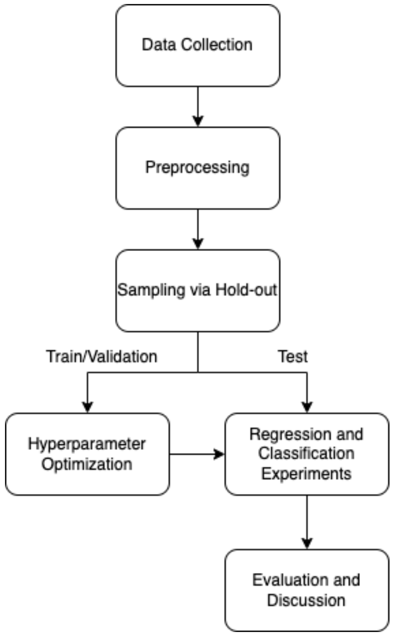


Fig. 1: Flowchart for the classification and regression tasks. Source: author's own.

TABLE I: Weather parameters of INMET stations describing the weather conditions.

Parameter	Description	Units
Date	Date of reference	Datetime
Hour	Hour of reference	UTC
Temperature	Instant	Celsius
Temperature	Maximum	Celsius
Temperature	Minimum	Celsius
Humidity	Instant	Percent
Humidity	Maximum	Percent
Humidity	Minimum	Percent
Dew point	Instant	Celsius
Dew point	Maximum	Celsius
Dew point	Minimum	Celsius
Pressure	Instant	hPa
Pressure	Maximum	hPa
Pressure	Minimum	hPa
Wind	Speed	m/s
Wind	Direction	Degree
Wind	Burst	m/s
Radiation	UV radiation	Kj/m²
Rain	Quantity	mm

weather parameters as well as their measurement units. Each measurement corresponds to a weather condition, which is characterized by 19 atmospheric parameters. Those parameters are all taken into account to design the classification and regression models.

B. Data preprocessing

First, a preprocessing is needed to prepare the dataset for the machine learning models. Figure 2 shows the dataset before preprocessing and figure 3 shows that same dataset after preprocessing. We replaced the null values to zero. As the atmospheric parameters are commonly represented as float values, we replaced the commas to dots in order to separate the whole-number from the decimal part, this was necessary due to the Brazilian format of the dataset. In "Date", the datetime stamp was replaced for a integer numeric representation of each date, in ascending order.

C. Classification models

Machine learning techniques have been successfully applied in various knowledge domains. Thus, our goal is to explore their performance on a binary classification task for predicting rain based on current atmospheric parameters. We chose the rain prediction task instead of the other atmospheric parameters due to its well-known potential on bringing shortcomings to the citizens when extreme events happens especially in urban environments. The literature has reported that rain prediction is a challenging task since it depends on various factors that cannot be captured only on surface weather stations [19] [20].

We describe below the considered state-of-the-art techniques for the devised rain prediction task. It is worth noting that the considered classification models are nowadays known as shallow learning due to its incapability of handling data presenting temporal characteristics [21] [22]. In this sense, the rain prediction task receives a measurement of the current weather conditions and outputs if there is rain or not.

1) Support Vector Machines (SVM): Figure 4 shows the architecture of the Support Vector Machines (SVM). Although SVM can be used for both classification and regression, in this research it will only be used for the first purpose.

Let T be a a training set with n data x_i and respective labels y_i, in which $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ is a multidimensional data instance and x_i is the value of an attribute and $\mathbf{y} = [y_1, y_2, \ldots, y_n]$ is in {+1,-1}. T is linearly separable if it is possible to separate data from classes +1 and -1 by a hyperplane. The equation of a hyperplane is presented by (1)

$$f(x) = \mathbf{w} \cdot \mathbf{x} + b = 0 \tag{1}$$

where $\mathbf{w} \cdot \mathbf{x}$ is the dot product of two vectors $\mathbf{w}$ and $\mathbf{x}$, which $\mathbf{w} \in X$ is the normal vector to the described hyperplane and b is the bias.

This equation divides the data space X into two regions: $\mathbf{w} \cdot \mathbf{x} + b > 0$ and $\mathbf{w} \cdot \mathbf{x} + b < 0$. A signal function $g(x) = sgn(f(x))$, as shown in Eq. (2) can then be used to obtain classifications, which in this case is the event of rain prediction.

$$g(x) = sgn(f(x)) = \begin{cases} +1 & \text{if} \quad \mathbf{w} \cdot \mathbf{x} + b > 0 \\ -1 & \text{if} \quad \mathbf{w} \cdot \mathbf{x} + b < 0 \end{cases} \tag{2}$$

Thus, if the dot product is positive, it means that it is located above the optimal hyperplan, otherwise, it should be located below, as shown in Figure 5.

	Data	Hora (UTC)	Temp. Ins. (C)	Temp. Max. (C)	Temp. Min. (C)	Umi. Ins. (%)	Umi. Max. (%)	Umi. Min. (%)	Pto Orvalho Ins. (C)	Pto Orvalho Max. (C)	Pto Orvalho Min. (C)	Pressao Ins. (hPa)	Pressao Max. (hPa)	Pressao Min. (hPa)	Vel. Vento (m/s)	Dir. Vento (m/s)	Raj. Vento (m/s)	Radiacao (KJ/m²)	Chuva (mm)
0	06/08/2022	0	19,9	21,1	19,3	40,0	45,0	37,0	6,0	7,0	5,5	887,4	887,4	886,9	1,3	124,0	1,9	NaN	0,0
1	06/08/2022	100	18,5	20,6	18,3	49,0	49,0	37,0	7,6	7,6	5,5	887,7	887,7	887,4	0,5	108,0	2,0	NaN	0,0
2	06/08/2022	200	18,7	21,4	18,3	46,0	49,0	38,0	6,8	7,7	6,3	887,8	887,9	887,7	1,6	115,0	4,2	NaN	0,0
3	06/08/2022	300	20,1	20,4	17,5	43,0	55,0	43,0	7,3	8,7	7,2	887,6	887,8	887,6	1,8	122,0	3,0	NaN	0,0
4	06/08/2022	400	19,7	20,5	19,1	41,0	43,0	39,0	6,2	7,3	5,8	887,5	887,7	887,4	1,6	98,0	2,8	NaN	0,0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Fig. 2: Dataset before preprocessing.

	Data	Hora (UTC)	Temp. Ins. (C)	Temp. Max. (C)	Temp. Min. (C)	Umi. Ins. (%)	Umi. Max. (%)	Umi. Min. (%)	Pto Orvalho Ins. (C)	Pto Orvalho Max. (C)	Pto Orvalho Min. (C)	Pressao Ins. (hPa)	Pressao Max. (hPa)	Pressao Min. (hPa)	Vel. Vento (m/s)	Dir. Vento (m/s)	Raj. Vento (m/s)	Radiacao (KJ/m²)	Chuva (mm)
0	8	0	19.9	21.1	19.3	40.0	45.0	37.0	6.0	7.0	5.5	887.4	887.4	886.9	1.3	124.0	1.9	0	0.0
1	8	100	18.5	20.6	18.3	49.0	49.0	37.0	7.6	7.6	5.5	887.7	887.7	887.4	0.5	108.0	2.0	0	0.0
2	8	200	18.7	21.4	18.3	46.0	49.0	38.0	6.8	7.7	6.3	887.8	887.9	887.7	1.6	115.0	4.2	0	0.0
3	8	300	20.1	20.4	17.5	43.0	55.0	43.0	7.3	8.7	7.2	887.6	887.8	887.6	1.8	122.0	3.0	0	0.0
4	8	400	19.7	20.5	19.1	41.0	43.0	39.0	6.2	7.3	5.8	887.5	887.7	887.4	1.6	98.0	2.8	0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Fig. 3: Dataset after preprocessing.

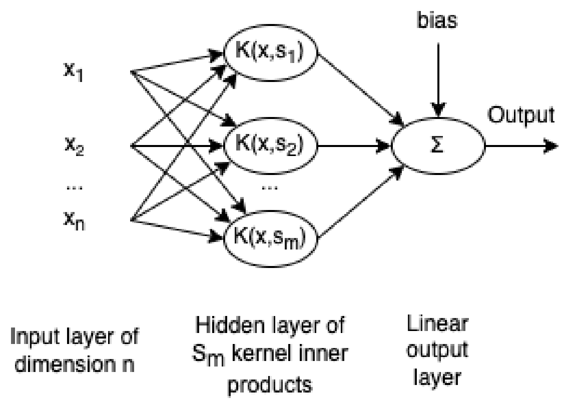


Fig. 4: Architecture of the SVM classifier. Source: author's own.

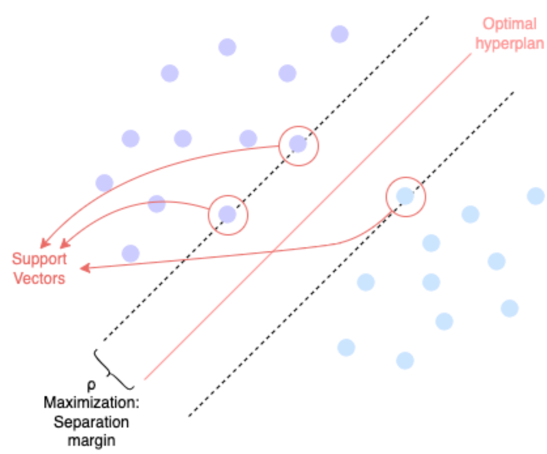


Fig. 5: Maximum margin hyperplane. Source: author's own.

2) Logistic Regression (LR): Figure 6 illustrates the diagram representation of the logistic regression model. Eq. (3) represents the model, in which y is the dependent variable, X is the independent variable, β_0 and β_1 are the two unknown constants that represent the intercept and slope, respectively. ε is the error term.

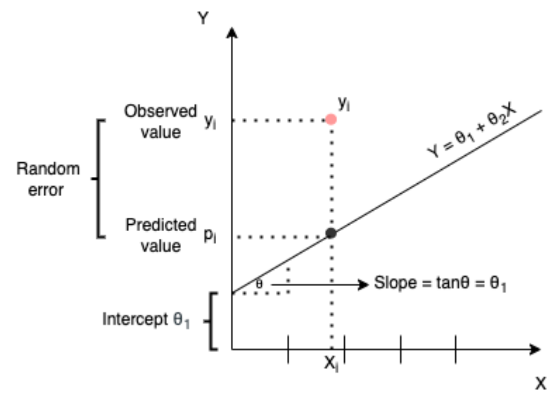


Fig. 6: Logistic Regression model. Source: author's own.

$$y = \beta_0 + \beta_1 X + \varepsilon \tag{3}$$

The primary goal while using logistic regression is to locate the best-fit line, which implies that the error between the predicted vs actual values should be kept to a minimum. Eq. (4) calculates a prediction of y on the basis of $X = x$. θ_1 and θ_2 are the estimates to $\hat{\beta}_0$ and $\hat{\beta}_1$, respectively.

$$\hat{y} = \theta_1 + \theta_2 x \tag{4}$$

3) Random Forest (RF): Figure 7 shows the diagram representation of Random Forest (RF). This algorithm combines the output of multiple decision trees to reach a single prediction result. The algorithm behaves randomly to choose subsets and form decision trees to, finally, take an average of these and predict the task in question. Unlike Artificial Neural Network (ANN) and SVM, the training procedure for random forest algorithm is simple and presents high prediction accuracy [23]. On the other hand, this algorithm can be very time-consuming as it is computing data for each individual decision tree. Morever, RF requires more resources to store data since they are dealing with large datasets.

4) Multilayer Perceptron (MLP): Let $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ be a multidimensional data instance, in which x_i is the value of an attribute. Multilayer Perceptron (MLP) is composed by

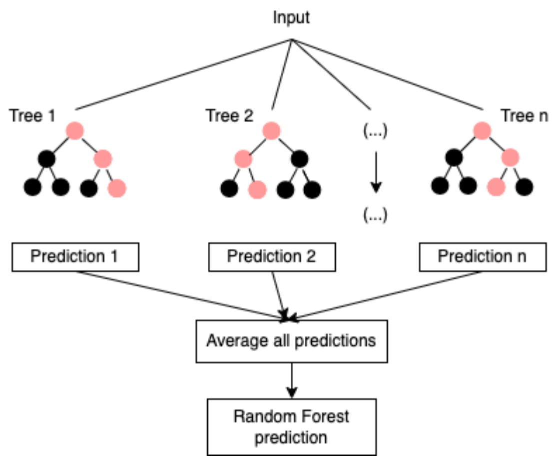


Fig. 7: Random Forest diagram. Source: author's own.

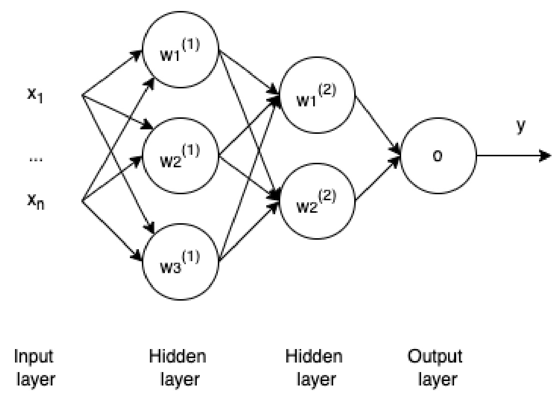


Fig. 8: MLP architecture. Source: author's own.

several units called Perceptron neurons, which is a mathematical model described by Eq. (5):

$$y = \varphi\left(\sum_{i=1}^{n} x_i \cdot w_i + b\right), \tag{5}$$

where w_i is the weight, b is the bias and φ is the activation function. The weights associated with these neurons are the parameters, which are adjusted during training. y is the output value of the perceptron and its value depends on the activation function, in which sigmoid and rectified linear unit (relu) are traditional choices. In supervised machine learning, it is required to compare y with the expected output $\hat{y}$ (also known as label) for a given input instance $\mathbf{x}$.

The diagram of a MLP architecture is represented by Figure 8. Generally, MLP contains an input layer, which represents the dimensions of the input dataset. The architecture in the example shows the hidden layers 1 and 2, which includes the neurons and the output layer, which has a number of neurons that represents the dimension of the output data. It can be seen that each neuron of a specific layer is connected to the neurons of the subsequent and previous layers.

The MLP training is part of the context of supervised machine learning, in which each data sample has a classification label associated that fits. This occurs due to the algorithm of backpropagation. The idea of the backpropagation algorithm is to recalculate the value of the weights of the vector w of the last layer of neurons and thus proceed to the previous layers, backwards.

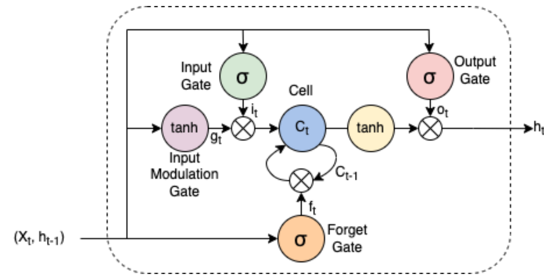


Fig. 9: Long Short-Term Memory (LSTM) model structure diagram. Source: author's own.

D. Regression models

1) Long Short-Term Memory (LSTM): The LSTM-based model is shown in Figure 10. The LSTM is a special cyclic neural network proposed by Hochreiter and Schmidhuber [24]. This type of Recurrent Neural Network (RNN) differs from others because of its 'memory blocks', enabling the LSTM network to classify, process, and forecast time series with time intervals of arbitrary length. The structure of the LSTM network is the same as traditional cyclic neural network, as it is composed of an input layer, a hidden layer and an output layer.

The LSTM cell structure is shown in Figure 9. It is composed by one loop connected unit and three gate structures: Input gate, output gate and forget gate. The formulas of the forget gate, input gate, output gate, input modulation gate, cell memory state and hidden layer output of the memory cell module will be shown in equations (6)–(11), respectively. This chain structure cell can retain information and uses the memory from previously measurements to generate the next measurement, which are controlled by the aforementioned gates.

$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1} + b_f) \tag{6}$$

$$i_t = \sigma(W_x^i x_t + W_h^i h_{t-1} + b_i) \tag{7}$$

$$o_t = \sigma(W_x^o x_t + W_h^o h_{t-1} + b_o) \tag{8}$$

$$g_t = tanh(W_x^g x_t + W_h^g h_{t-1} + b_g) \tag{9}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{10}$$

$$h_t = o_t \circ tanh(c_t) \tag{11}$$

In LSTM, the backpropagation mentioned before on the MLP model III-C4 also occurs, the difference is that it occurs through time, as we are dealing with a sequence data like a time series.

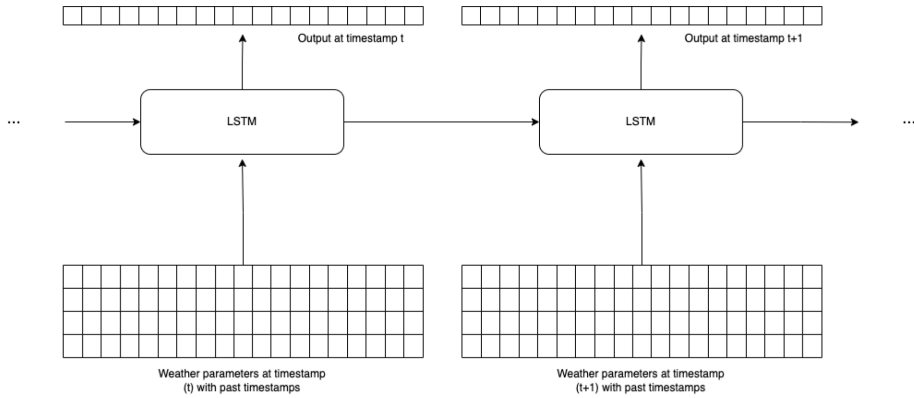


Fig. 10: LSTM for Time Series. Source: author's own.

2) Bidirectional Long Short-Term Memory (BiLSTM): Figure 11 illustrates the BiLSTM architecture. It differs from LSTM as it adds one additional layer of LSTM and the input sequence flows backward on it. For instance, if we consider 5 previous measurements, the LSTM will then consider the last 5 measurements and moves forward through time, beginning from the start of the sequence. Each measurement consists of 19 attributes, which are the columns of the tabular dataset. The BiLSTM model follows that same path, however it also moves backwards through time, beginning from the end of the sequence.

3) Transformer for Time-Series (T-TS): Figure 12 illustrates the Transformer architecture [25], which is basically an encoder-decoder architecture. The encoder block includes a multi-head attention layer and a feed forward neural network. Although the decoder block have the same two layers as the encoder block, it has one additional layer, the masked multi-head attention. The decoder process is a sequential process in load prediction, which means that when decoding the feature vector in position t it should only read the positions before t

$$(t-1, t-2, \ldots, 1)$$

. This particularity is solved by adding this additional layer mentioned before, proposed by Mnih et al [26].

IV. Experimental results

This section presents the results for both rain prediction IV-A and weather forecasting IV-B tasks. As for the first, the evaluation parameters used to compare different types of machine learning models were precision, recall and f1-score, as they are used for categorical attributes since we are dealing with a classification task.

The proposed methodology was coded in Python 3.10, in which the data preprocessing was performed using Pandas[2] and Numpy[3]. The classification models SVM and Random Forest were trained using the respective classes on Scikit-Learn[4]. Multilayer Perceptron, Logistic Regression, LSTM, BiLSTM and Transformers were coded using Keras/TensorFlow[5].

[2]https://pandas.pydata.org/
[3]https://numpy.org/
[4]https://scikit-learn.org/stable/
[5]https://keras.io/

Equation (12) shows how to calculate the precision parameter. After correcting the proposed model, we then compare the predicted output with test data to evaluate the model's performance. A confusion matrix is drawn between actual and predicted data and is used to evaluate the performance of the trained model. Therefore, the precision is calculated dividing the true positives (TP) by the sum of TP and false positives (FP), giving us the real positives from all data classified as positives.

$$Precision = \frac{TP}{TP + FP}. \tag{12}$$

Equation (13) gives the percentage of true positives taking into account the real positives - The sum of true positives (TP) and false negatives (FN).

$$Recall = \frac{TP}{TP + FN}. \tag{13}$$

Eq. (14) shows the final f1-score, which is a metric to measure the general quality of our model.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{14}$$

As for the Weather Time-series Forecasting, the Root Mean Squared Error (RMSE), shown in equation (15) was used to reflect the overall error of the estimated results, where y_i is the i_{th} expected value in the dataset and $\hat{y}_i$ is the i_{th} predicted value. Unlike Mean Squared Error (MSE), RMSE provides a measure in the same units as the target variable.

$$RMSE = \frac{\sum_{j=1}^{n} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_{ij} - \hat{y}_{ij})^2}}{N}. \tag{15}$$

The Mean Squared Error (MSE) was also used to evaluate the models, which is calculated as shown in equation 16.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2. \tag{16}$$

Another metric considered in this study was the standard deviation. Equation (17) shows the formula used to calculate it, where x_i represents each value of the dataset, $\overline{x}$ the mean of all values and N the number of values in the dataset. This

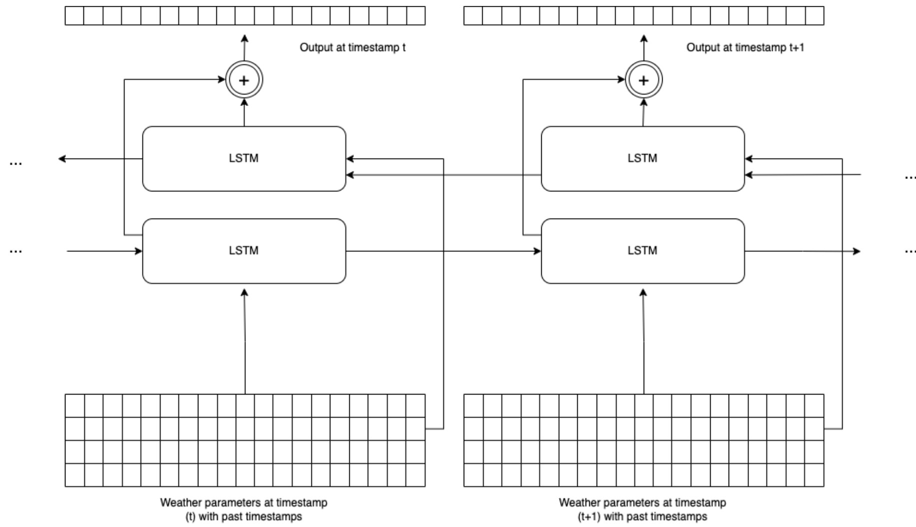


Fig. 11: BiLSTM for time series. Source: author's own.

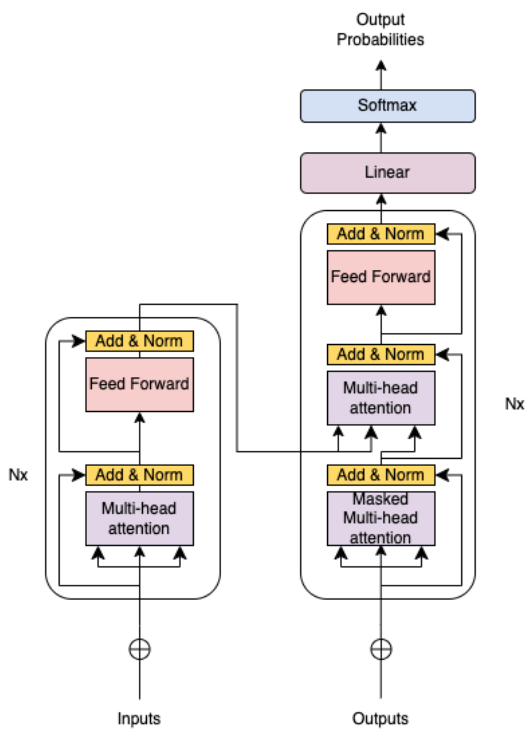


Fig. 12: Transformer for Time Series [25].

statistical measurement can facilitate the understanding of how dispersed the data is in relation to the mean, which will be discussed later in subsection IV-B.

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n}(x_{ij} - \overline{x})^2}{N}}. \tag{17}$$

A. Rain Prediction Model

For this task, we consider the 18 atmospheric attributes as the feature space for training the classification models. The attribute "Rain" is considered the class label for the prediction task. We converted this attribute to a categorical binary indicating the registration of rain or not.

TABLE II: Optimal hyperparameter values for classification models.

Classification model	Optimal hyperparameter values
SVM	Kernel = linear, C = 20
MLP	Three hidden layers Respective number of neurons on each: 128, 16 and 32, learning rate = 10^{-4}
Random Forest	max depth = 3
Logistic Regression	learning rate = 10^{-4}

In order to obtain the more appropriate classifiers for each classification model, we performed a hyperparameter optimization on the validation set. Table II shows the optimal hyperparameter values for each classification model.

Table III shows the results for the rain prediction task, which considered precision, recall and f1-score as evaluation metrics. The models taking into consideration were Logistic Regression, Random Forest, SVM and MLP. The SVM presented better results, with a precision of 0.95 and f1-score of 0.28. Figure 13 shows the confusion matrices for SVM, Random Forest, Logistic Regression and MLP, respectively. It is possible to observe that the SVM presented better results, as there were 4244 correct hits of the total of 4245 non-rainy instances and 21 correct predictions among the total of 129 rainy instances.

B. Weather time series forecasting

Table V shows the global RMSE and MSE for the weather time series forecasting. The deep learning methods compared were LSTM, BiLSTM and Transformers. The input variable past was varied from values 5, 20 and 50. The BiLSTM model with past 50 presented better results for RMSE when compared to others, with a train score of 115.310 and train score of 0.0139 for MSE.

Figure 14 shows the comparison between actual and predicted values when considering BiLSTM with past 50. This

TABLE III: F1-Score for rain prediction by considering the Brasilia INMET weather station.

Model	Precision	Recall	F1-Score
Logistic Regression	0.00	0.00	0.00
Random Forest	0.00	0.00	0.00
SVM	0.95	0.16	0.28
Multilayer Perceptron	0.00	0.00	0.00

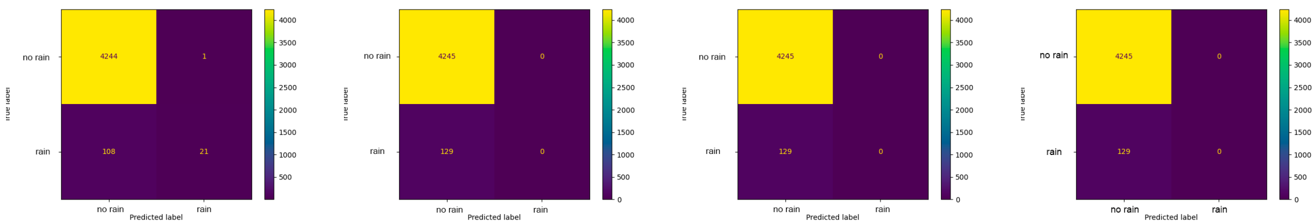


Fig. 13: Confusion Matrices for SVM, Random Forest, Logistic Regression and MLP, respectively. Source: author's own.

TABLE IV: Optimal hyperparameter values for regression models.

Regression Model	Past	Optimal hyperparameter values
LSTM	5	The optimal number of units in the outter LSTM layer is 512, The optimal number of prob in the Droput layer is 0.3, learning rate = 5e-05
LSTM	20	The optimal number of units in the outter LSTM layer is 512, The optimal number of prob in the Droput layer is 0.3, learning rate = 5e-05.
LSTM	50	The optimal number of units in the outter LSTM layer is 512, The optimal number of prob in the Droput layer is 0.3, learning rate = 5e-05.
BiLSTM	5	The optimal number of units in the outter BiLSTM layer is 576, The optimal number of prob in the Droput layer is 0.3, learning rate = 0.0005.
BiLSTM	20	The optimal number of units in the outter BiLSTM layer is 512, The optimal number of prob in the Droput layer is 0.3, learning rate = 5e-05.
BiLSTM	50	The optimal number of units in the outter BiLSTM layer is 512, The optimal number of prob in the Droput layer is 0.3, learning rate = 5e-05.
Transformer	5	learning rate = 0.001
Transformer	20	learning rate = 3e-05
Transformers	50	learning rate = 0.0005

model presented better values for RMSE, thus the graphics show that the BiLSTM predicted values are similar to the actual values at some points. On the other hand, some attributes are harder to predict as they present higher values for standard deviation, which is the case for "Wind Direction" that presented a standard deviation of 56.0059 in the prediction model, compared to a value of 86.0656 for actual model, as shown in table VII.

V. Discussion

Regarding the regression task, the proposed BiLSTM model with past 50 performed better than the LSTM model. It is know that the BiLSTM works with more hyperparameters than LSTM, besides making time-series analysis forward-through and backwards through time. The BiLSTM can also capture more long duration patterns when compared to other models in order to make better predictions. This can be observed by looking at the RMSE of test data for different values of hyper-parameter past.

As shown in tables VI and VII, the predicted model was able to replicate the pattern of standard deviation values when compared to the actual values. The values were calculated based on the BiLSTM model with past 50. Attributes like Temperature, Dew Point and Rain presented similar values of standard deviation when compared to the actual model. For other attributes like Wind Direction, the standard deviation is considerably high and this could lead to a higher error rate, thus the difference from the predicted to actual model standard deviation is 30.0597, considering the non-normalized values.

As for the Transformers model, the values for RMSE were quite high when compared to other models. Zeng et al [27] questions the effectiveness of Transformer-based solutions for the long- term time-series forecasting (LTSF) problem, demonstrating that linear models outperformed LTSF-Transformer mainly caused by the overfit toward sudden change noises in the training data, resulting in significant accuracy degradation.

In the classification task, the presence of missing values possibly affected the models' performances. Although we filled these missing values as zero-valued fields, the literature

TABLE V: Global RMSE and MSE for optimal models by considering the Brasilia INMET weather station.

Model	Past	Test Score Global RMSE	Test Score Global MSE
LSTM	5	134.309	0.0286
LSTM	20	166.974	0.0162
LSTM	50	120.850	0.0156
BiLSTM	5	123.505	0.0132
BiLSTM	20	214.732	0.0183
BiLSTM	50	115.310	0.0139
Transformer	5	262.542	0.0429
Transformer	20	376.079	0.0395
Transformer	50	250.768	0.0402

TABLE VI: Standard deviation of considered attributes for actual and predicted model for normalized values.

Attribute	standard deviation of actual model	standard deviation of predicted model
Inst. Temperature	0.1829	0.1296
Max. Temperature	0.1902	0.1420
Min. Temperature	0.1840	0.1392
Inst. Humidity	0.2354	0.2063
Max. Humidity	0.2340	0.2120
Min. Humidity	0.2362	0.2209
Inst. Dew Point	0.1698	0.1622
Max. Dew Point	0.1734	0.1801
Min. Dew Point	0.1770	0.1687
Inst. Pressure	0.1504	0.1432
Max. Pressure	0.1519	0.0576
Min. Pressure	0.1511	0.0524
Wind Speed	0.1328	0.1022
Wind Direction	0.2397	0.1560
Wind Raj.	0.1299	0.0965
Radiation	0.2614	0.2264
Rain	0.0436	0.0

TABLE VII: Standard deviation of considered attributes for actual and predicted model for non-normalized values.

Attribute	standard deviation of actual model	standard deviation of predicted model
Inst. Temperature	3.8046	2.6974
Max. Temperature	3.8421	2.8697
Min. Temperature	3.7353	2.8271
Inst. Humidity	18.3615	16.0970
Max. Humidity	17.7845	16.1164
Min. Humidity	18.6624	17.4526
Inst. Dew Point	3.6524	3.4875
Max. Dew Point	3.6240	3.7659
Min. Dew Point	3.6825	3.5110
Inst. Pressure	2.3020	2.1911
Max. Pressure	2.2792	0.8641
Min. Pressure	2.3120	0.8029
Wind Speed	1.0094	0.7774
Wind Direction	86.0656	56.0059
Wind Raj.	2.1059	1.5649
Radiation	1067.710	924.9722
Rain	1.0211	0.0

has reported strategies for imputation that could be considered. Alternatively, the instances with missing values can also be removed, though there would be a gap between measurement that can be challenging to the classification models.

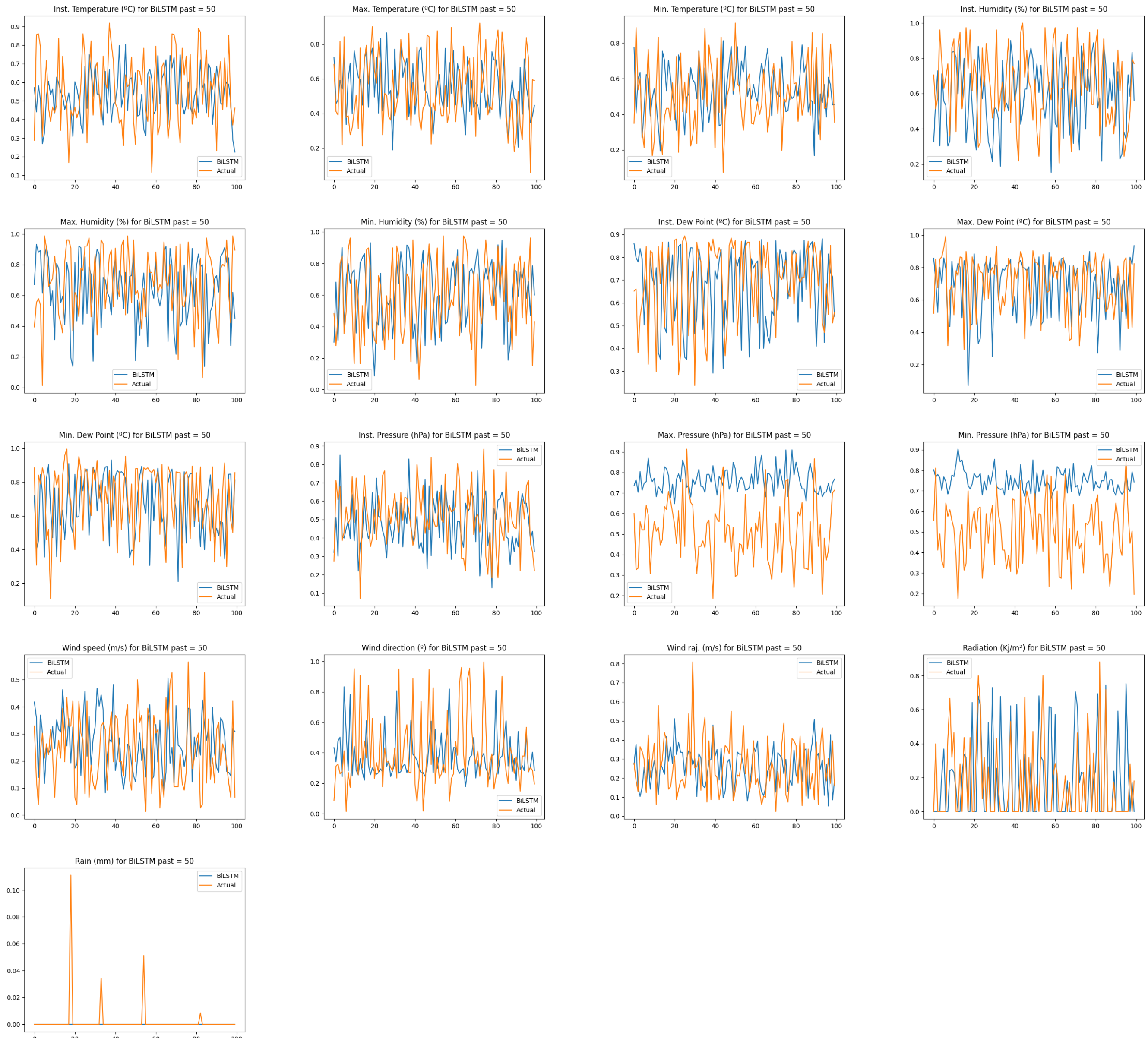


Fig. 14: Comparison of actual and predicted values using BiLSTM with past = 50.

Another factor is the considerable class imbalance on weather data, since non-rain are predominant in the measurements. This led the models to learn better the patterns of non-rainy weather conditions. Data under-sampling and oversampling strategies could be considered for generating a relative balanced dataset. However, this procedure is challenging to be performed in time series as well as it can affect the original statistical distribution of the dataset.

Finally, the classification models presented limited ability to handle temporal data, since an atmospheric condition is dependent on the previous conditions. Thus, the predictions would be more accurate if the models could take into account those past measurements. Therefore, the rain prediction could be derived from the regression task by analyzing the attribute “Rain”.

VI. Conclusion

In this paper, we explored and compared different machine learning approaches for short-term weather forecasting, as it is a relevant task for the human activities. We collected data made available by INMET from a weather station located in Brasília in period of 2 years. The experiment was divided in two main tasks — Rain prediction and weather time-series forecasting, which represent classification and regression tasks, respectively.

Regarding rain forecasting, we modeled a binary classification task aiming at predicting if there is rain event or not by considering atmospheric attributes. For this task, traditional classification models were employed: SVM, Random Forest, Logistic Regression and Feedforward Multilayer Perceptron. The experimental results did not show satisfactory results, in which SVM achieved the best F1-Score. Those poor performances among all classification models can be attributed to the considerable class imbalance in the considered weather station and the presence of missing values.

We also explored the use of deep learning models as a regression task for time series forecasting, so that all the atmospheric parameters can be predicted by also taking into account past weather measurements. We considered the state-of-the art models LSTM, BiLSTM and Transformers as well as we varied the number of past measurements when training the models. The experiments showed that the BiLSTM performing predictions based on the past 50 measurements yielded the best results when considering the RMSE and MSE as the evaluation metrics. However, this method is not efficient when compared to the other deep learning models due to high time-consumption. The experiments showed that machine learning techniques can be used for weather forecasting.

For future work, we plan to consider strategies for the imputation of missing values, which is a common situation in the data collected from the weather stations. Furthermore, weather stations from other cities can also be taken into account since the performance of models are related to the local climate characteristics.

Acknowledgment

The authors would like to thank *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq) and ProIC/UnB for the grant that supported this research, processes numbers 144451/2021-2 and 135189/2022-5.

References

[1] J. W. Jones, J. W. Hansen, F. S. Royce, and C. D. Messina, "Potential benefits of climate forecasting to agriculture," *Agriculture, Ecosystems & Environment*, vol. 82, no. 1-3, pp. 169–184, 2000.

[2] K. E. Ukhurebor, C. O. Adetunji, O. T. Olugbemi, W. Nwankwo, A. S. Olayinka, C. Umezuruike, and D. I. Hefft, "Precision agriculture: Weather forecasting for future farming," in *AI, Edge and IoT-based Smart Agriculture*. Elsevier, 2022, pp. 101–121.

[3] I. Gultepe, R. Sharman, P. D. Williams, B. Zhou, G. Ellrod, P. Minnis, S. Trier, S. Griffin, S. S. Yum, B. Gharabaghi *et al.*, "A review of high impact weather for aviation meteorology," *Pure and Applied Geophysics*, vol. 176, pp. 1869–1921, 2019.

[4] J. A. Patz, D. Engelberg, and J. Last, "The effects of changing weather on public health," *Annual Review of Public Health*, vol. 21, no. 1, pp. 271–307, 2000.

[5] H. Wang, Y. Hu, Y. Guo, Z. Wu, and D. Yan, "Urban flood forecasting based on the coupling of numerical weather model and stormwater model: A case study of zhengzhou city," *Journal of Hydrology: Regional Studies*, vol. 39, p. 100985, 2022.

[6] B. E. Potter and D. McEvoy, "Weather factors associated with extremely large fires and fire growth days," *Earth Interactions*, vol. 25, no. 1, pp. 160–176, 2021.

[7] K. A. Browning, "Review lecture: Local weather forecasting," *Proceedings of the royal society of London. A. Mathematical and Physical Sciences*, vol. 371, no. 1745, pp. 179–211, 1980.

[8] D. S. R. Junior, S. S. Sombra, A. A. Costa, O. Neto, A. C. S. dos Santos, and M. B. M. Firmino, "VerificaÇÃo da previsÃo operacional de chuva sobre bacias hidrogrÁficas do estado do cearÁ."

[9] V. Krishnamurthy, "Predictability of weather and climate," *Earth and Space Science*, vol. 6, no. 7, pp. 1043–1056, 2019.

[10] F. Yang, X. Li, M. Wang, H. Zang, W. Pang, and M. Wang, "Waveform: graph enhanced wavelet learning for long sequence forecasting of multivariate time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 10 754–10 761.

[11] E. Fu, Y. Zhang, F. Yang, and S. Wang, "Temporal self-attention-based conv-lstm network for multivariate time series prediction," *Neurocomputing*, vol. 501, pp. 162–173, 2022.

[12] S. Bhanja and A. Das, "Deep neural network for multivariate time-series forecasting," in *Proceedings of International Conference on Frontiers in Computing and Systems: COMSYS 2020*. Springer, 2021, pp. 267–277.

[13] C.-J. Zhang, J. Zeng, H.-Y. Wang, L.-M. Ma, and H. Chu, "Correction model for rainfall forecasts using the lstm with multiple meteorological factors," *Meteorological Applications*, vol. 27, no. 1, p. e1852, 2020. [Online]. Available: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/met.1852

[14] C.-J. Zhang, H.-Y. Wang, J. Zeng, L.-M. Ma, and L. Guan, "Tiny-rainnet: a deep convolutional neural network with bi-directional long short-term memory model for short-term rainfall prediction," *Meteorological Applications*, vol. 27, no. 5, p. e1956, 2020. [Online]. Available: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/met.1956

[15] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single layer and multi-layer long short-term memory (lstm) model with intermediate variables for weather forecasting," *Procedia Computer Science*, vol. 135, pp. 89–98, 2018, the 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705091831439X

[16] H. Chaudhary, U. Mishra, A. Gupta, and A. Singh, "Comparative analysis of rainfall prediction using machine learning and deep learning techniques," in *2022 3rd International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*. IEEE, 2022, pp. 1–6.

[17] C.-J. Zhang, J. Zeng, H.-Y. Wang, L.-M. Ma, and H. Chu, "Correction model for rainfall forecasts using the lstm with multiple meteorological factors," *Meteorological Applications*, vol. 27, no. 1, p. e1852, 2020. [Online]. Available: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/met.1852

[18] J. A. Weyn, D. R. Durran, and R. Caruana, "Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data," *Journal of Advances in Modeling Earth Systems*, vol. 11, no. 8, pp. 2680–2693, 2019. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001705

[19] N. Srinu and B. H. Bindu, "A review on machine learning and deep learning based rainfall prediction methods," in *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*. IEEE, 2022, pp. 1–4.

[20] S. Manandhar, S. Dev, Y. H. Lee, Y. S. Meng, and S. Winkler, "A data-driven approach to detect precipitation from meteorological sensor data," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 3872–3875.

[21] Z. Wang, T. Hong, and M. A. Piette, "Building thermal load prediction through shallow machine learning and deep learning," *Applied Energy*, vol. 263, p. 114683, 2020.

[22] S. Xu, Y. Song, and X. Hao, "A comparative study of shallow machine learning models and deep learning models for landslide susceptibility assessment based on imbalanced data," *Forests*, vol. 13, no. 11, p. 1908, 2022.

[23] R. Meenal, P. Angel, D. Pamela, and E. Rajasekaran, "Weather prediction using random forest machine learning model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, p. 1208, 05 2021.

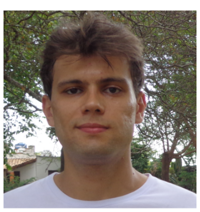
[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] 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.

[26] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/09c6c3783b4a70054da74f2538ed47c6-Paper.pdf

[27] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" 2022.

Carolina E. Machado Undergraduate of Computer Science at University of Brasília. Her main research interests comprise machine learning, cloud classification and software engineering.



Vinicius R. P. Borges Assistant professor at the Department of Computer Science at University of Brasília. Obtained his B.Sc. (2008) and M.Sc. (2011) degrees in computer science from the Federal University of Uberlândia. Received his Ph.D. (2016) in computer science at the Instituto de Ciências Matemáticas e de Computação at University of São Paulo. His main research interests comprise machine learning, pattern recognition, information visualization and visual analytics.