SENAI CIMATEC UNIVERSITY CENTER

PROGRAM IN COMPUTATIONAL MODELING AND INDUSTRIAL TECHNOLOGY AT THE GRADUATE LEVEL

Master's in Computational Modelling and Industrial Technology

Master's Dissertation

Short range lightning forecast based on deep learning models using globally available gridded meteorological data

Presented by: Mirella Lima Saraiva Araujo
Supervisor: Erick Giovani Sperandio Nascimento
Co-supervisor: Diogo Nunes da Silva Ramos

October 2023

Mirella Lima Saraiva Araujo

Short range lightning forecast based on deep learning models using globally available gridded meteorological data

This Master's Dissertation was presented to Program in Computational Modeling and Industrial Technology at the Graduate Level, the Course of Master's in Computational Modelling and Industrial Technology of the SENAI CIMATEC University Center, as a partial requirement for the degree of **Master in Computational Modelling and Industrial Technology**.

Supervisor: Erick Giovani Sperandio Nascimento

Co-supervisor: Diogo Nunes da Silva Ramos

Salvador
2023

Ficha catalográfica elaborada pela Biblioteca do Centro Universitário SENAI CIMATEC

A658s Araujo, Mirella Lima Saraiva

Short range lightning forecast based on deep learning models using globally available gridded meteorological data / Mirella Lima Saraiva Araujo. – Salvador, 2023.

86 f. : il. color.

Orientador: Prof. Dr. Erick Giovani Sperandio Nascimento.
Coorientador: Prof. Dr. Diogo Nunes da Silva Ramos

Dissertação (Mestrado em Modelagem Computacional e Tecnologia Industrial) – Programa de Pós-Graduação, Centro Universitário SENAI CIMATEC, Salvador, 2023.
Inclui referências.

1. Deep learning. 2. Previsão de raios. 3. Redes neurais artificiais. 4. Machine learning. I. Centro Universitário SENAI CIMATEC. II. Nascimento, Erick Giovani Sperandio. III. Ramos, Diogo Nunes da Silva. IV. Título.

CDD 693.898

NDI - 03

SENAI CIMATEC University Center

Program in Computational Modeling and Industrial Technology at the Graduate Level

Master's in Computational Modelling and Industrial Technology

The Examination Committee, composed of the professors listed below, have read and recommend the approval with distinction of the Master's Dissertation, entitled "Short range lightning forecast based on deep learning models using globally available gridded meteorological data", presented on 30 of October, 2023, as a partial requirement for the degree of **Master in Computational Modelling and Industrial Technology**.

Supervisor :

Electronically signed by:
Erick Giovani Sperandio Nascimento
CPF: ***.666.177-**
Date: 11/14/2023 7:37:00 AM +00:00

Prof. Dr. Erick Giovani Sperandio Nascimento

SENAI CIMATEC University Center

Co-supervisor :

Assinado eletronicamente por:
Diogo Nunes da Silva Ramos
CPF: ***.300.584-**
Data: 14/11/2023 10:58:50 -03:00

Prof. Dr. Diogo Nunes da Silva Ramos

SENAI CIMATEC University Center

Internal Committee Member:

Assinado eletronicamente por:
MARCELO Albano MORET Simões Gonçalves
CPF: ***.131.127-**
Data: 20/11/2023 06:44:14 -03:00

Prof. Dr. Marcelo Albano Moret Simões Gonçcalve

SENAI CIMATEC University Center

External Committee Member:

Assinado eletronicamente por:
Diego Gervasio Frías Suárez
CPF: ***.047.757-**
Data: 21/11/2023 21:01:37 -03:00

Prof. Dr. Diego Gervasio Frías Suárez

UNEB - Universidade do Estado da Bahia

Acknowledgement

I would like to begin by expressing my deep gratitude to my supervisor, Prof. Dr. Erick Giovani Sperandio Nascimento, who placed trust in my work and opened the doors for this Master's degree, and for that, I am forever grateful. His expertise, understanding, and patience greatly enriched this study.

In addition, I am profoundly thankful to my co-supervisor, Professor Diogo Nunes, for his insightful guidance and constructive criticism during the creation of this dissertation.

My heartfelt appreciation extends to my family, who have consistently demonstrated unwavering faith in my capabilities and never ceased to provide emotional and moral support during challenging times.

Moreover, I am deeply indebted to my loving boyfriend, who offered a comforting sanctuary from my studies and a reminder that there exists a world beyond academia. Your support and encouragement have been pivotal in accomplishing this journey.

Lastly, I wish to express my gratitude to the Support Team at the Centro de Super Computação of Senai CIMATEC Salvador, who was available 24 hours a day, 7 days a week, with the same energy and attention, to assist in resolving any connection and stability issues with the HPC. This was utilized to run the models developed and examined in this work. Thank you all for your commitment and expertise.

Salvador, Brazil
dia de October 2023

Mirella Lima Saraiva Araujo

Resumo

No contexto das mudanças climáticas e seus efeitos nos eventos climáticos extremos, a previsão e monitoramento de raios é crucial, especialmente em regiões propensas a esses eventos. Neste contexto, este trabalho visa desenvolver e avaliar modelos de aprendizado profundo para prever raios com até 6 horas de antecedência, utilizando como entrada dados meteorológicos globalmente disponíveis do Global Data Assimilation System (GDAS), e como alvo dados de ocorrências de raios fornecidos pelo grupo de eletricidade atmosférica (ELAT) do Instituto Nacional de Pesquisas Espaciais (INPE). Como estudo de caso, foi selecionada a região que compreende o Estado da Bahia, Brasil, por possuir uma vasta área onde muitos empreendimentos elétricos estão sendo desenvolvidos, o que justifica o estudo e aplicação dessa pesquisa para essa região. Vários modelos de aprendizagem profunda foram testados a fim de identificar um modelo preciso e de alto desempenho para a previsão antecipada de ocorrências de raios. No total, foram considerados seis modelos de aprendizado profundo, incluindo redes neurais de perceptron multicamadas (MLP), redes neurais convolucionais (CNN), redes neurais recorrentes (RNN) e redes híbridas. Esses modelos foram otimizados para obter a máxima precisão na previsão. Cada modelo foi minuciosamente treinado e avaliado e seu desempenho foi avaliado usando métricas tradicionais para classificação, sendo elas: *F1-score*, *Area Under the Receiver Operating Characteristic Curve* (ROC AUC), *Area Under the Precision-Recall Curve* (PRC AUC), *precision*, *recall*, *specificity* e *accuracy*. O teste de Delong foi usado para avaliar a significância estatística e detectar diferenças entre os modelos. O estudo identificou com sucesso modelos que tiveram um desempenho superior na previsão de casos em que ocorreram e que também não ocorreram raios, e esses modelos foram considerados significativamente distintos entre si. Especificamente, o modelo CNN-GRU demonstrou superioridade em relação aos demais, com boa eficiência computacional. Ademais, foram identificadas potenciais limitações resultando em recomendações para trabalhos futuros na previsão de raios usando aprendizado profundo. Com base nos resultados, é possível verificar que técnicas avançadas de aprendizado profundo podem contribuir significativamente para previsões de ocorrências de raios com precisão e bom desempenho computacional. Ao identificar os modelos mais adequados para prever raios, esta pesquisa serve como uma base para o desenvolvimento de sistemas de alerta mais eficazes, contribuindo para a segurança da sociedade e da indústria, e para a resiliência climática.

Palavras-chave: Deep Learning, Previsão de raios, Redes Neurais Artificiais, Machine Learning

Abstract

In the context of climate change and its effects on extreme weather events, the prediction and monitoring of lightning strikes are crucial, especially in regions prone to such events. In this context, this study aims to develop and evaluate deep learning models to predict lightning strikes up to 6 hours in advance, using globally available meteorological data from the Global Data Assimilation System (GDAS) as input, and lightning occurrence data provided by the Atmospheric Electricity Group (ELAT) of the National Institute for Space Research (INPE) as the target. The state of Bahia, Brazil, was selected as a case study due to its vast region where many electrical projects are being developed, justifying the study and application of this research to that region. Several deep learning models were tested to identify an accurate and high-performing model for early prediction of lightning occurrences. In total, six deep learning models were considered, including Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and hybrid networks. These models were optimized to achieve maximum prediction accuracy. Each model was meticulously trained and evaluated, and its performance was assessed using traditional classification metrics, including F1-score, Area Under the Receiver Operating Characteristic Curve (ROC AUC), Area Under the Precision-Recall Curve (PRC AUC), precision, recall, specificity, and accuracy. The Delong test was used to evaluate statistical significance and detect differences between the models. The study successfully identified models that outperformed others in predicting both lightning and no lightning cases, and these models were found to be significantly distinct from each other. Specifically, the CNN-GRU model demonstrated superiority compared to the others, with good computational efficiency. Additionally, potential limitations were identified, resulting in recommendations for future work in lightning prediction using deep learning. Based on the results, it can be concluded that advanced deep learning techniques can significantly contribute to accurate lightning occurrence predictions with good computational performance. By identifying the most suitable models for lightning prediction, this research serves as a foundation for the development of more effective alert systems, contributing to the safety of society and industry, as well as climate resilience.

Keywords: Deep Learning, Lightning forecast, Artificial Neural Networks, Machine learning

Contents

List of Tables

List of Figures

List of Acronyms

ADSNet	Attention-based Dual-Source Spatiotemporal Neural Network
AI	Artificial Inteligence
ANA	Agência Nacional de Água e Saneamento Básico
ANN	Artificial Neural Network
AR	AutoRegressive
ARIMA	AutoRegressive Integrated Moving Average
AUC	Area Under the Curve
BPTT	Backpropagation Through Time
BrasilDAT ..	Sistema Brasileiro de Detecção de Descargas atmosféricas
CAPE	Convective Available Potential Energy.
CEC	Constant Error Carousel
CNN	Convolutional Neural Networks
Conv1D	1D Convolutional Layer
ELAT	Grupo de Eletricidade Atmosférica
ENSO	El Niño Southern Oscillation
FAA	Federal Aviation Administration
FAR	False Alarm Ratio
FN	False Negatives
FP	False Positives
FPR	False Positive Rate
GBDT	Gradient Boost Decision Tree
GDAS	Global Data Assimilation System
GFS	Global Forecast System
GRU	Gated Recurrent Units
HPC	High-Performance Computing
IBGE	Instituto Brasileiro de Geografia e Estatística
INMET	Instituto Nacional de Meteorologia
INPE	Instituto Nacional de Pesquisas Espaciais
JLRL	Johannesburg Lightning Research Laboratory
LF	Low-frequency
LSTM	Long Short-Term Memory
LSTM-FC ...	Fully-Connected Network
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MI	Ministério da Integração Nacional
MMS	Meteorological Malaysian Services
MSE	Mean Squared Error
NetCDF	Network Common Data Form
NCEP	National Centers for Environmental Prediction
POD	Probability of Detection
PPGMCTI ..	Programa de Pós-Graduação em Modelagem Computacional
PRC	Precision-Recall Curve
ReLU	Rectified Linear Unit
RINDAT	Rede Integrada Nacional de Detecção de Descargas Atmosféricas
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks

ROC	Receiver Operating Characteristic
SALDN	South African Lightning Detection Network
SAWS	South African Weather Service
SELU	Scaled Exponential Linear Unit
SMOTE	Synthetic Minority Over-sampling Technique
SST	Sea Surface Temperature
StepDeep ...	Spatiotemporal Deep Neural Network
TanH	Hyperbolic Tangent
TLS	Thunderstorm Location System
TN	True Negatives
TNA	Tropical North Atlantic
TP	True Positives
TPR	True Positive Rate
TSA	Tropical South Atlantic
UNEB	Universidade do Estado da Bahia
VHF	Very High-frequency
WRF	Weather Research and Forecasting
WTI	West Texas Intermediate

List of Symbols

$\hat{\theta}$	Theta Hat - Empirical AUC
θ	Theta - Mean AUC
ψ	Psi
Φ	Phi - The standard normal cumulative distribution function
µs	Macroseconds

Chapter One

Introduction

Lightning is a natural electric discharge sparked by an imbalance of electrical charge, manifesting in various forms: between a cloud and the ground, within distinct regions of the same cloud, between separate clouds, or from a cloud to the surrounding air (The Royal Meteorological Society, 2017). This natural event has a highly destructive capability, which can cause negative economic and social impacts, interrupting electricity distribution, causing forest fires, disrupting transportation through airplanes and ships, damaging telecommunications systems, and causing injuries or even death of humans and animals.

According to the Federal Aviation Administration (FAA) part of the United States Department of Transportation, severe weather is the leading to the largest cause of air traffic delay in the National Airspace System, accounting for a total of 75.48%. Thunderstorms can block busy jet routes, causing flights to reroute into neighboring airspace, which can become overcrowded. (FAA, 2022)

As a case in point, on August 23, 2023, thunderstorm activity in Porto Alegre area led to suspended activity in the Salgado Filho airport for 4 hours. The suspension affected more than 30 flights with 12 aircraft having to wait on the runway to park after landing, 15 planes waited to take off, and nine flights were canceled.(FOSTER; GONÇALVES, 2023)

Therefore, forecast plays a crucial role in ensuring the effective management of traffic flow, allowing time for pre-departure planning and the filing of amended flight plans. FAA states that strategic traffic flow managers need to make predictions about convective weather impacts on airspace capacity 4-8 hours in advance for long-haul flights and 2-6 hours for shorter flights.(FAA, 2022)

The occurrences of thunderstorms have a greater concentration in tropical regions and on, or near, landmasses (KAPLAN; LAU, 2021), such as in Brazil, which is one of the countries most impacted by lightning as it is a vast tropical country.

The *Grupo de Eletricidade Atmosférica* (ELAT) at *Instituto Nacional de Pesquisas Espaciais* (INPE) estimates that Brazil is struck by lightning 77.8 million times per year, causing 110 deaths, making Brazil the seventh country with more deaths by lightning, causing approximately 1 billion dollars worth of economic losses (Instituto Nacional de Pesquisas Espaciais, 2007; PINTO JUNIOR; PINTO; REGINA, 2021). It is also worth noting that in Brazil the average number of death by lightning is about 70% higher than

in the United States. When looking at economic impacts, it is estimated that 70% of the total shutdown of transmission lines and 40% of the distribution networks in the country are due to lightning(PINTO JUNIOR; PINTO; REGINA, 2021). Therefore, an accurate lightning forecasting tool, that could anticipate these events with high accuracy within hours in advance, can help prevent or mitigate such impacts.

In recent years, the field of lightning forecasting has witnessed notable advancements as researchers strive to improve the accuracy and lead time of predictions. Existing methods often rely on localized weather data, which may not capture the full complexity of lightning occurrence patterns or numerical models that tend to have a high computational cost. These limitations necessitate the exploration of alternative approaches, such as leveraging deep learning techniques, to enhance lightning prediction capabilities.

In the unique context of Bahia, Brazil, accurate lightning forecasting plays a crucial role due to the region's distinct geographical and meteorological features. With its expansive coastal areas, diverse landscapes, and unique atmospheric conditions, Bahia experiences a notable frequency of lightning strikes, especially in the western region Abreu et al. (2020).To minimize the potential ramifications of these extreme meteorological phenomena, there's a critical need for developing robust and timely lightning forecasting models tailored specifically for this region. These models, given the diversity of Bahia's features, hold potential for effective application in regions exhibiting similar characteristics, thereby broadening their realm of impact.

Taking a look back to November 5, 2021, the severity of the issue becomes noticeable: Bahia recorded the highest number of lightning strikes across all Brazilian states, with a staggering total of 125,969 strikes. Valença bore the brunt of this weather anomaly, accounting for 6,269 strikes, followed closely by Maracãs with 3,354 strikes, according to TV Bahia (2021). This underlines the urgent necessity for tailored lightning forecast models in effectively safeguarding lives and infrastructure.

In this context, deep learning has emerged as a powerful tool for extracting complex patterns and relationships inherent in large-scale datasets. Its capacity to sift through large data quantities and identify sophisticated dependencies establishes it as an ideal candidate for handling meteorological data, thus, enhancing the precision of lightning predictions. This dissertation seeks to construct and assess various models expressly designed for lightning forecasting in Bahia. Through thorough evaluation, the objective of this study is to pinpoint the most effective deep learning structure that demonstrates superior performance and accuracy in predicting lightning occurrences for the Bahia region.

A key strength of this research is the use of globally accessible data sources while developing a model that can accurately forecast these extreme weather phenomena. This

strategy enhances the accessibility and usability of such forecasting tools, especially in areas with limited local data or remote sensing technologies, since it not only broadens the geographical applicability of the model but also successfully addresses the challenge posed by the scarcity of local data and remote sensing equipment. The ramifications of these findings may have a profound influence, potentially revolutionizing disaster preparedness, risk mitigation strategies, and public safety measures.

1.1 Overall and Specific Research Goals

The overall objective of this dissertation is to develop and evaluate different deep learning models with distinct architectures to forecast lightning occurrences in the Bahia region up to 6 hours in advance, utilizing globally available data. The aim is to identify and select the most suitable model that demonstrates higher performance and accuracy in lightning prediction, thereby contributing to the advancement of understanding and monitoring extreme weather events in this specific region.

1.2 Specific Goals

- Analyze globally accessible meteorological and lightning occurrence data, for the Bahia region as case study.
- Develop six different deep learning models with varying architectures, using techniques such as multilayer neural networks (MLPs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid networks, adjusting hyperparameters and optimizing the models to achieve the best prediction results.
- Determine the optimal deep learning model for predicting lightning occurrences in the Bahia region by validating models with appropriate datasets, evaluating performance metrics, and considering predictive accuracy, robustness, and computational efficiency.
- Perform an assessment of the limitations and potential sources of uncertainty of the selected models, providing recommendations and insights on possible improvements and future work related to lightning prediction using deep learning models.

1.3 Structure of the Master's Dissertation

This document presents 5 chapters organized in the following matter:

- **Chapter 1 - Introduction**: Presents the research context and outlines the objectives and structure of the dissertation, establishing a foundation for the subsequent chapters;
- **Chapter 2 - Background Theory and Fundamentals**: Explores the foundational concepts and theories of deep learning and lightning physics. This chapter aims to provide a comprehensive understanding of the theoretical underpinnings necessary for the research;
- **Chapter 3 - Methodology**: Presents the methodology employed in the research. This chapter details the approaches, techniques, and data used for developing and evaluating different deep learning models for lightning prediction in the Bahia region. It discusses the characteristics of the study area, data preprocessing, and the selection and design of the models;
- **Chapter 4 - Results and Discussion**: Reports the results obtained from the experiments conducted in the research. This chapter presents the findings of the developed deep learning models, and their performance metrics for the task of lightning predictions up to 6 hours in advance for the Bahia region. The results are analyzed and discussed in detail, considering their implications, limitations, and potential areas for improvement.;
- **Chapter 5 - Conclusions**: Summarizes the main findings and contributions of the research, providing a comprehensive conclusion to the dissertation. This chapter also offers suggestions for future research activities that can further advance the field.

Chapter Two

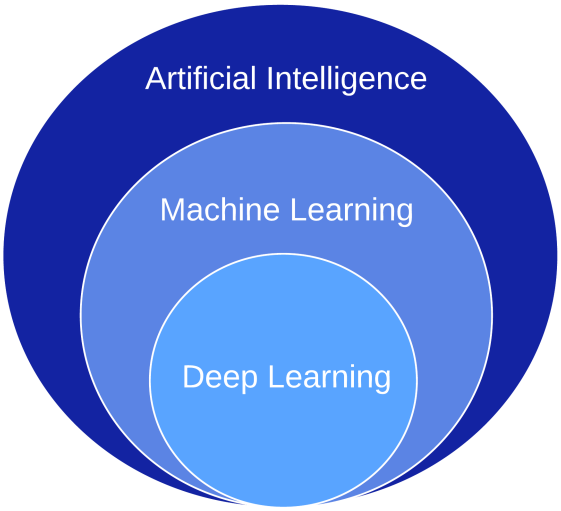
Background Theory and Fundamentals

2.1 *Fundamentals of Deep Learning*

Over the past few years, the field of artificial intelligence (AI) has grown exponentially, playing a transformative role in various research areas and industries. AI focuses on designing intelligent machines capable of executing tasks that typically necessitate human intelligence. Machine learning, a subset of AI, employs algorithms and statistical models to empower machines to learn in an automatized form, without explicit programming, the discovery of rules within the data for solving complex problems (CAI; BILESCHI; NIELSEN, 2020; GOODFELLOW; BENGIO; COURVILLE, 2016).

One of the most promising branches of machine learning is deep learning, the hierarchy between those fields is exemplified in Figure 2.1. The deep learning subfield utilizes multiple stacked layers of processing units to learn and recognize patterns from raw data. The term "deep" in deep learning refers to the numerous hidden layers these networks possess, enabling them to learn increasingly intricate features and representations (CHOLLET, 2018). Deep learning has achieved outstanding results in diverse applications, such as computer vision, natural language processing, and speech recognition.

Figure 2.1: Artificial Intelligence, machine learning and deep learning



Source: Modified from Chollet (2018).

There are four learning paradigms that guide the development and application of algorithms: supervised learning, unsupervised learning, reinforcement learning and, semi-

supervised learning. These paradigms differ in their approach to data and the objectives they aim to achieve.

Supervised learning, the paradigm primarily used in this research, is an approach wherein labeled data, where the input-output pairs are explicitly provided, serves as a guide or instructor to help the algorithm learn to make predictions or classifications on new, unseen data (GOODFELLOW; BENGIO; COURVILLE, 2016). This guidance, allows the algorithm to understand the desired output for a given input, not through memorization, but by learning the underlying patterns or relationships. This learning is achieved by iteratively adjusting the model's parameters to improve its performance, ultimately leading to better generalization when encountering new, unseen data. Supervised learning tasks are typically divided into two categories: regression and classification problems.

On contrast, unsupervised learning, refers to the absence of a guide or instructor to provide annotated examples as the target to guide the learning process (GOODFELLOW; BENGIO; COURVILLE, 2016). Instead, the algorithm explores the data independently and identifies the underlying structure, relationships, or patterns within it. This self-guided learning process can be more challenging, as the algorithm has to make sense of the data without any explicit guidance.

Reinforcement learning, another key paradigm, for that an agent learning how to properly interact with the environment by receiving rewards or penalties as feedback depending on its chosen actions (CHOLLET, 2018).

Lastly, semi-supervised learning operates in an intermediate space between supervised and unsupervised learning. It uses a combination of a small amount of labeled data and a large amount of unlabeled data for training. For that, unsupervised learning techniques can be used to identify and comprehend the underlying structure in the input variables. Meanwhile, supervised learning methodologies can be used to make educated predictions for the unlabeled data. This predicted data can subsequently be used as additional training data within the supervised learning algorithm, and this enhanced model can then be applied to predict outcomes on new unseen data.

While all four paradigms provide unique advantages, this research will focus on the application of supervised learning due to its proven ability to provide accurate and reliable predictions when the target outcomes are well-defined and labeled data is available.

In the following section, the concepts related to machine learning and deep learning are deeply explored by examining various types of machine learning, such as the difference between classification and regression problems, and providing a comprehensive analysis of neural network architectures. The subsection 2.1.2 is dedicated to neural networks

and covers the training process, different architectures, and crucial hyperparameters. The subsection 2.1.3 explores the evaluation metrics that were effectively used. This comprehensive examination will facilitate a deeper understanding of the model's design and potential capabilities to accurately predict lightning strikes, which is a classification problem that requires an unsupervised deep learning model.

2.1.1 Regression and Classification Problems

Data representation plays a crucial role in machine learning and deep learning, as it determines how information is processed and utilized by the algorithms. The types of problems that can be tackled by supervised learning can be separated by the way the output is represented, whether it is continuous numerical data or a discrete class of labels. Understanding the nature of the data is essential for addressing different types of problems effectively.

Continuous numerical data represents values that can assume any value within a specified range. This type of data is common in regression tasks, where the goal is to predict a continuous target variable. Examples of continuous numerical data include temperatures, prices, distances, and various other measurements that can take a continuous range of values.

Discrete class labels represent categorical data, where each instance belongs to one of the predefined categories or classes. When classifying data points, the type of classification problem can vary depending on whether each data point should be assigned to only one category or can belong to multiple categories. If each data point should be assigned to only one category, the problem is a single-label, multiclass classification problem, which involves assigning instances to a single class from several predefined categories. In contrast, binary classification involves determining between two mutually exclusive categories, while multilabel classification involves assigning instances to multiple categories(CHOLLET, 2018). Examples of discrete class labels include species of plants, types of clothing, sentiment labels in text analysis, and other categorical attributes.

It is important to note that machine learning algorithms, including deep learning models, can only process digital information. Consequently, categorical data often needs to be converted into numerical representations to be utilized effectively by these models. There are several techniques available for encoding class labels in a manner that can be easily understood by deep learning algorithms.

For classification problems, encoding class labels is a critical preprocessing step. By transforming categorical data into a numerical format, machine learning practitioners can

ensure that their models can effectively process and learn from the information contained in the dataset. This, in turn, helps improve the overall performance and reliability of the models in making accurate predictions and classifications.

Several techniques can be employed to achieve this transformation, depending on the nature of the data and the problem at hand. One common technique is one-hot encoding, where each class is represented by a binary vector with a length equal to the number of classes, with a '1' in the position corresponding to the class and '0' elsewhere, the prediction is the probability of that data point to belong to each class(BROWNLEE, 2018). Other methods include ordinal encoding, where each category is assigned a unique integer value based on its rank, and binary encoding (also called dummy variables), which converts the integers from ordinal encoding into binary code. The binary encoding follows the same principle as the one-hot encoding but reduces the size of the vector needed to represent the data in case of too many classes (BROWNLEE, 2020). By choosing the appropriate encoding technique, machine learning practitioners can ensure that their models can effectively process and learn from the categorical data, ultimately improving their performance and reliability.

2.1.2 Principles of Neural Networks

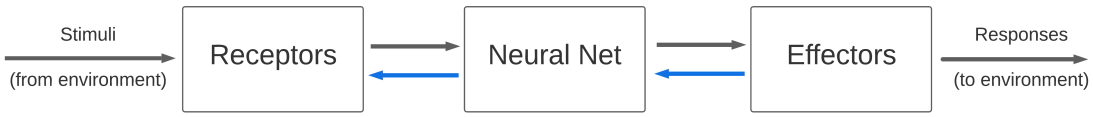
Artificial neural networks (ANNs) are a type of machine learning algorithm that is modeled after the structure and function of the human brain. ANNs are a powerful tool for solving complex problems that involve large amounts of data, such as image or speech recognition. At their core, ANNs are a series of interconnected nodes, or artificial neurons, that process and transmit information. These nodes are arranged in layers, each responsible for processing a specific type of input data.

ANNs are an abstraction of reality, though, simplifying the complexity of the human brain and its processes into mathematical functions and algorithms. Therefore, it only takes inspiration from the brain's ability to learn and does not attempt to replicate it perfectly. Additionally, ANNs are limited by the quality and quantity of data used to train them, as well as the assumptions made during their design.

According to Arbib (1987), the basic premise for constructing an ANN model is that the neural system's operation relevant to it is solely mediated through the transmission of electrical impulses by specialized cells called neurons, simplifying the neural system into a network of neurons. The nervous system streamlined process is shown in Figure 2.2 with three basic stages. The receptors receive input, which is information from the human senses. An intricate neural network in which the information that flows through is processed, resulting in the generation of impulses that control the effectors. The effectors

are responsible for producing the body's response to the given stimuli. The process of transmitting information-bearing signals forward through the system is depicted by the black arrows in Figure 2.2, where information flows from the receptors to the effectors. On the other hand, when the flow is in the opposite direction, it indicates the presence of feedback in the system, which is illustrated by the blue arrows.

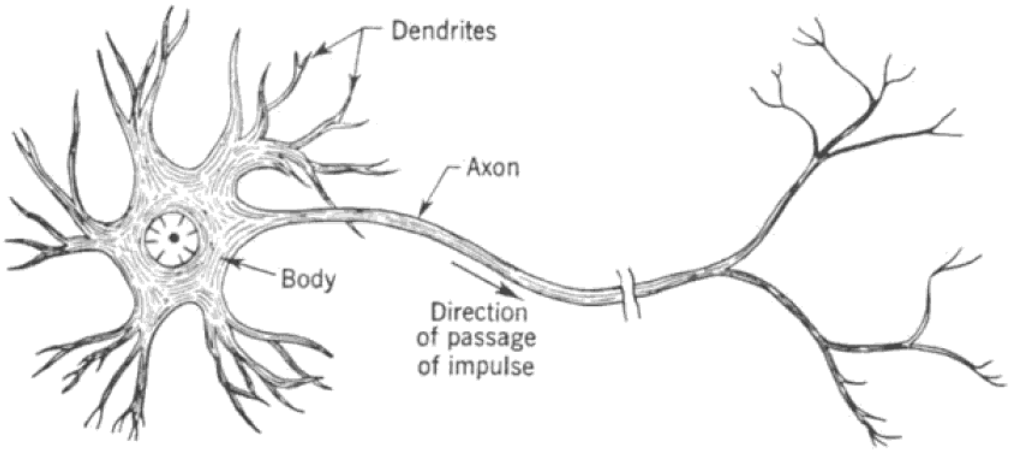
Figure 2.2: The nervous system diagram



Source: Modified from Arbib (1987).

The neuron is the fundamental unit for this process and is made up of dendrites, axons, and a cell body, as in the schematic drawing in Figure 2.3. The dendrites receive input from other neurons in the form of neurotransmitters, which are chemical signals that bind to specific receptors on the dendritic spines. The axon is a long, slender projection of the neuron that carries the electrical signals, away from the cell body towards the axon terminals. At the axon terminals, the electrical signal is converted back into a chemical signal through the release of neurotransmitters, which bind to receptors on the dendrites of other neurons, and the process repeats (HAYKIN, 2009).

Figure 2.3: Biological Neuron



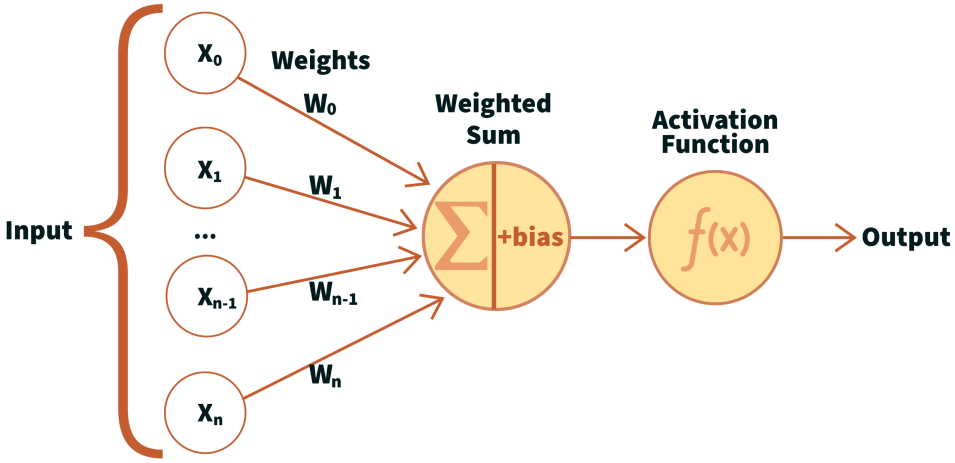
Source: Arbib (1987).

A neuron's firing is dependent on the summation of signals received from other neurons through its dendrites, called a period of latent summation. These signals can either excite or inhibit the neuron from firing an impulse. For a neuron to fire, the excitatory signals must exceed the inhibitory signals by a critical threshold. To model this behavior, weights positive can be assigned to excitatory synapses and negative weights to inhibitory synapses. The total weighted sum of inputs is then compared to the threshold, and if it

exceeds the threshold, the neuron fires an action potential along its axon (ARBIB, 1987).

Analogously, an artificial neuron, also referred to as a perceptron, is a pivotal information-processing unit in a neural network. The block diagram of a neuron model, as shown in Figure 2.4, serves as the cornerstone for designing a diverse range of neural networks with varying functionalities.

Figure 2.4: Perceptron



Source: Developed by the author.

In biological systems, neurons receive input from other neurons via dendrites, process information in the cell body, and transmit electrical signals through axons. Similarly, in ANNs, perceptrons receive input from other perceptrons and process the information. Both biological neural networks and ANNs comprise interconnected neurons, with information flowing from input (receptors) to output (effectors) layers. Feedback mechanisms exist in both systems, as represented in Figure 2.2, though their specific mechanisms in ANNs are purely mathematical.

Neural networks learn by adjusting the weights and biases of the connections between neurons. The learning process typically involves two main phases: forward propagation and backpropagation. During forward propagation, the input data is passed through the network, and the activations of neurons are computed layer by layer. The output layer's activations represent the model's predictions. In the backpropagation phase, the error between the predictions and the actual target values is used to update the weights and biases in the network to minimize a loss function.

The connections between the neurons have a corresponding weight associated. The weights are numerical values that are used to scale the input values from the previous layer before they are passed on to the subsequent layer. When data are introduced to the model, the input values are multiplied by their corresponding weights, and the

products are then summed at each neuron in the first hidden layer. This weighted sum is along with a bias term and passed through an activation function, which introduces non-linearity into the model. This process happens inside each neuron of the network and is schematically represented in Figure 2.4. The output of the activation function for each neuron in the first hidden layer is then used as input for the neurons in the subsequent layer, and the process is repeated until the output layer is reached (HAYKIN, 2009).

The final output prediction is then compared to the corresponding true label for the example, and the derivative of the loss function with respect to the predicted output is computed. The loss function serves as a measure of how well the neural network's predictions match the actual target values. Specifically, it quantifies the difference between the predicted output and the true output for a given input. The ultimate goal of the training process is to minimize this loss function to achieve high accuracy in predicting new, unseen data.

Backpropagation is the fundamental algorithm for training neural networks to minimize the loss function. To elaborate, backpropagation operates through computing the gradient (partial derivatives) of the loss function concerning each weight and bias in the network. This process starts from the output layer and moves backward through the network, updating the weights along the way. The chain rule of calculus is applied to efficiently calculate the gradients. These updates are made by subtracting a fraction of the computed gradients from the current weights and biases.(GOODFELLOW; BENGIO; COURVILLE, 2016) By iteratively adjusting the weights and biases using the computed gradients, the neural network can learn the underlying patterns in the data and improve its performance on the given task.

2.1.2.1 Understanding Key Hyperparameters

When training neural networks, there are several crucial hyperparameters that can be adjusted to optimize the model's performance. These hyperparameters include the number of layers and neurons, batch size, epochs, learning rate, among others, and they play a significant role in how the model learns from the training data. The ultimate goal is to uncover the underlying patterns in the data and make accurate predictions on unseen data. Thus, carefully tuning these hyperparameters is essential to building effective neural networks.

Batch size refers to the number of training examples used in a single update of the neural network's weights and biases. Brownlee (2018) tested different batch sizes for a classification problem and found that small batch sizes tend to lead to rapid learning, but also result in a volatile learning process with higher variance in the classification

accuracy. On the other hand, larger batch sizes slow down the learning process in terms of the learning curves. However, the final stages of training typically result in convergence to a more stable model with a lower variance in classification accuracy. Concluding that, although the optimal batch size depends on the specific problem and available data, it has a significant impact on the speed and stability of the learning process, making it an important hyperparameter to tune.

The learning rate determines the extent to which weights are modified at each epoch. When training a neural network using backpropagation, the amount of error attributed to each weight in the network is estimated. The weight is then updated using a fraction of the estimated error, which is determined by a hyperparameter known as the learning rate. For instance, a learning rate of 0.1, which is a common default value, means that weights in the network are updated by 10% of the estimated weight error each time the weights are updated (BROWNLEE, 2018). If the learning rate is too large, the model may not converge to the correct answer, but if it is too small, it could require significant computational resources.

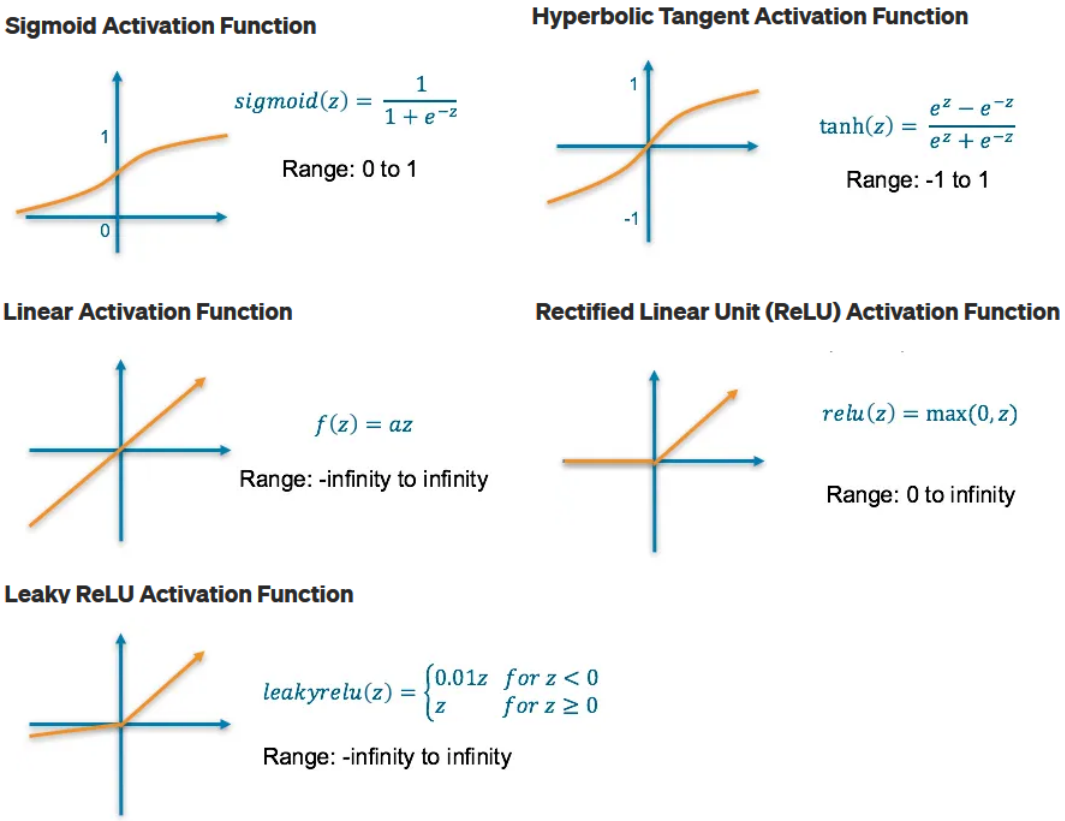
Epochs refer to a single pass through the entire training dataset during the training process. During each epoch, the neural network processes all the batches and updates its weights and biases accordingly (EKMAN, 2021). Typically, the training process involves multiple epochs to ensure that the model has seen the training data multiple times and has had a chance to learn the underlying patterns. The optimal number of epochs depends on the problem and dataset and is often determined through experimentation or by using techniques such as early stopping to prevent overfitting.

As previously discussed, the activation function plays a crucial role in determining the final output of a neuron, or of another neuron that it is connected to, by applying a weighted sum of its inputs. The behavior of the network and the characteristics of its output are therefore heavily influenced by the activation function that is used. Figure 2.5 provides a graphical representation and the corresponding mathematical functions for some commonly used activation functions.

As depicted in the figure, each activation function has unique behavior and range. For example, in the step function, the output is limited to either 0 or 1. This can be too restrictive for other applications, hindering the network's ability to learn effectively. Furthermore, since it returns a fixed value based on a specific threshold, its derivative is almost always zero, rendering it unsuitable for use in backpropagation, which relies on the calculation of derivatives to optimize the neural network.

On the other hand, the sigmoid function can produce continuous outputs within the same range, allowing for more flexibility in learning. Alternatively, hyperbolic tangent (TanH)

Figure 2.5: Graphical and Mathematical Representation of Commonly Used Activation Functions



Source: Modified from Ronaghan (2018).

function has the same shape as the sigmoid but ranges from -1 to 1, allowing for negative numbers as well. This can be advantageous in some cases, such as when the input data has negative values.

The softmax function is commonly used for multiclass classification problems, unlike the sigmoid function, which is often used in the output layer for binary classification problems. However, the dimension of the softmax function graph cannot be easily determined, since it depends on the number of classes in the problem. The softmax function, defined in equation 2.1, where z_i is the input value for the i-th element of the vector, and the function returns the probability of the input belonging to the i-th class. The denominator of the equation ensures that the sum of all probabilities for all classes is equal to 1 (RONAGHAN, 2018).

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{2.1}$$

The linear activation function is a basic function that has its own limitations. It simply

returns the input value, which can lead to problems with gradient calculation during backpropagation. The gradient for a linear function is constant, which means that the same weight and bias updates are applied to all input values, regardless of their value, making it difficult for the neural network to effectively learn and capture complex patterns from the data (SZANDAŁA, 2021). However, there are alternative activation functions that can address these limitations.

One popular alternative is the Rectified Linear Unit (ReLU), which is both simple and effective. ReLU returns the input value if it is positive and zero otherwise. By introducing non-linearity, ReLU allows for more complex patterns to be learned, and the gradient is not constant, allowing for more effective backpropagation. As a result, ReLU is often the preferred activation function in deep learning models. However, in some cases during training, if the weights of a network are initialized in a way that causes numerous neurons to output negative values, those neurons can become "stuck" in a state where they always output zero, regardless of their input, and since the ReLU does not support numbers under zero this prevents from learning (SZANDAŁA, 2021).

As a response, Leaky ReLU is a variant of ReLU that allows a small gradient when the input is negative, which can help to overcome the "dying ReLU" problem. These activation functions can improve the performance of a neural network and allow for more complex representations of the input data.

Another variation, intended to prevent the issue with the ReLU where the negative region is not activated, is the Scaled Exponential Linear Unit (SELU) activation function. This makes SELU distinct, as it has active functionality in both positive and negative regions (KILIÇARSLAN; ADEM; ÇELIK, 2021).

Activation functions play a crucial role in the performance of deep neural networks, and selecting the appropriate function for a particular task can have a significant impact on the network's success. Table 2.1, incremented upon the work of Szandała (2021), provides a comprehensive overview of the recommended use of various activation functions in deep neural networks.

Additionally, to establish a resilient model, it is crucial to avoid evaluating the model with the same data utilized during training. Doing so would introduce an unjust bias in the outcomes, as the model has already encountered the same input and its ability to extrapolate to unfamiliar data would remain unchallenged (CHOLLET, 2018). Consequently, data is typically segregated into three separate sets: training, validation, and testing. The training set is used for the actual learning process, during which the neural network adjusts its weights and biases based on the patterns in the data. The validation set is used to evaluate the model's accuracy on previously unseen data after each training

Table 2.1: Activation Function Overview

Function	Comment	When to use?
Step	Does not work with backpropagation algorithm	Rather never
Linear	Constant gradient; Disregard input value during weight and bias update	Output for regression problems
Sigmoid	Prone to the vanishing gradient function and high fluctuations during training due to not being zero centered	Can fit into Boolean gates simulation
TanH	Also prone to vanishing gradient	In recurrent neural network
ReLU	The most popular function for the hidden layer. Although, under rare circumstances, prone to the "dying ReLU" problem	First to go choice
Leaky ReLU	Comes with all pros of ReLU, but due to not-zero output will never "die"	Use only if "dying ReLU" problem is expected
SeLU	Self-normalizing property; Effective in environments where keeping the mean and variance of the outputs constant over time is beneficial	Preferred where it can help maintain stable activations mitigating the vanishing gradient problem; When dealing with negative inputs (KILIÇARSLAN; ADEM; ÇELIK, 2021)
Maxout	More advanced activation function than ReLU, immune to "dying," but high computation cost	Use as last resort
Softplus	Similar to ReLU, but a bit smoother near 0. Comes with comparable benefits as ReLU, but has a more complex formula, therefore network will be slower	Rather never
Swish	Same as leaky ReLU, and does not outperform it. Might be more useful in networks with great dept	Worth trying in very deep networks
Softmax		For output layer in classification networks
Openmax		For output layer in classification with open classes possibility

Source: Modified from Szandała (2021).

epoch. This helps to monitor the model's performance. Lastly, the test set provides a final evaluation of the model's performance, assessing its ability to generalize and make accurate predictions on entirely new data.

Through meticulous hyperparameter tuning and monitoring of the model's performance on the validation set, it is possible to train a neural network that can effectively learn patterns in the data and generalize for new, unseen examples. This iterative process of training, validation, and testing helps to ensure that the model is not merely memorizing the training data but is genuinely capable of making accurate predictions beyond the known data.

The loss function is a fundamental hyperparameter of neural networks and is used to evaluate the performance of AI models. The selection of a loss function is closely related to the activation function employed in the output layer of a neural network, as specific functions cater to different types of problems. For instance, there are tailored functions for regression problems, binary classification, and multi-class classification problems (BROWNLEE, 2018). Table 2.2, which is based on Cai, Bileschi e Nielsen (2020), provides an overview of some of the loss functions that for its correspondent machine-learning task. Also, it's useful to consider that a function that penalizes high errors heavily may cause the network to struggle to reach the desired minimum values and be very sensitive to outliers, while a function with low penalties may generate a model that does not learn sufficiently well during the training process. This must be adjusted according to the needs of the problem. Additionally, it is possible to add a regularization term to the loss function responsible for preventing large weights from being assigned to the network, another way to avoid overfitting.

Table 2.2: Machine-Learning Tasks, Last-Layer Activation Function, and Loss Function Overview

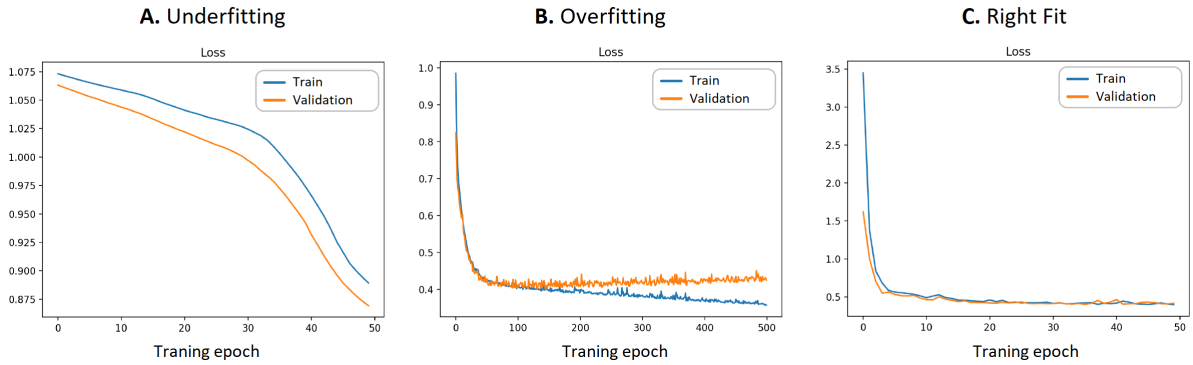
Type of task		Activation of the output layer	Loss function
Regression		Linear, ReLU, TanH	Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Huber Loss
Classification	Binary classification	Sigmoid	Binary Cross Entropy, Hinge, Squared Hinge
	Single-label, Multiclass classification	Softmax	Categorical Cross Entropy, Sparse Categorical Cross Entropy, Kullback-Leibler Divergence

A model with high accuracy will learn from known examples and generalize from those

known examples to new examples in the future. A model that did not learn enough is said to have underfitted, and it will perform poorly on training data as well as in the validation. To address underfitting, the complexity of the model can be increased, improving its capacity to fit a variety of functions. Increasing the capacity of a model can be achieved by changing its structure, such as adding more layers and/or more nodes to layers. In contrast, overfitting is more commonly encountered, as it occurs when a model learns the training data too well, including the noise and irrelevant patterns, losing its ability to generalize with unseen data. (BROWNLEE, 2018)

Diagnosing an overfit model can be done by monitoring its performance of the loss function through the epochs during training on both the training dataset and a validation dataset. A typical pattern for overfitting models is that the training performance continues to improve, while the validation performance plateaus or starts to degrade as shown in Figure 2.6 panel B. In such cases, techniques like regularization, early stopping, or reducing the model's complexity can be employed to mitigate overfitting and improve the model's ability to generalize to new data (BROWNLEE, 2018).

Figure 2.6: Example of Training Learning Curve for Three Simplified Cases: Underfitting (panel A), Overfitting (panel B), and Just-Right Fitting (panel C).



Source: Modified from (BROWNLEE, 2018)

2.1.2.2 Architectures

Neural networks, a highly adaptable and powerful class of machine learning models, are suitable for a wide range of tasks, from image recognition to natural language processing and beyond. The flexibility of neural networks lies in their ability to accommodate a variety of architectural designs, which can be tailored to specific tasks and data types. This adaptability is crucial for achieving optimal performance in diverse machine-learning applications. These architectural configurations encompass the number of layers, layer types, neuron count per layer, interconnections between layers, and the activation functions employed in each layer. This myriad of neural network architectures enables the

customization of neural networks to cater to various tasks and data types, substantially influencing the network's overall performance. Consequently, selecting the appropriate neural network architecture is a crucial consideration in the design and execution of a high-performing machine learning model.

This discussion will delve into a selection of prominent neural network architectures, including multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), long short-term memory (LSTM) networks, gated recurrent units (GRUs), and hybrid architectures such as CNN-LSTM and CNN-GRU.

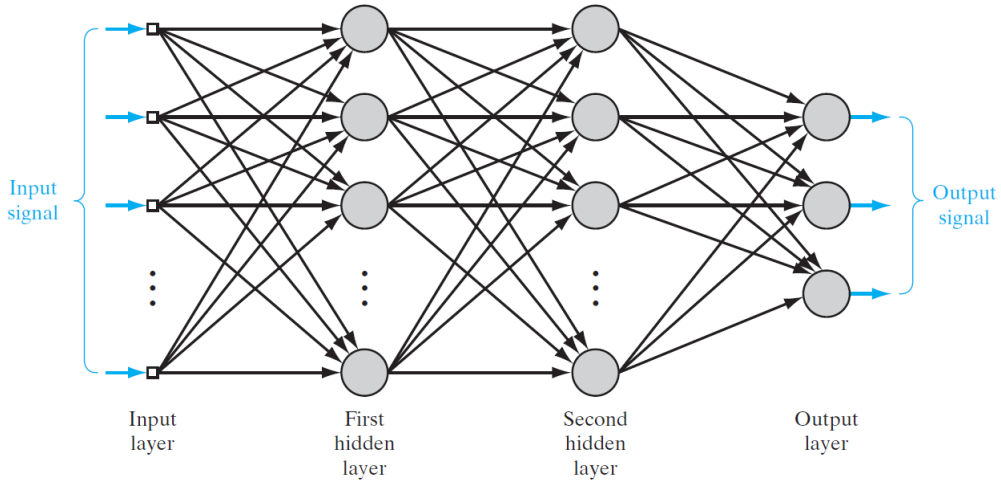
Each architecture has specific strengths and weaknesses, making them suitable for different tasks and data types. By understanding the fundamental principles behind these architectures and their applications, practitioners can make informed decisions when designing and implementing high-performing machine learning models. As the field of artificial intelligence continues to evolve, the development of new and improved neural network architectures will undoubtedly play a crucial role in advancing our ability to solve complex problems and enhance various aspects of our lives.

MLPs are a fundamental type of neural network architecture, composed of multiple fully connected layers where all neurons in adjacent layers are interconnected, earning the name of a fully connected network (EKMAN, 2021). The information flow in MLPs is unidirectional, moving from input to output without any loops or recurrent connections, classifying them as feedforward networks (AGGARWAL, 2018). Often regarded as the foundation for more complex neural network architectures, MLPs serve as an essential building block in the field of artificial intelligence.

Through successfully addressing the limitations of single-layer networks, the MLP emerges as an innovative and impactful neural network architecture. It achieves this by incorporating nonlinear activation functions, hidden layers, and high connectivity between neurons. To be considered an MLP the network has to be composed of at least three fundamental layers: the input layer, a hidden layer, and an output layer. Figure 2.7 demonstrates an example of a structure for an MLP with two hidden layers. The hidden layers are made up of the same perceptrons as the input and output layers, with as few or as many layers as needed to compose the network's depth. Despite their importance, those layers cannot be directly accessed by the programmer, since they are located in the middle of the network, serving only as part of the processing units and not generating the final output. As a result, they are referred to as "hidden" (GOODFELLOW; BENGIO; COURVILLE, 2016).

The neural network processes function signals, which are input signals that propagate through the network and produce output signals. Conversely, error signals originate from

Figure 2.7: Multi-layer Perceptron Architecture with Two Hidden Layers



Source: Haykin (2009).

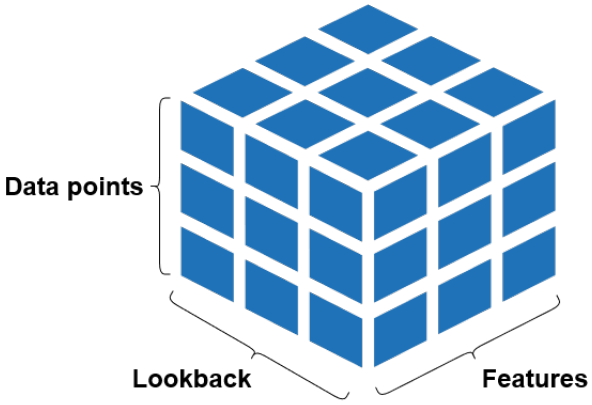
the output layer and propagate backward through the network layer by layer for error correction. To compute the function signal, each neuron in the network applies the input signal and associated weights, then performs a nonlinear function. Moreover, each neuron estimates the gradient vector, which is used for error correction during the backward pass through the network. Each neuron in the hidden or output layers performs both the computation of the function signal and the estimation of the gradient vector (HAYKIN, 2009).

Despite its effectiveness, the fully connected network has some limitations that should be considered. One of the drawbacks is that it provides limited information to the network at a time, making it difficult to work with time-series predictions that involve seasonal trends, as they cannot be learned with just one historical data point (EKMAN, 2021). Additionally, the network still does not have access to all historical data unless the input layer is infinitely wide, which is not practical. This limitation means that the network may not be able to capture all the temporal dependencies in the data, which can lead to suboptimal performance when working with time-series forecasting.

However, recurrent neural networks (RNNs) and CNNs are able to overcome this limitation, as they are commonly employed as they are able to process sequential data of varying lengths. RNNs, in particular, are well-suited for time-series forecasting due to their ability to capture temporal dependencies in the data. These models can utilize both current and past, time steps of a variable to provide more accurate predictions (GOODFELLOW; BENGIO; COURVILLE, 2016). To utilize these models effectively, the data must be pre-processed to incorporate the notion of time, by defining a lookback window, as represented in Figure 2.8. This creates a new dimension in the data that indicates how many steps in the past the network should consider while analyzing each data point,

providing a contextual frame of reference for the input data. By doing so, the network can learn meaningful patterns for prediction based on a historical analysis of the data.

Figure 2.8: Three-Dimensional Data: Lookback Window Representation



Source: Developed by the author.

RNNs are designed to handle sequential data that have a temporal relationship. Unlike feedforward neural networks, the input vector for a recurrent layer needs to include both the current input and the previous output. That means that the RNNs have loops in their structure that allow information to persist and are able to take into account past information when processing new inputs. This makes RNNs particularly useful for tasks in which the sequence of the data is relevant, such as natural language processing, and speech recognition, where the meaning of a particular word or sound is often dependent on the preceding words or sounds.

In those networks, the hidden neurons serve as the network's internal state. By using feedback loops to connect the output of the hidden layer back to the input layer via unit-time delays, the network can maintain a memory of past inputs and process sequences of data. The input layer of an RNN consists of feedback nodes and source nodes that are concatenated together. Feedback nodes receive the output of the hidden layer from the previous time step, while source nodes receive input from the external environment. The number of unit-time delays in the feedback loop determines the order of the model. This dictates the number of time steps that the network can remember and use to inform its processing of new input. A higher-order model can capture longer-term dependencies in the data, but at the cost of increased computation and a greater risk of overfitting (HAYKIN, 2009).

RNNs can be trained using backpropagation through time (BPTT), which involves unrolling the network and backpropagating the error through the network in the same way as for a feedforward network. However, since RNNs have recurrent connections, weight sharing needs to be taken into account when updating the weights (EKMAN, 2021). However, this introduces both the vanishing gradients problem and the exploding gradient problem.

If the recurrent weight is smaller than 1, it will be multiplied by itself many times, resulting in a value that approaches 0. This causes the gradients used in backpropagation to become very small, making it difficult to update the weights in earlier layers and leading to the vanishing gradient problem. Conversely, if the recurrent weight is larger than 1, it will be multiplied by itself many times, resulting in a value that approaches infinity. This causes the gradients to become very large, leading to the exploding gradient problem.

Although deep networks with too many layers are prone to vanishing gradients, in the case of RNNs, weight sharing across time steps can compound the problem. This is because the impact of certain inputs on the output layer is not limited to the neurons directly below it, but rather extends to all the neurons from previous time steps, leading to a further weakening of gradients.

Various modifications have been proposed to address these issues, including using different activation functions that avoid saturation, regularization techniques to prevent overfitting, and specialized RNN architectures such as LSTM networks and GRUs. These innovative architectures were designed to overcome the vanishing gradients problem and have proven to be effective in many applications.

LSTMs are highly popular recurrent neural networks that differ from other types of networks in the way they update the states in the hidden layer and propagate information over time. These networks have different memory mechanisms that allow the recurrent neurons to better retain information over longer periods, preventing the loss of information during processing (CHOLLET, 2018). The LSTM network is designed to interact with the memory cells via the gates, which allow for the formulation of output using the state from previous activations of the neuron.

There are three types of gates that help to govern the flow of information within the cell in an LSTM network: the forget gate, input gate, and output gate. The forget gate determines what information should be discarded from the cell, while the input gate decides which values from the input should be used to update the memory state. The output gate determines what should be outputted based on both the input and the memory of the cell. These gates are weighted functions that allow the LSTM network to selectively remember or forget information, which is critical for handling long-term dependencies in sequential data (BROWNLEE, 2017). These gates, together with the constant error carousel (CEC), keep each cell stable and allow for bridging long-time lags. The internal architecture of each memory cell guarantees a constant error flow within the CEC.

This is achieved by using a neuron that implements a neuron with the derivative of the activation function as 1, a recurrent weight of 1, and a loop that is formed through time.

The ramifications of the output of this neuron passing through this series of gates is that the network's ability to propagate gradients through time is maintained, preventing the vanishing or exploding gradient problem(EKMAN, 2021).

The cells in GRU networks are simplified versions of LSTM cells, and like LSTMs, they have a memory mechanism but with much fewer parameters. GRU networks are commonly applied in problems with few parameters, where processing speed is more important (CHOLLET, 2018). In particular, GRUs have only two gates, a reset gate, and an update gate, which control the flow of information into and out of the cell.

The reset gate in a GRU network determines how much of the previous hidden state should be carried over or forgotten. This allows for a partial copy of the hidden state from the previous layer, making gradient flow more stable during backpropagation. The update gate in a GRU combines the roles of both the input and forget gates in an LSTM, enabling the network to selectively decide how much of the previous hidden state to retain and how much to update with the new input at each time step (AGGARWAL, 2018). Although the GRU is generally simpler and more computationally efficient, the LSTM is often viewed as a safer option for processing longer sequences and larger datasets due to its explicit internal memory and greater control over updates through the use of separate gates.

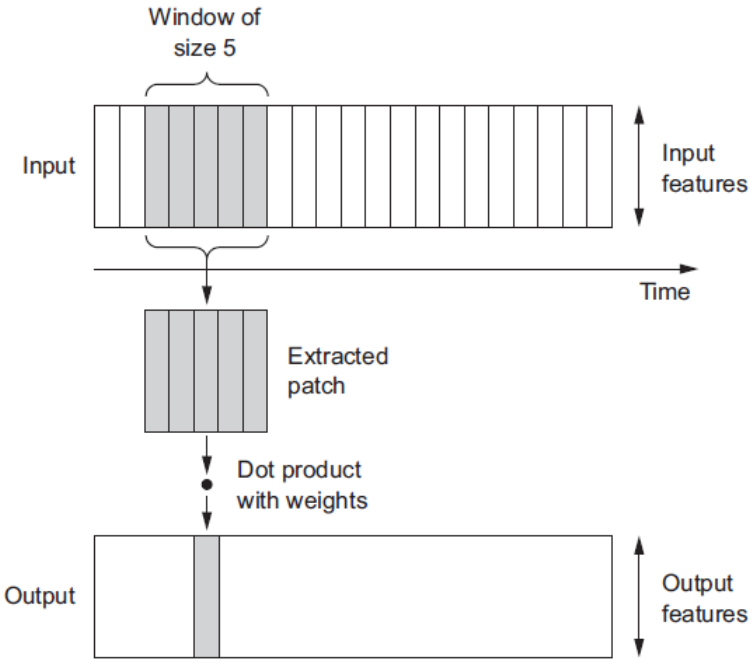
After exploring the benefits and limitations of recurrent networks, it's worth considering another powerful model in deep learning, the convolutional neural network. It is worth noting that CNNs have also become a popular choice in the field of deep learning. Unlike GRUs and LSTMs, CNNs are commonly used for processing images, but they have also been successfully applied in other domains, including time series forecast. CNNs use a convolutional structure to capture complex patterns and relationships in the input data, and they are highly effective in scenarios where local spatial or temporal relationships between input data points are important.

In CNNs, filters or kernels are applied to the input data to obtain a set of feature maps that are then passed through non-linear activation functions. These filters allow the network to efficiently extract relevant features from the input data, while the pooling layers help in reducing the dimensionality of the feature maps that will pass through subsequent layers.The convolutional operation in a CNN involves computing the dot product between the filter weights and a spatial region in the previous layer to determine the hidden state in the next layer(AGGARWAL, 2018).

This network was originally designed for processing and classifying image data, which naturally have three dimensions - height, width, and color channels (or feature maps), excluding the batch size dimension. However, with the rise of time-series data analysis, a modified version of CNNs called Conv1D was developed to handle sequential data.

Time-series data only has two dimensions - the lookback window and the features - which makes it incompatible with traditional CNNs. Conv1D, on the other hand, is specifically designed to operate on sequential data by performing 1D convolutions across the time dimension, allowing it to learn and extract patterns and features from sequential data. So while CNNs were originally developed for image processing, Conv1D was developed to address the unique challenges of sequential data analysis(CHOLLET, 2018). Figure 2.9 shows how the convolution operation works with time series sequential data, where a kernel slides through the patches of data with size 5 and outputs the dot product between it and the weights in the kernel.

Figure 2.9: Convolution Operation on Time Series Sequential Data



Source: Chollet (2018).

The kernel size determines the size of the window that will slide over the input data, while the stride determines the distance that the filter will move after each convolution operation. When a stride of 2 is used, for example, the filter moves two units at a time horizontally and vertically across the input image. This results in the downsampling of the feature map by a factor of 2 in both dimensions. Padding is a technique used to enable the convolution operation to be applied to the edges of the feature map as well as the central regions. It involves adding an appropriate number of rows and columns of zeros around the edges of the input feature map. The purpose of this is to prevent the output feature map from shrinking and to ensure that the convolution operation can be applied to the entire input. By adding padding, the output feature map will have the same spatial dimensions as the input feature map.(CHOLLET, 2018)

While 1D CNNs are effective at capturing local temporal patterns, they can be less sen-

sitive to long-range dependencies in time series data, such as seasonal trends in meteorological data, unlike the RNNs that are designed to handle that type of sequential data (CHOLLET, 2018). This is due to the fact that they process input patches independently and are limited by the size of the convolution windows. Hybrid models, such as CNN-LSTM and CNN-GRU, address this issue by combining convolutional and recurrent networks and have shown great promise in handling time-series data due to their ability to capture both spatial and temporal dependencies within the data. This is also particularly useful for handling multivariate data, which recurrent networks traditionally struggle with.

The model first applies a convolutional layer to the input data, which applies filters to the input data to extract relevant features. Each filter is a small matrix of weights that convolves over the input data, producing an output feature map. The output feature map contains feature maps for each filter and summarizes the input data's most important features.

Next, a pooling layer is applied after the convolutional layer, which reduces the spatial dimensions of the feature maps while retaining the most important information. The pooling layer outputs a feature map that has reduced spatial dimensions and preserves the most important information and is done by extracting windows from the input feature maps and outputting the max value of each channel. The max-pooling uses the max value of each channel, and it is preferred over other downsampling methods, such as strided convolutions and average pooling, as it tends to lead to results(CHOLLET, 2018).

The output of the pooling layer is then fed into a recurrent layer, such as an LSTM or GRU layer. The RNN layer takes the pooled feature maps as input and applies the RNN operation to model the temporal dependencies in the input data. Finally, one or more dense layers can be added to the model to perform classification or regression tasks on the RNN layer's output.

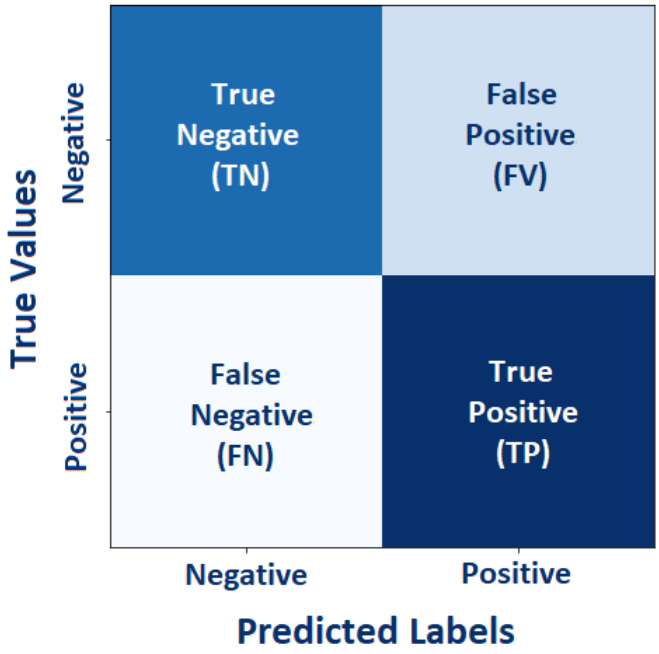
2.1.3 Evaluation metrics for deep learning classification models

This study aims to provide a binary forecast for future lightning events by predicting the presence or absence of lightning in the next six hours. To ensure a proper evaluation of the deep learning classifier model, multiple evaluation metrics need to be analyzed.

A confusion matrix is an essential tool for evaluating the performance of a classification model as it provides a clear visual representation of the quantities and proportions of predictions classified correctly, and incorrectly classified. The matrix has two axes, with the true labels of the values the model is using as targets on the y-axis and the predicted

values on the x-axis. The diagonal of the matrix represents the values that were classified correctly. True positives (TP) represent the correctly classified positive class values, while true negatives (TN) represent the correctly classified negative class values. False negatives (FN) occur when a data point that should have been classified in the positive class is wrongly classified in the negative class, and false positives (FP) occur when a data point that should have been classified in the negative class is wrongly classified in the positive class. Figure 2.10 shows the confusion matrix for a binary classification problem, but the same logic can be applied to construct a confusion matrix for a multiclassification problem.

Figure 2.10: General Confusion Matrix



Source: Developed by the author.

One commonly used metric is accuracy, as defined in Equation 2.2, which measures the proportion of correct predictions made by a model over the total number of predictions. While accuracy is easy to understand and interpret, it may not be suitable for all scenarios, especially when the classes are imbalanced or when the cost of false positives or false negatives is high. (EKMAN, 2021)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.2}$$

Precision (equation 2.3) is the fraction of TP out of all predicted positives. In other words, precision measures the proportion of instances that were correctly classified as positive among all instances that were predicted as positive by the classifier. High precision

indicates that the classifier has a low false positive rate, meaning that the instances classified as positive are highly likely to be truly positive. However, high precision does not necessarily imply high recall (equation 2.4), which measures the fraction of TP among all actual positive instances. Both have to be taken into consideration, since, the choice of which metric to prioritize depends on the specific application requirements. For example, for lightning occurrences, high recall may be more important than high precision because it is crucial to identify all positive cases even if it means some false positives may be included. For that reason, the F1-score 2.5) makes a valuable metric, since it takes into account both precision and recall, providing a balanced measure of the model's ability to correctly identify positive samples while minimizing FP and FN.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.4}$$

$$\text{F1-Score} = \frac{2 \cdot (\text{ Precision } \cdot \text{ Recall })}{\text{Precision} + \text{Recall}} \tag{2.5}$$

Specificity, is the equivalent of the precision for the negative class, measuring the proportion of TN out of all the actual negatives, as the equation 2.6 shows. Therefore, specificity tells us how well the model can correctly identify negative cases in relation to all the negative cases present.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{2.6}$$

The Receiver Operating Characteristic Area Under the Curve(ROC AUC) provides a single number that reflects the overall ranking performance of the model. The ROC curve is a graphical representation of the trade-off between the true positive rate (recall) and the false positive rate (equation 2.7) at various classification thresholds. The ROC AUC is a measure of the model's ability to distinguish between positive and negative classes. A perfect classifier has a ROC AUC score of 1, while a random classifier has an AUC score of 0.5. The ROC AUC is a robust metric that is insensitive to class imbalance, making it useful for evaluating the performance of models in real-world scenarios. ROC AUC metric is equal to the likelihood that a negative example chosen randomly will have a lower probability estimate of belonging to the positive class compared to a positive example chosen randomly. (HUANG; LING, 2005)

$$\text{False Positive Rate} = \frac{FP}{TN + FP} \tag{2.7}$$

The Precision-Recall Curve (PRC) is obtained by plotting the precision against the recall for various threshold values. As previously stated, ideally, a good classifier should achieve high precision and high recall simultaneously, but these two metrics are typically in tension with each other. Therefore, the PRC curve provides a visual representation of the trade-off between precision and recall, and the AUC of the PRC curve measures the overall quality of the classifier. (VUJOVI'c, 2021)

The area under the curve can be calculated by using the trapezoidal rule, which involves approximating the area under the curve as a series of trapezoids. This method involves calculating the PRC AUC for each segment of the curve between two adjacent points, using the trapezoidal rule, and then summing the areas of these segments to obtain an estimate of the AUC for the entire curve. Miao e Zhu (2021) defines the equation as seen in equation 2.8, where f is the function of FPR and t is of TPR, and the trajectory is partitioned into $n - 1$ sections.

$$PRCAUC_{trapezoid} = \frac{1}{2} \sum_{i=1}^{n} (f_{1+i} - f_i) \cdot (t_{i+1} + t_i) \tag{2.8}$$

To calculate the AUC for a ROC curve, the trapezoidal rule involves dividing the curve into a set of vertical strips of equal width and summing the areas of the trapezoids formed by each strip. The AUC can then be calculated as the sum of these areas. This metric will be better discussed in the next subsection 2.1.4.

2.1.4 DeLong's test

DeLong's test is a statistical method proposed by DeLong, DeLong e Clarke-Pearson (1988) used for comparing the areas under two or more correlated ROC curves. This can be used in many fields, but it's frequently used in medical statistics where ROC curves are a common tool to visualize and analyze the performance of diagnostic tests but can also be used to compare machine learning model performance, such as in the works of Furtado et al. (2022), Guliyev e Mustafayev (2022), and Rahman et al. (2021).

Furtado et al. (2022) the Delong's test was used to compare a deep learning algorithms that support the detection of pneumonia caused by COVID-19 in chest radiographs with a well-known public solution. Guliyev e Mustafayev (2022) applied to compare machine

learning models trained to predicted West Texas Intermediate (WTI) oil price dynamics. Rahman et al. (2021) used both Wilcoxon's signed ranked test and and DeLong's test to compare five machine learning models and three deep learning architectures created to classify a driver's cognitive load. This cognitive load represents mental effort and workload experienced by a person while driving and it is a variable that correlates with the disposition of road accidents.

The DeLong test offers a way to statistically compare the performance of two models or tests based on their ROC curves, taking into account the correlation between them. This is important because the AUC is a summary measure of the ROC curve, which represents the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance.

DeLong's method for comparing the AUCs of two correlated classifiers has several steps and involves the computation of a covariance matrix and a z-score statistic.

DeLong, DeLong e Clarke-Pearson (1988) stated that Receiver Operating Characteristic (ROC) curve, when computed using the trapezoidal rule, is equivalent to the Mann-Whitney two-sample statistic applied to the two sample sets X_i and Y_j. Thus, the AUC can be computed as the equation 2.9.

$$AUC_{empirical} = Mann - Whitney statistic = \hat{\theta} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \psi(X_i, Y_j) \tag{2.9}$$

In this context, m represents the number of positive instances and n represents the number of negative instances. The sets X_i and Y_j correspond to the predicted probabilities for the positive and negative classes, respectively. In adition, the equation 2.9 adhere to the constraints imposed by the ψ function, also known as the Heaviside step function (equation 2.10).

$$\psi(X_i, Y_j) = \begin{cases} 1 & \text{if } Y < X \\ \frac{1}{2} & \text{if } Y = X \\ 0 & \text{if } Y > X \end{cases} \tag{2.10}$$

The logic behind this is that it has a positive impact on the AUC value when the probability of making accurate predictions is higher. In another words, when Y $<$ X, this means that the predicted occurrence probability of a no lightning event is less than the predicted occurrence probability of a lightning event. This is desirable because no lightning

instances should have a lower predicted occurrence risk than actual lightning instances. Therefore, the model is rewarded for its accurate prediction, which contributes positively to the model's AUC with a value of $+1/mn$ (DRAELOS, 2020).

When Y = X, this means that the predicted occurrence probability of a no lightning event is equal to the predicted occurrence probability of a lightning event. In this case, the contribution to the model's AUC is $(+1/2)/mn$. In the worst case scenario where Y > X, it signifies that the predicted occurrence probability of a no lightning event is greater than the predicted occurrence probability of a lightning event. No contribution is added to the model's AUC (+0) (DRAELOS, 2020).

Consequently, when considering the probabilities (Pr), it is important to note that according to the Heaviside function, the weight assigned to $Pr(X = Y)$ is zero. Therefore, when calculating the mean of the AUCs, the following approach is employed of the equation 2.11.

$$AUC_{mean} = \theta = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} Pr(X > Y) + 0.5 \times Pr(X = Y) \tag{2.11}$$

The next step involves computing the DeLong covariance, which takes into account the correlation between the two classifiers. The construction of this covariance matrix is carried out as follows:

$$S = \frac{Sx}{m} + \frac{Sy}{n} \tag{2.12}$$

Where Sx and Sy are these covariance matrixes computed based on the ranking of the predictions, Sx is used for the positive examples and Sy for the negative examples.

After calculating the AUCs and their covariance, a test statistic (denoted as z) can be computed. This statistic measures the standard deviations difference between the AUCs, normalized by their covariance. It's given by:

$$z = \frac{(AUC_A - AUC_B)}{sqrt(var(AUC_A) + var(AUC_B) - 2 * cov(AUC_A, AUC_B))} \tag{2.13}$$

where $var(AUC_A)$ and $var(AUC_B)$ are the variances of AUC_A and AUC_B respectively, and $cov(AUC_A, AUC_B)$ is the covariance between AUC_A and AUC_B.

Then, the p-value associated with this Z-score is calculated using the survival function (also known as the complementary cumulative distribution function) of the standard normal distribution. In the formula 2.14, $|z|$ represents the absolute value of the z-score. The standard normal cumulative distribution function, represented by $\Phi(.)$, gives the probability of observing a value less than or equal to z in a standard normal distribution with mean 0 and standard deviation 1 (DEMLER; PENCINA; D'AGOSTINO, 2012).

$$p_{value} = 2(1 - \Phi(|z|)) \tag{2.14}$$

In statistical hypothesis testing, the null hypothesis is a statement that assumes there is no significant difference or relationship between two or more models. The z-value, calculated based on the output data, is used to determine the level of deviation from the null hypothesis.

When the null hypothesis is true, the z-value can be approximated by the standard normal distribution. In a standard normal distribution, approximately 95% of the values fall within 1.96 standard deviations of the mean. Therefore, as Sun e Xu (2014) stated, if the z-value calculated for a particular comparison deviates too much from zero (e.g., z $>$ 1.96),it implies that the observed difference between the groups is unlikely to have occurred due to random chance alone, supporting the idea that the models of interest are not equal.

2.2 The Physics of Lightning

To begin discussing the physics of lightning, it's important to understand that lightning is a natural electrical phenomenon that occurs mostly during thunderstorms. Thunderstorms are a highly intricate weather phenomenon, and predicting future changes in lightning activity requires a comprehensive understanding of the complex interrelationships between different atmospheric system components. For a thundercloud to form, first when the air in contact with the ground gets warmer and rises, is necessary that this rising air parcel contains water or humidity. As the air parcel rises, it cools down and the moisture in the air starts condensing into microscopic water droplets, forming visible clouds. Due to the fact that the rising air parcel contains moisture, it cools down at a slower rate than normal, which is called the wet adiabatic lapse rate. If the temperature of the atmosphere decreases faster than the wet adiabatic lapse rate, the air parcel continues to rise, forming a thundercloud, also known as cumulonimbus (COORAY, 2015).

Cooray (2015) describes that the full cycle of a thundercloud consists of three stages: the

cumulus stage, the mature stage, and the final stage. During the cumulus stage, the air rises through the atmosphere and forms a tall cloud. During the mature stage, the cloud is capable of generating lightning flashes, and downdrafts occur as falling particles drag down the surrounding air. According to Almeida (2016) an isolated cloud can keep active for approximately 20 to 40 minutes. Finally, during the final stage, the cloud vaporizes and disappears. Thunderstorms can be caused by the direct heating of moist air at ground level by the sun, by the lifting of incoming air along the slopes of a mountain, or by the penetration of cold air into a region with warm moist air. It is necessary to note that thunderstorms are not the only source of lightning, they can also be generated by sandstorms, volcano eruptions, and around the fireballs of a nuclear explosion.

Thunderclouds have a typical charge structure with three vertically stacked point charges: a main positive charge at the top, a main negative charge in the middle, and a lower positive charge at the bottom. According to Rakov e Uman (2003) that is an increasing consensus that the graupel-ice mechanism is the dominant mechanism for cloud electrification, with collisions between graupel and small ice crystals producing electric charges in the presence of water droplets.

2.2.1 Lightning formation and classification

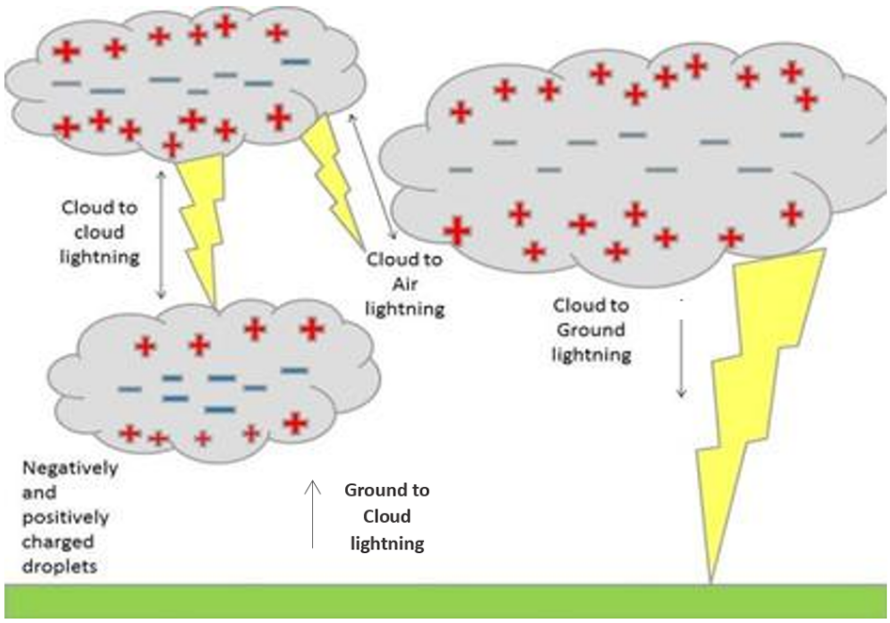
Lightning is an abrupt and massive discharge of electrical energy in the atmosphere that can produce a voltage ranging from 300,000 to tens of millions of volts, with a peak current ranging from 5,000 to 200,000 amperes. Lightning occurs when the atmospheric electromagnetic field within a thunderstorm cloud exceeds the dielectric strength or insulating capacity of the air at a certain location. The cloud becomes polarized, with the positive and negative charges separate, and while small ice crystals and splinters positively charged are drawn upwards the graupel-ice, charged negatively descends (ELSOM, 2015). Lightning is then produced by the electric charges generated in collisions that happen on those different ice particle paths within a thunderstorm cloud. The majority of electric charges are generated at altitudes where the temperature is between -10º C and -20º C, at approximately 5 to 6 kilometers above ground level, and wind currents can transport them within the cloud (PINTO JUNIOR; PINTO; REGINA, 2021).

The first discharge to make its way from the thundercloud toward the ground is called a stepped leader, which leaves a downward path of ionized air. After it hits the ground a return stroke is created, which is a streak that flows upward through the path, the stepped leader is traced because after the first connecting is created the electrons near are free to flow rapidly through. Although the flash appears to be upward, the electrons are flowing earthward (ELSOM, 2015).

Lightning flashes can take different paths depending on the polarity of the charge being transported and the charged regions involved. Positive ground flashes also occur but are less common. In some cases, lightning flashes can also be triggered by tall structures at ground level, resulting in upward-initiated lightning flashes. Finally, bipolar lightning flashes occur when a lightning flash starts as a negative or positive ground flash but later makes a connection to the opposite charge center and transports a charge of opposite polarity to the ground (RAKOV; UMAN, 2003).

In regard to the path, Figure 2.11 illustrates the various types of lightning paths. These include intracloud lightning flashes, which occur between two main charge centers within a single cloud, and intercloud lightning flashes, taking place between opposite charge centers of neighboring clouds. There are also air discharges, which involve lightning flashes that originate from a cloud but do not reach the ground. Lastly, Cloud-to-Ground flashes, the most common type of lightning, occur between the negative charge center and the ground (ALMEIDA, 2016).

Figure 2.11: Types of Lightning



Source: The Royal Meteorological Society (2017).

The cloud-to-ground lightning flashes can also be distinguished by the direction it travels, ascending or descending. Descending negative flashes is the most common type, being responsible for 90% of the occurrences. The ascending ones represent only 0.1% of the total (PINTO JUNIOR; PINTO; REGINA, 2021).

2.2.2 Climatological scenario affects lightning occurence

Anthropological and natural factors can influence the increase or decrease in lightning generation. The proximity to bodies of water, topography, and temperature are some of the natural factors, the ones influenced by human presence are urban areas, heat islands, and pollution. Larger scale factors, such as continental distribution, and the presence of air masses that moves according to the atmospheric global circulation, are responsible for indicating the distribution of lightning across the globe, but regional variations are more important to determine local behavior(PINTO JUNIOR; PINTO; REGINA, 2021).

It is possible to perceive that certain meteorological conditions are more prone to lightning activity. For instance, during El Niño events, there is generally an increase in lightning activity and thunderstorms in tropical land areas, while La Niña events tend to have the opposite effect. This is attributed to the relationship between ENSO and the vertical development of clouds into the ice region, which affects lightning activity. The percentage of occurrences increasing from strong La Niña to strong El Niño varies from 17 to 45% in the South and Southeast, 28% in the North, and more than 400% in the Northeast (PINTO JUNIOR, 2015).

Moreover, in regards the influence of the SST of the Tropical Atlantic Ocean, the TSA running mean is above 25%, the likelihood of lightning occurrences increases, particularly in the North and Northeast regions. In this case, the percentage of occurrences increases from 39%, during spring and summer to 65% during fall and winter in the Northeast. The Tropical North Atlantic (TNA) - the TNA index representing surface temperatures in the eastern tropical North Atlantic Ocean- was found to be influential only in the Northeast region, where the likelihood increases when the Tropical South Atlantic (TSA) - an indicator of surface temperatures in the Gulf of Guinea- is running mean below 25% varies from 63% during spring and summer to 138% in the fall and winter in the Northeast. Overall, the Northeast region also showed the most significant changes in all parameters investigated (ENSO, TSA, and TNA) (PINTO JUNIOR, 2015).

2.3 State of the art

Lightning, a powerful natural electric discharge resulting from charge imbalances within a cloud, between clouds, or between a cloud and the ground, poses considerable risks and can lead to destructive consequences. Therefore, accurate and timely nowcasting of lightning events is of paramount importance. Over the years, numerous studies have been conducted to address this challenge, employing various approaches and methodologies. These endeavors have aimed to enhance our understanding of lightning behavior and

develop effective forecasting techniques to mitigate the potential impact of lightning-related hazards.

For instance, in India, K, Gayatri et al. (2022) focuses on the pre-monsoon season in Maharashtra, and employs the Weather Research and Forecasting (WRF) model with various microphysical schemes to simulate and assess lightning flash counts. The researchers use a dataset consisting of sixteen selected lightning events and analyze both three-hour intervals and a 24-hour period. The results demonstrate promising performance, achieving high accuracy, with a Probability of Detection (POD) ranging from 0.82 to 0.86 and a low False Alarm Ratio (FAR) ranging from 0.25 to 0.29.

Recently, machine learning techniques such as neural networks have gained prominence in meteorological research, due to their ability to process vast amounts of meteorological data, identify patterns in complex relationships among parameters, and adapt and generalize.

Abdullah, Adnan e Ruslan (2018) used a multi-layer perceptron for total lightning per month forecasts in Malaysia, achieving a root mean squared error (RMSE) of 0.9990, illustrating the method's effectiveness. The model development involved using atmospheric observations at ground level, including air pressure, temperature, humidity, wind speed, and precipitation, as inputs for predicting lightning occurrences from Meteorological Malaysian Services (MMS).

Alves et al. (2021) conducted a comparative study of predictive models based on artificial neural networks to forecast cloud-to-ground lightning strikes in the northeastern region of Pará, Brazil. The study utilizes data obtained from the NOAA-19 environmental satellite, with a prediction horizon of up to five hours after satellite passage. The study aimed to address the class imbalance issue by employing the Synthetic Minority Over-sampling Technique (SMOTE) technique to balance the minority classes of lightning occurrences. By comparing the performance of the models trained on raw data and SMOTE processed data, the researchers found that the models trained on balanced data exhibited superior performance, resulting in improved accuracy rates. Specifically, the accuracy rates increased from 82.14% and 90.36% to 86.49% and 94.64%, respectively, when utilizing the balanced data.

The work of Zhou et al. (2020), in China, employed a multi-source dataset consisting of geostationary meteorological satellite data, Doppler weather radar network data, and Cloud-to-Ground lightning location system data within a fully convolutional network with an encoder-decoder architecture, for nowcasting lightning with a lead time of 0-1 hour. The results surpassed those of a traditional algorithm, reaching 0.633, 0.386, 0.931, and 0.453 for the probability of detection, false alarm ratio, and area under the relative op-

erating characteristic curve, respectively. Similarly, in Geng et al. (2020), researchers developed a heterogeneous spatiotemporal network aimed at extracting knowledge from both spatial and temporal domains. By using three data sources, namely WRF simulations, lightning observations, and weather station observations, the network demonstrated superior efficiency compared to baseline models.

Lin et al. (2019) proposed a model called Attention-based Dual-Source Spatiotemporal Neural Network (ADSNet), which combines recent lightning observations and numerical simulations using an attention mechanism. It compared the performance of ADSNet with other models, including lightning parameterization schemes, Gradient Boost Decision Tree (GBDT), and spatiotemporal deep neural network (StepDeep). ADSNet outperformed the lightning parameterization schemes, GBDT, and StepDeep in most metrics. ADSNet achieved the best performance among all the selected methods.

ADSNet's attention mechanism explains the impact of input variables on lightning forecasting. Attention weights analysis shows that maximum vertical wind speed and graupel mixing ratio are influential parameters, consistent with existing lightning parameterization schemes. Furthermore, the distribution of temperature levels corresponds to the temperature range associated with lightning occurrence.

Swiss researchers (MOSTAJABI et al., 2019) developed a four-parameter machine learning model that can nowcast the occurrence of lightning within a 30 km radius up to 30 minutes in advance using commonly available weather parameters from local weather stations. The machine learning model, XGBoost, outperformed three baseline models - persistence, electrostatic field model, and Convective Available Potential Energy (CAPE) model for all lead time ranges, achieving over 76% accuracy in predicting long-range lightning threats. Additionally, a feature reduction analysis showed that including all meteorological variables provided the best results, and sensitivity analysis indicated that surface pressure, relative humidity, and surface temperature were more important than wind speed in predicting long-range lightning activity.

Several studies conducted in South Africa explored the efficacy of lightning forecasting by employing a diverse array of machine learning models and techniques. These investigations focused on evaluating the accuracy and predictive capabilities of different approaches.

Gijben, Dyson e Loots (2017) utilizes lightning data from the Southern Africa Lightning Detection Network (SALDN) of the South African Weather Service (SAWS) and NWP (Numerical Weather Prediction) model data from the Unified Model (UM) at SAWS to develop a lightning threat index (LTI). Through pre-processing techniques including stepwise logistic regression and rare-event logistic regression to address the imbalance in the dataset and feature selection from 25 candidate parameters. The LTI is developed

separately for the spring and summer seasons, and its performance is evaluated through probabilistic assessments. The results show high sensitivity, specificity, and accuracy of the LTI in predicting lightning occurrences, with slightly over-forecasting during spring and reliable forecasts during summer.

First, in Essa, Ajoodha e Hunt (2020) aimed to evaluate the use of an LSTM neural network model on predicting short-term lightning flash densities in Bloemfontein and Piet Retief, two areas in South Africa. The LSTM model utilized historical lightning flash data from the SALDN. The results showed that the model successfully predicted approximately 30% of major lightning events. Specifically, for the Piet Retief area, around 50% of the model's predictions aligned with actual lightning occurrences. However, the accuracy was lower for Bloemfontein, with only 20% of predictions matching observed lightning events.

Later on, Essa, Hunt e Ajoodha (2021) compared different models including LSTM, AutoRegressive (AR), and AutoRegressive Integrated Moving Average (ARIMA) for predicting lightning strikes. The dataset used in the study was the Historical Cloud-to-ground Lightning Data in South Africa for the year 2018, obtained from SALDN. The forecast horizon for the models was short-term, specifically, the number of lightning strokes every three hours. The results showed that the LSTM model outperformed the AR and ARIMA models, with a lower Mean Absolute Percentage Error (MAPE) of 3,705 and RMSE of 9,426. In comparison, the AR model had a MAPE of 15,312 and an RMSE of 8,579, while the ARIMA model had an MAPE of 15,080 and an RMSE of 8,301.

In a subsequent study, Essa et al. (2022) aimed to predict thunderstorm severity in North-Eastern South Africa by utilizing lightning flash frequency data from SALDN and weather station data from the South African Weather Service (SAWS). LSTM followed by a fully-connected network (LSTM-FC), Convolutional neural network LSTM (CNN-LSTM), and ConvLSTM.

The LSTM-FC model combined the LSTM architecture with a fully-connected network, while the CNN-LSTM model incorporated convolutional neural network layers before feeding the output into the LSTM layer. On the other hand, the ConvLSTM model utilized convolutional operations within the LSTM cell gates. Among the three models, the CNN-LSTM model demonstrates the best performance with the lowest MAE of 51 flashes per hour. The LSTM-FC model performs slightly worse with an MAE of 67 flashes per hour, while the ConvLSTM model has the highest MAE of 86 flashes per hour.

Marope et al. (2023) study the prediction of lightning in Johannesburg via logistic regression, random forest, and LSTM machine learning models. These models use parameters such as air temperature, relative humidity, dew point, and electric field data from the Johannesburg Lightning Research Laboratory (JLRL) and SALDN to forecast lightning

within 30 km of Johannesburg's city center.

The challenge of imbalanced datasets, predominantly consisting of no lightning cases, is addressed by employing a threshold-sweeping strategy, which adjusts the decision boundary.

Although the LSTM model showed the highest precision and F1 score, it fell short in lightning prediction effectiveness. In contrast, the logistic regression model excelled with a recall score of 93% and a ROC-AUC score of 90%, surpassing both the random forest model (80% recall and 85% ROC-AUC) and the LSTM model (53% recall and 76% ROC-AUC).

The existing literature reveals a noticeable knowledge gap when it comes to assessing the overall performance and efficacy of lightning forecasting models in diverse geographical areas, particularly on a broader scale. Additionally, it is important to acknowledge the limitations imposed by the absence of meteorological stations in certain locations, which restricts the availability of data for those areas. To mitigate these challenges, incorporating globally available data can enhance the model's adaptability and generalization capabilities, while extending its accessibility to regions lacking local data and remote sensing technologies.

To address these issues, this research will focus on investigating lightning forecasting within a large region that comprises the entire state of Bahia, Brazil, which size is comparable to the countries of Spain and France, for instance. By encompassing a larger geographical area, the study aims to broaden our understanding of how these models perform across diverse regions with different meteorological characteristics. Furthermore, this work will tackle the challenge posed by the absence of meteorological stations in certain locations by utilizing globally available gridded meteorological data.

Therefore, by conducting a comprehensive analysis, considering diverse geographical areas, advanced modeling techniques, and globally available data, this study aims to significantly contribute to the advancement of lightning forecasting methodologies, their applicability in various regions, and the resolution of data limitations. The utilization of complex deep learning models and hybrid architectures will provide valuable insights into their utility and effectiveness in lightning forecasting, turning this research a unique and novel approach that can provide new scientific foundations for better forecasting lightning events across large regions with limited data and computational resources.

Chapter Three

Methodology

This chapter will be discussed in sequence, detailed information about the study site, including its physical and climatic characteristics. Also, the data analyzes the description of how the data was preprocessed, leading up to the structural features of the AI models.

The experiments were conducted using the High-Performance Computing (HPC) system named Ògún at SENAI CIMATEC. The CPU of the system supports both 32-bit and 64-bit operating modes, and it features an Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz. The Python version used for the experiments was 3.9.5, while TensorFlow library version 2.4.1 was employed. For performance evaluation, we made use of scikit-learn library version 0.24.2.

3.1 Environmental Characteristics of Study Area

The geographical location of a study area can have a significant impact on the research conducted within it. The meteorological conditions that prevail in the area can affect the study's outcome. Furthermore, any relevant historical events or trends that have affected the study area in the past should be accounted for when interpreting the data.

Describing the geographical location of the study area, the meteorological conditions that prevail in the area, and any relevant historical events or trends that have affected the area in the past is crucial to providing a better understanding of the context in which the study was conducted.

First, in subsection 3.1.1 an overview of the study area's geographical location is provided. This includes information on the area's size, boundaries, climate and biome classification, and notable topographical features. Next, in subsection 3.1.2 the meteorological conditions that prevailed in the study area, especially during the time in which the data used for this research was acquired.

3.1.1 Overview of Bahia

The study area is the Brazilian state of Bahia, located in the northeastern region of the country, with a diverse landscape and rich cultural history. According to the *Instituto*

Brasileiro de Geografia e Estatística (IBGE, accessed 2023-04-10) census of 2010, the states cover an area of 564,760.429 square kilometers, similar but still larger than European countries, such as Spain and France. Figure 3.1 displays the location of Bahia state, highlighted in blue, as shown, it is bordered by the Atlantic Ocean to the east and the states of Sergipe, Alagoas, Pernambuco, and Piauí to the north. To the south, it is bordered by the states of Minas Gerais and Espírito Santo.

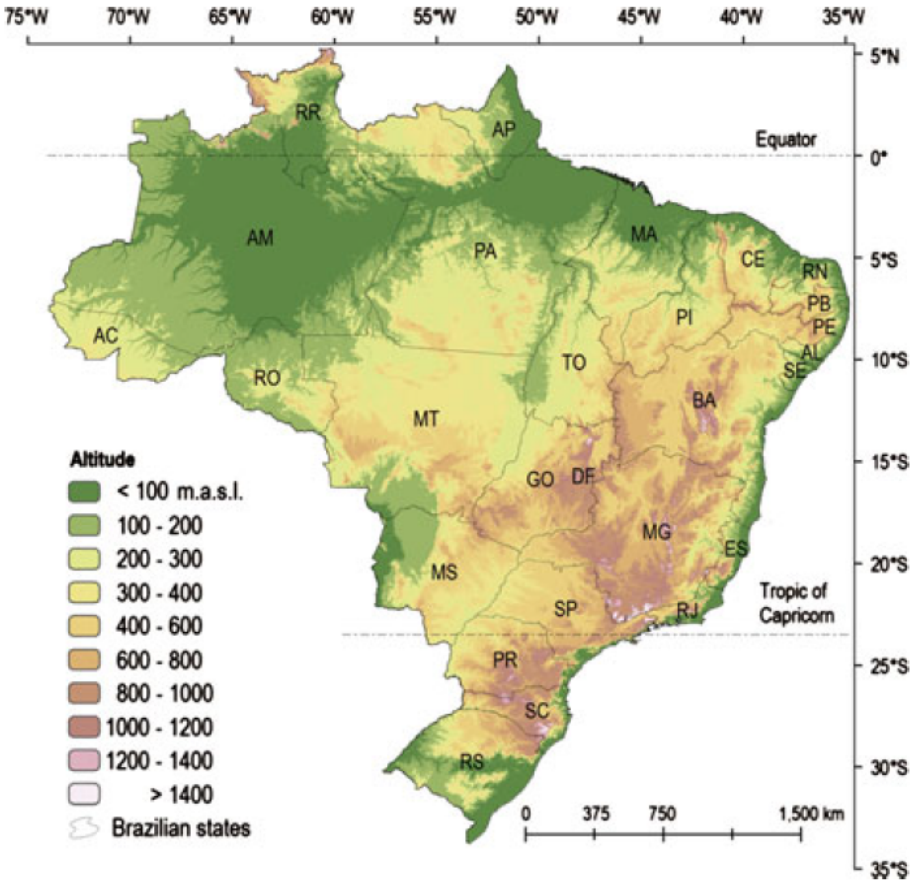
Figure 3.1: Geographical Location of Bahia State

Source: Developed by the author utilizing the (OpenStreetMap contributors, 2023) data.

In 2010, Bahia had a population density of 24.82 inhabitants per square kilometer, which is relatively low compared to more densely populated areas like cities or countries with smaller landmasses and larger populations. This can be attributed to the fact that urbanized areas, which tend to be more densely populated, only cover 2,814.29 square kilometers. However, it is estimated that the state was experienced population growth since, the latest estimated data from IBGE (accessed 2023-04-10) to 2021, the population stands at 14,985,284 people, showing an increase compared to the last census conducted in 2010.

Regarding the topography, Bahia exhibits distinct topographical variations, as evidenced by the elevation map of Brazil presented in Figure 3.2. The state's topography features a diverse landscape, with coastal lowlands in the east and elevated areas in the west-central region, which are only interrupted by the lower elevations of the São Francisco River Valley crossing through the area.

Figure 3.2: Digital elevation model of Brazil.



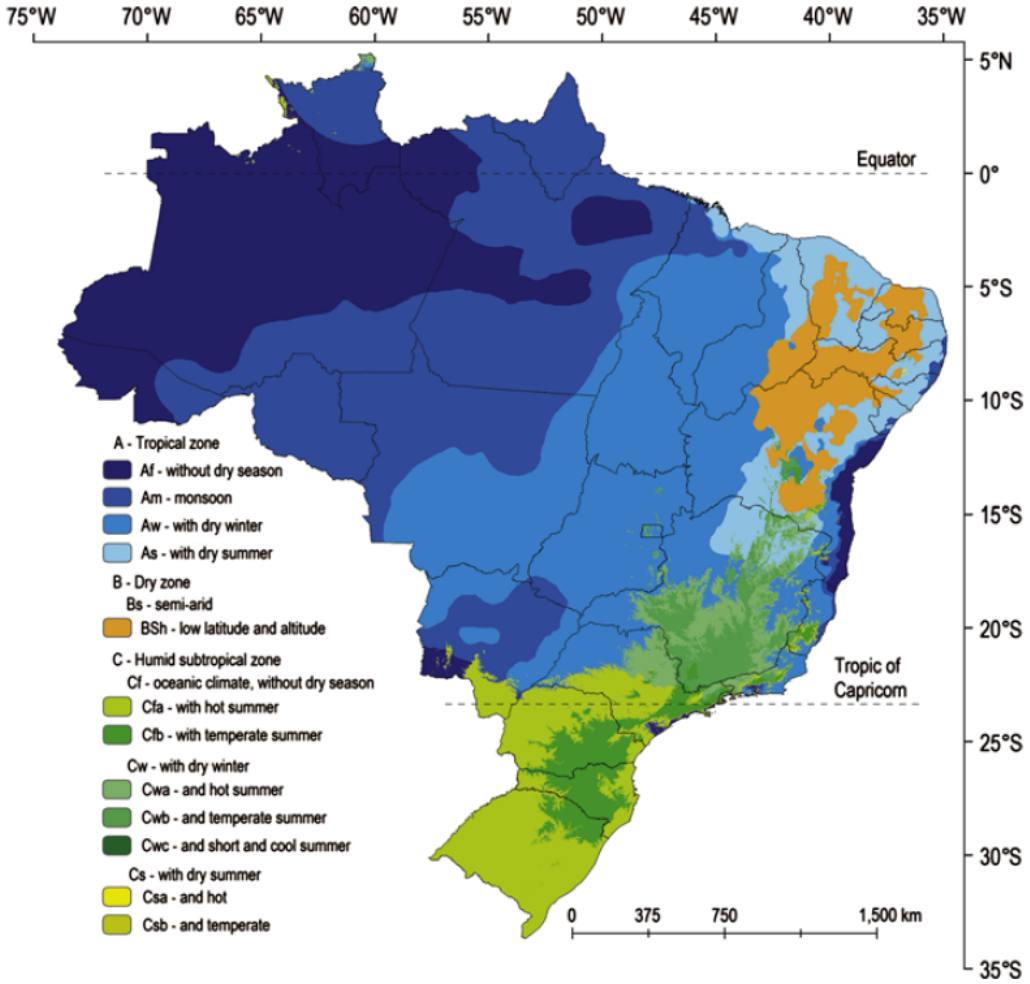
Source: Alvares et al. (2013)

The climate in Bahia varies greatly across its vast territory. According to Almeida (2016) the Köppen (1936) climate classification method was created and refined by Wladimir Köppen is based on an assumption that natural vegetation of each major region on Earth is essentially an expression of the prevailing climate, thus, it incorporates and combines the information on air temperature, precipitation, and seasonal characteristics to classify. According to Alvares et al. (2013) who applied the Köppen method to the Brazilian territory, as shown in Figure 3.3, Bahia presented nine different types of Köppen climate, however, with BSh, Aw, As and Af, being the most prevalent, listed in order of the area covered.

The BSh classification in Köppen's climate classification system refers to a hot semi-arid climate and extends throughout the North and Central region of Bahia. In this

climate type, the annual precipitation is less than the threshold needed to sustain a full cover of vegetation, and potential evaporation rates are high due to the high temperatures throughout the year, which was an annual average temperature above 18°C. The A classes characterize tropical climates, typically featuring high temperatures and precipitation rates. The Aw type, predominant in the western region, refers to a tropical savanna climate with a distinct dry season and consistently warm to hot temperatures throughout the year. The As is also a tropical savanna climate, however, the dry season is during the summer and claims the territory central-eastern part of the state. Finally, the Af is the tropical rainforest climate that is without a dry season and is predominant in the coastal region on the east.

Figure 3.3: Brazil's climate classification according to the Köppen (1936) criteria

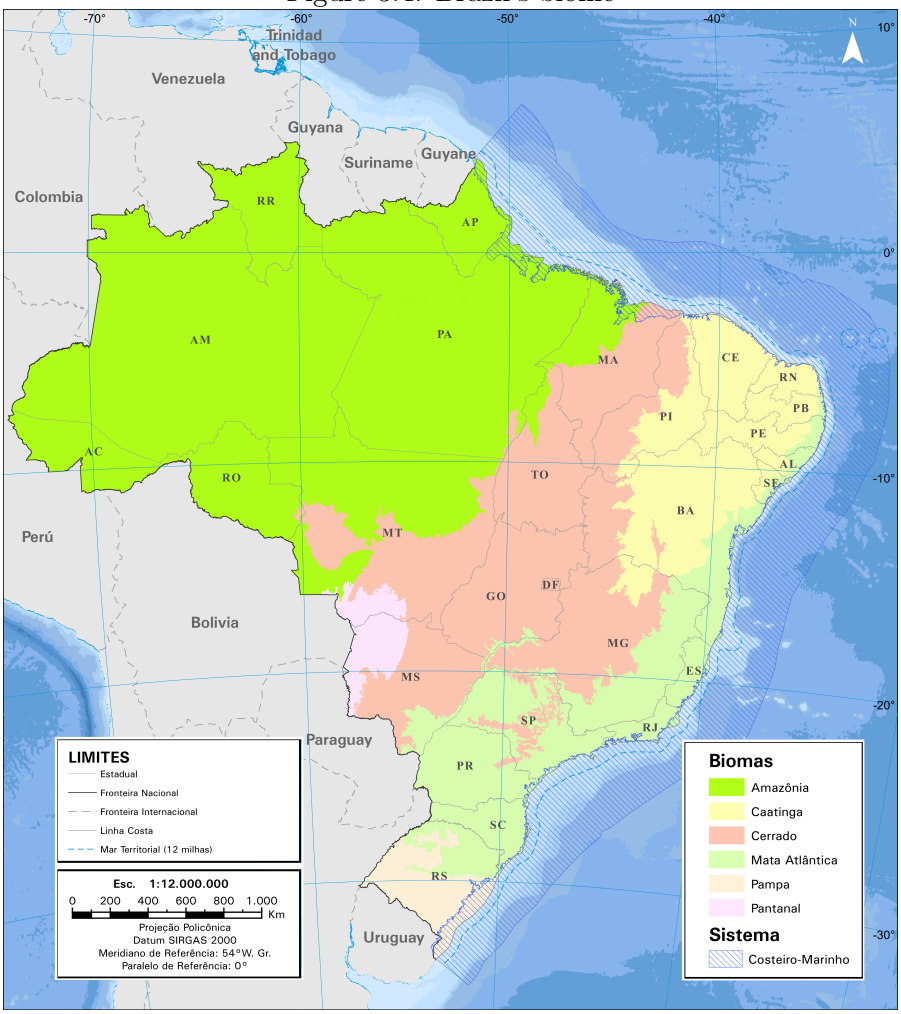


Source: Alvares et al. (2013)

In the Brazilian biome Map in Figure 3.4 it is clear that Bahia has three different biomes: Caatinga, Cerrado and Mata Atlântica. The Caatinga biome, located in a semi-arid climate area, presents a great variety of landscapes and biological richness, with species that are exclusive to this biome. The occurrence of periodic droughts leaves the vegetation

without leaves, which regrows and becomes green during short periods of rain. However, human activities such as deforestation and burning for agriculture and livestock have altered the vegetation, negatively impacting wildlife, water quality, and soil and climate balance. Cerrado is known as the richest savanna due to the wide range of plant and animal species, including many endemic to the region. The Mata Atlântica biome is located on the coast region and is the fifth most threatened and richest biome in endemic species in the world. The biome is currently reduced and fragmented, with remaining forest remnants mainly located in areas that are difficult to access(IBGE, accessed 2023-04-14).

Figure 3.4: Brazil's biome



Source: modified from IBGE (accessed 2023-04-13)

3.1.2 *Climatological scenario*

This study uses data collected between November 2017 and November 2018. It is important to consider the prevailing climatological conditions during this period to identify any

potential anomalies in the input data.

El Niño Southern Oscillation (ENSO) is a well-known atmospheric phenomenon that impacts weather patterns in tropical and mid-latitudinal regions. The two phases of ENSO are El Niño (hot phase) and La Niña (cold phase). These phases have a significant effect on precipitation, causing more rainfall than usual, while in others, it can cause drought due to changes in atmospheric circulation patterns. For example, during the 2015-2016 El Niño event, the Northeast region of Brazil experienced a severe drought due to changes in the rainfall patterns caused by the warming of the Pacific Ocean (GATEAU-REY et al., 2018). El Niño years are characterized by higher sea surface temperatures (SST) in the equatorial Pacific Ocean, resulting in higher air temperatures and lower precipitation rates in Brazil. On the other hand, La Niña years are characterized by lower SST in the equatorial Pacific Ocean, resulting in lower air temperatures and higher precipitation levels in Brazil (SILVA et al., 2020).

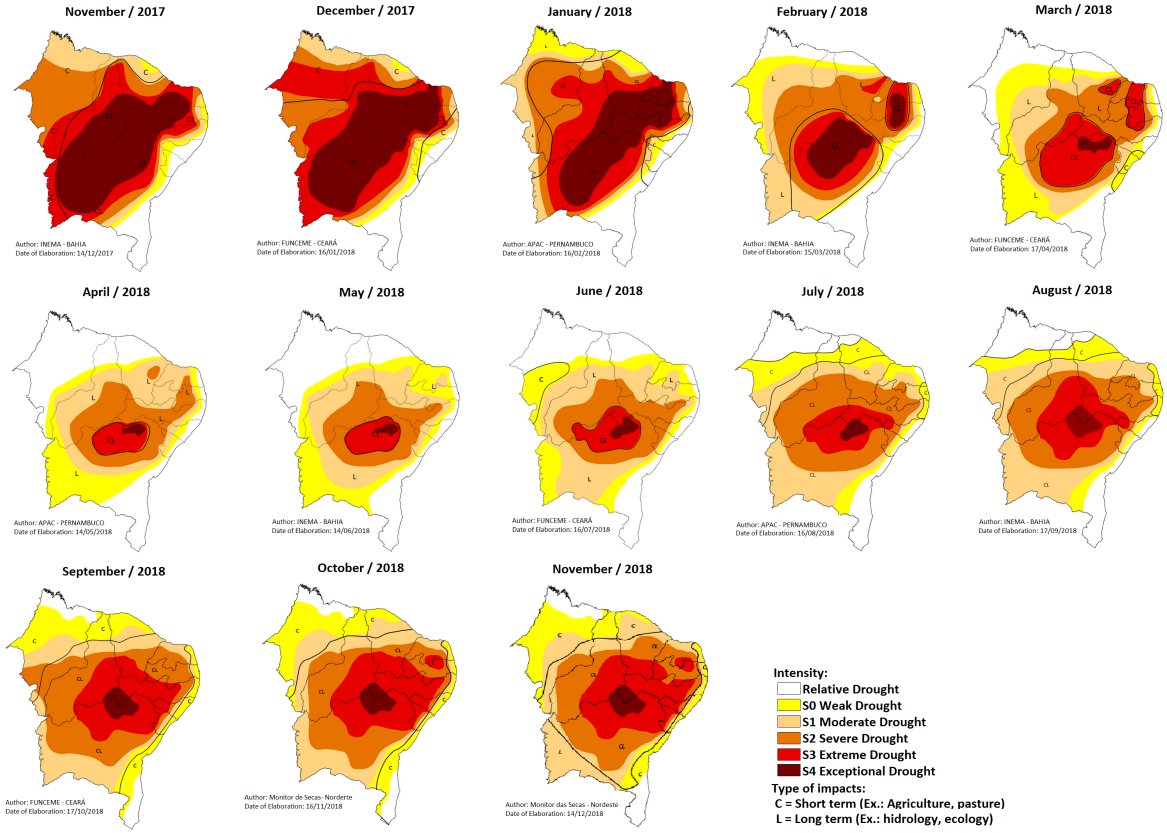
According to L'Heureux (2019), a La Niña occurred between September and November 2017, with the event reaching its peak between November 2017 and January 2018. By early May 2018, the La Niña Advisory was discontinued as the tropical Pacific Ocean returned to an ENSO-neutral state and turned into a weak El Niño around October 2018 until May 2019.

However, it should be noted that the study period coincided with an unusual scenario. As stated by Brito et al. (2021), it marked the end of a severe drought that began in 2012 and gradually decelerated, entering a normal stage in 2018. Its impacts are visible while analyzing the drought monitor in Figure 3.5, which was developed through a multistakeholder effort and is currently coordinated by the *Agência Nacional de Águas e Saneamento Básico* (ANA) in partnership with the *Ministério da Integração Nacional* (MI) and the *Instituto Nacional de Meteorologia* (INMET) at the federal level and with the participation of all Northeastern states. Figure 3.5 represents the affected area and severity of the drought effects in the Northeast within the period of interest. Severe drought was still in a significant portion of the Northeastern region from November 2017 to January 2018 as it continues to decrease until April and May.

3.2 Data Collection

The dataset employed as input in this study encompasses various meteorological variables, collected at six-hour intervals between November 2017 and November 2018 from the Global Data Assimilation System (GDAS). This dataset is the same data utilized in the Global Forecast System (GFS) at the National Centers for Environmental Prediction (NCEP)(National Centers for Environmental Prediction/National Weather Ser-

Figure 3.5: Drought Monitor: Affected Area and Severity in the Northeast from November 2017 to November 2018



Source: Modified from ANA (accessed 2023-04-10)

vice/NOAA/U.S. Department of Commerce, 2015). While this data is globally available, the scope of this research focused exclusively on the region corresponding to the state of Bahia, Brazil. As a result, the dataset was limited to data localized within the specified geographical boundaries, confining the range for latitude from 8°S to 18°S and longitudes from 46°W to 37°W.

Furthermore, the meteorological characteristics of the study area have been detailed in subsection 3.1. However, it is worth noting that Bahia presents nine different types of climate according to the Köppen (1936) classification method (ALVARES et al., 2013), with hot semi-arid climate, tropical savanna climate with dry winter, tropical savanna climate with dry summer, and tropical rainforest, being the most predominant.

As for the target, lightning occurrence data were obtained from the Brazilian Lightning Detection System (BrasilDAT), a network for lightning detection and monitoring managed by the *Grupo de Eletricidade Atmosférica* (ELAT) at the *Instituto Nacional de Pesquisas Espaciais* (INPE) in Brazil.

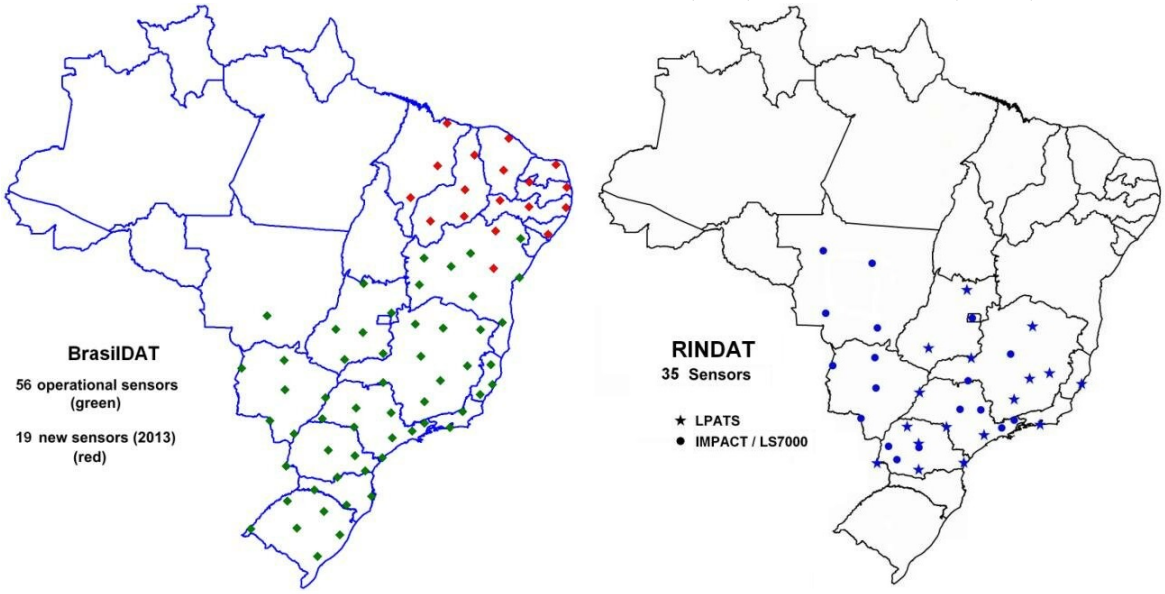
The BrasilDAT dataset is a comprehensive collection of lightning data in Brazil, created by combining information from multiple lightning locating systems. It incorporates data from more than 110 sensors deployed across the country, including the Thunderstorm Location System (TLS), the *Rede Integrada Nacional de Detecção de Descargas Atmosféricas* (RINDAT), and the *Sistema Brasileiro de Detecção de Descargas atmosféricas* (BrasilDAT). The decision to merge data from these systems was made after realizing that each system measured different aspects of lightning activity, making them complementary to one another. Consequently, the data from the various sensors were merged together using a grouping method based on time-of-event, location uncertainty, stroke type, and peak current. This grouping process results in the creation of the BrasilDAT dataset. The dataset captures various aspects of lightning activity, such as cloud-to-ground flashes, intracloud flashes, peak currents, and stroke types. For each stroke detected, the dataset includes precise location data in space and time, as well as information regarding the charge of the current (PINTO JUNIOR; PINTO, 2018).

Naccarato et al. (2012) presented a comprehensive overview of the sensor locations in Figure 3.6. The figure depicted the sensor distribution in 2012, including sensors that were already functional as well as those under construction. The BrasilDAT network eventually expanded to include 75 sensors, providing coverage across various regions in Brazil. On the other hand, the RINDAT network consisted of 35 sensors deployed in 8 states of Brazil, primarily concentrated in the mid-southern region of the country. The hybrid network comprised a combination of low-frequency (LF) sensors such as LPATS, IMPACT, and LS7000, along with very high-frequency (VHF) sensors like LS8000. The latter was still under installation and thus not shown on the map. Based on the available information, it can be inferred that the map presented in this study represents the most recent information on the location of the sensors. However, it is important to note that the map may not reflect the current state due to potential updates or changes since its creation.

3.2.1 Data Exploration

The BrasilDAT dataset spans over a period of 362 days, starting on 20/11/2017 at 20:00 and ending on 17/11/2018. It comprises 1,909,576 data entries detailing the location, time, and electrical current produced by each lightning event. Out of these data points, 475636 are lightning occurrences, while 1758 are non-occurrences. Analyzing this data allows insights into the behavior of lightning during that period. In Figure 3.7, the total number of lightning occurrences registered by month is displayed, showing a more active period from December 2017 through March 2018. This period coincides with the end of the drought in the region. The number of lightning occurrences rises again in October, corresponding to the period when the El Niño phenomenon is starting to take place.

Figure 3.6: Configuration of Sensors for BrasilDAT (Left) and RINDAT (Right) in 2012



Source: Altered from Naccarato et al. (2012).

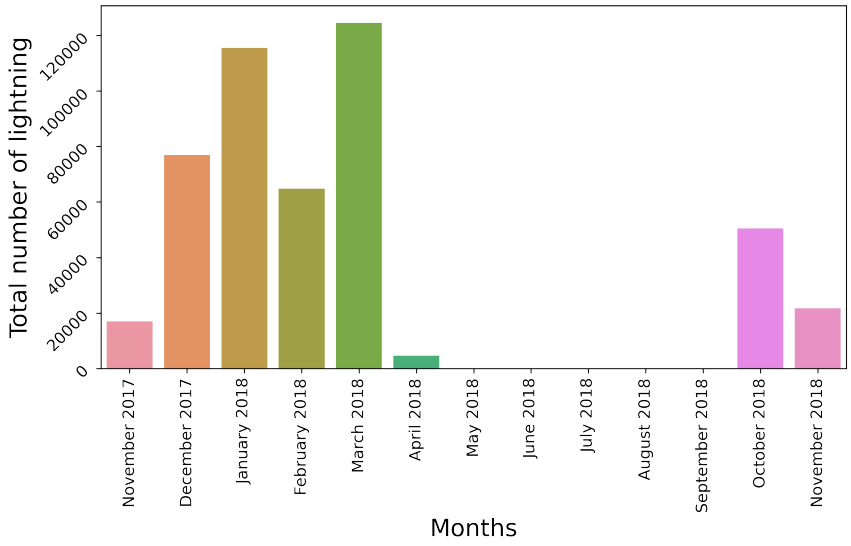
Figure 3.8 provides the information on the time of day this event occurs with more frequency at the end of the day, peaking at 19:00 (military time). Therefore, it can be assumed that the likelihood of a lightning occurrence is near that hour.

Figure 3.9 displays a heatmap illustrating lightning strikes in Bahia throughout the entire study period. The heatmap highlights regions that are more frequently affected by lightning strikes. This visualization was created using BrasilDAT, which has been manipulated to align with a spatial grid at a resolution of 0.25 degrees.

The Figure shows that during this period, the western region of the state, which includes the Cerrado biome and has higher altitudes, has the highest number of lightning activities. In contrast, the central region of the state, while also on higher ground, shows less activity, which can be attributed to the Caatinga biome. This observation aligns with Abreu et al. (2020)'s findings, which note a significant frequency of lightning strikes in the western region of Bahia.

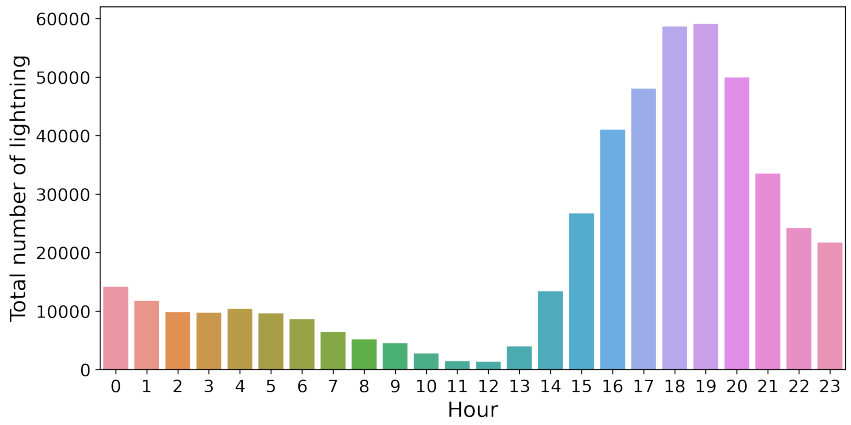
It is worth noting that humidity and temperature are key factors in the occurrence of lightning, and the Cerrado biome, with its higher humidity and temperature, may contribute to the higher lightning activity in the western region of the state.

Figure 3.7: Number of lightning occurrences though the months



Source: Developed by the author.

Figure 3.8: Number of lightning occurrences by hour

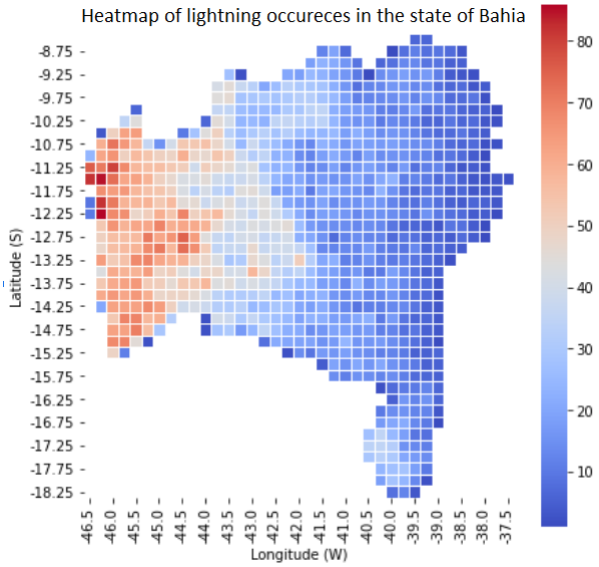


Source: Developed by the author.

3.2.2 Preprocessing

GDAS provides meteorological data in the Network Common Data Form (NetCDF) format, with each variable associated with a varying number of specific coordinates. For instance, a temperature variable may have latitude, longitude, and time coordinates, while other temperature variables may include an additional dimension of altitude above mean sea level. In total, the data consisted of 102 variables and 12 coordinates. In order to use this data with Pandas, we converted it to a format where each combination of variable and dimension was represented as a feature, resulting in a dataset with 355 features.

Figure 3.9: Density Heatmap of Lightning Strikes in Bahia throughout the Study Period



Source: Developed by the author.

In order to ensure that the right information is being fed to the deep learning model, every dataset has to undergo a quality check process. While minimal manipulation is required for the input dataset in this study, handling missing data is a crucial step. The final dataset comprises most of the available parameters, however, variables with missing data were discarded. A variable with missing data represented snow, was filled with zeros since snow is not expected in Bahia. The input dataset for the model included 332 meteorological variables.

The GDAS data were collected every six hours with grid cells at a resolution of 0.25º (approximately 27 km in lower and medium latitudes), for the whole Bahia territory, while the BrasilDAT gives a precise time, location and current of each lightning strike. In order to use the BrasilDAT information used for the target the data had to be manipulated to fit the same pattern as the meteorological data, which meant fitting into the same spatial grid with 0.25 degrees resolution and in the 6 hours time-frequency, while also adjusting the local time UTC-time, so both datasets can match. The target turned into categorical data to represent the presence or absence of lightning, thus bifurcated into those two classes. Subsequently, a one-hot encoding technique was applied to this categorical data.

As aforementioned, Figure 3.9 displays a heatmap illustrating lightning strikes in Bahia. Within this visual representation, each cell corresponds to a specific spatial grid resolution of the data, fixed at 0.25 degrees. This specific representation not only highlighting regions with varying frequencies of lightning occurrences but allows for a clear depiction of the spatial resolution employed in the study but also provides an informative view of the area coverage by the data. To further elaborate, the training as well as the predictions are made for the whole grid, encompassing areas not monitored by local weather stations.

This approach not only allows for predictions in those data-scarce areas but also ensures replicability and applicability to similar regions, even those outside the Bahia state.

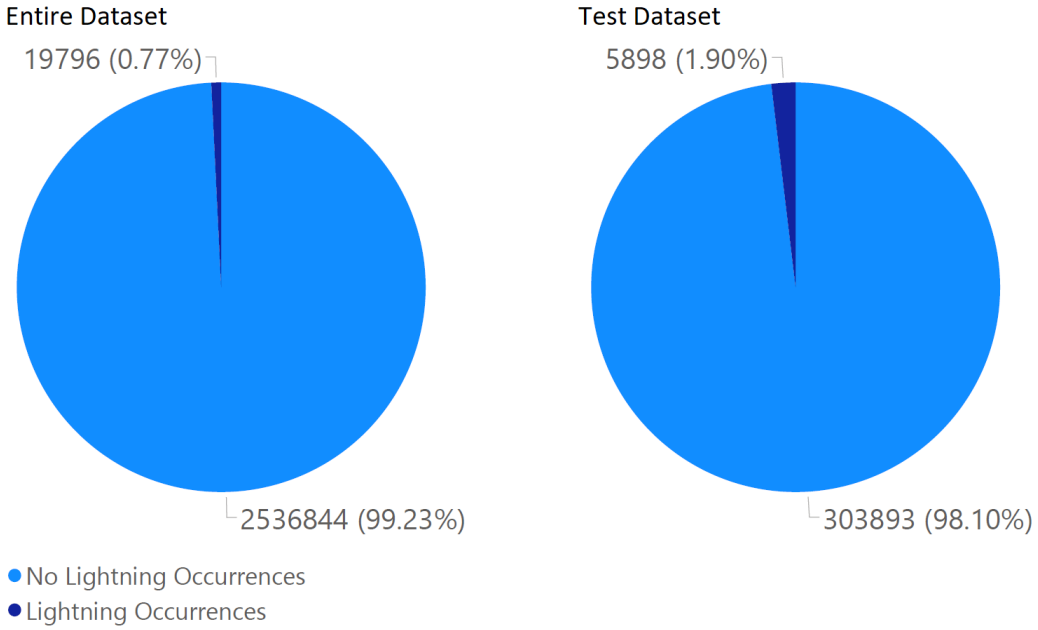
Supplying imbalanced data to a neural network can make it excessively difficult for the network to learn properly and operate as intended. This issue is particularly relevant for lightning data, as even during the rainy season, the moments when lightning occurs are significantly fewer than those when it does not, as noted by Alvares et al. (2013). To address this problem, a proportionate sampling approach was adopted to create the training and validation datasets. To elaborate, for each recorded instance of a lightning event at a specific grid location, another moment where no lightning event was recorded was randomly selected. This approach helped balance the dataset and provided more representative information to the neural network.

The test subset was created differently, consisting of 309,791 data points. Although it was ensured that this dataset contained 30% of randomly chosen lightning occurrence data, with the remaining 70% reserved for training and validation, a one-to-one proportion was not utilized. This dataset construction aimed to maintain a very similar unbalancing between lightning and no lightning instances perceived in the entire dataset into the test data, reflecting real-world scenarios. This approach aims to deal with the balance between providing a sufficient amount of test data that represents the real-world scenario for evaluation purposes while managing the computational costs associated with larger datasets. Thus, among these data points, 303,893 corresponded to lightning occurrences, while 5,898 represented instances of no lightning occurrences. Figure 3.10 illustrates this proportion between lightning and no lightning data distribution for the entire available period used in this study on the left side and for the test data on the right. Lightning occurrences accounted for only 0.77% of the entire dataset, while in the test dataset, it was 1.9%. Although this represents a very slight increase in the proportion, the dataset remains highly unbalanced, preserving the same order of magnitude of class unbalancing between both datasets.

As previously mentioned, the test subset consisted of 30% randomly chosen points of lightning occurrences data, thus leaving 70% to be divided by training and validation. The division process entailed allocating approximately 21% (which corresponds to 30% of the remaining 70%) for the validation set, while the remaining 49% was dedicated to the training set. This allocation strategy aimed to reserve a significant portion of the lightning data for training and validation while still maintaining a substantial amount for testing. The total amount of data, with both no lightning and lightning occurrences used for training and validation, was 19,126, and 8,197, respectively.

As aforementioned, to properly train a neural network such as RNNs and CNNs on temporal data, it is crucial to define a lookback window that provides enough historical context

Figure 3.10: Proportion between lightning and no lightning occurrences considering the entire (left) and the test (right) datasets



Source: Developed by the author.

for the model to understand the temporal aspect of the data. In this study, an 8-timestep lookback window was used, since each time step corresponds to 6 hours, the lookback total corresponds to 2 days. This procedure reduced the total number of samples, as any sample with missing data within this window could not be used as input in a neural network and was discarded. Careful consideration was given to ensure that the lookback window was appropriately generated to avoid mistaking random data points with a later point in time, thus maintaining the time series' quality.

Concurrently, the prediction horizon for this model is specifically set at a single step, or 6 hours, into the future. This configuration allows the model to project its prediction one timestep ahead, fitting the GDAS dataset's temporal frequency.

The final step of preprocessing is the scaling of the data between zero and one, that is a significant step, it ensures that all features are on the same scale, preventing certain features from dominating the others, which can lead to biased predictions, thus improving the convergence speed and the performance of the network. To verify unbiased evaluation, it is important to keep the test samples separated from the rest of the data during this phase. For that reason, the calculation of the minimum and maximum values is only performed on the training and validation sets, avoiding any contamination between the training and test data. The resulting normalization parameters are then used to scale the entire dataset, including the test set.

3.3 Deep Learning Models Development

When designing an ANN, the choice of its architecture is crucial. This involves determining the number of layers, the number of neurons per layer, activation functions, and hyperparameters. The selected architecture can significantly impact the network's performance, and identifying the optimal configuration for a specific problem can be challenging. In this study, we used an empirical testing methodology to select the optimal ANN architecture. The experiment conducted an iterative process of trial-and-error experiments to tune the architectures MLP, CNN, LSTM, GRU, and the hybrids CNN-LSTM and CNN-GRU. The process involved adjusting the number of layers, the number of neurons per layer, and other hyperparameters such as the learning rate and batch size. Thus, identifying the configuration that produced the best performance through the testing of various configurations on the target dataset. Figure 3.12 shows the final structures developed for each architecture.

During the process of training various neural network models, it became apparent that different architectures require varying numbers of epochs to converge on the loss graph. An epoch is a single pass through the entire training dataset, and convergence refers to the point where additional training does not significantly reduce the loss, indicating that the model has effectively learned the underlying patterns in the data. The method of trial and error was employed, closely monitoring the changes in the loss curve.

Some models responded differently to attempts to improve their performance, resulting in either overfitting or underfitting. By meticulously observing the loss graph's evolution across various numbers of epochs, the training process was fine-tuned, leading to more efficient and tailored model training the best balance achieved is represented by the loss curve in Figure 3.11.

The MLP model is designed with a total of eight layers. The initial layer is a Dense layer furnished with 512 neurons, using TanH as the activation function. A Dropout layer with a 0.5 dropout rate follows, randomly disabling 50% of the neurons during training to effectively mitigate the risk of overfitting.

This architecture then expands to include three subsequent Dense layers, each accommodating 256 neurons and utilizing the SELU as the activation function. To further the model's robustness, another Dropout layer is incorporated, retaining the 0.5 dropout rate, subsequent to the three Dense layers. Following this, a BatchNormalization layer is used to standardize the output of the preceding layer.

The architecture then grows with the addition of four Dense layers, each housing 128 neurons, continuing to utilize the TanH activation function. Post this integration, an

additional Dropout layer with a 0.5 dropout rate is included to boost the model's generalization capabilities. Subsequently, a Flatten layer is implemented to appropriately reshape the output from the prior layer.

The model then extends with the sequential addition of three Dense layers, each populated with 16 neurons. A SELU activation function is once again employed here, followed by yet another Dropout layer maintaining the 0.5 dropout rate. The model concludes with a final Dense layer consisting of two neurons, serving as the output layer for binary classification.

The model's compiling process leverages the Adam optimizer with a learning rate of 0.0001. As the output was encoded appropriately, the categorical cross-entropy loss function is selected, fitting the classification task at hand. The model is then trained utilizing a batch size of 256 across an extensive 1500 epochs, permitting the model to extract detailed patterns from the dataset.

The LSTM model initiates with a layer composed of 32 memory units. To regularize the model, this is followed by a Dropout layer featuring a rate of 0.5 and a BatchNormalization layer. Subsequently, an additional LSTM layer with 16 units is integrated into the model. This is once again accompanied by the same combination of Dropout and BatchNormalization layers for consistent regularization. A third LSTM layer, comprising another 16 units, is incorporated, succeeded by an additional Dropout layer. The model architecture culminates with a final Dense layer containing two neurons.

The model is compiled using the Adam optimizer, which operates with a particularly low learning rate of 0.00001 to ensure precise adjustments during the training process and the same loss function as the MLP. The model is then trained using a relatively large batch size of 256 to balance computation speed and gradient estimation accuracy. This process continues for an extended period, comprising 1500 epochs, to allow the model to thoroughly learn the patterns in the data.

The GRU model initiates with a layer comprised of 16 memory units. To enhance the model's overall performance, this is succeeded by a Dropout layer featuring a 0.3 dropout rate, as well as a BatchNormalization layer. An additional GRU layer with a scaled-down size of just 4 units is inserted, which is immediately followed by a second Dropout layer, this time with a reduced rate of 0.2. The final layer is a Dense layer, equipped with two neurons, which serves as the output layer dedicated to binary classification tasks. This refined structure offers an ideal balance between complexity and performance, ensuring the model effectively learns patterns in the data while avoiding overfitting. The training of this GRU model adopts the same optimizer, learning rate, loss function, and batch size as were used in the LSTM model. However, the training process extends over a longer period, encompassing a total of 2000 epochs.

The 1D CNN model initiates with a 1D convolutional layer furnished with 32 filters, utilizing the ReLu activation function to introduce non-linearity. This layer is succeeded by a MaxPooling1D layer with a pool size of 4, effectively reducing the spatial dimensions of the output. The model then incorporates a Flatten layer, transforming the multi-dimensional input into a one-dimensional format suitable for connection with subsequent dense layers. A Dropout layer is introduced next to prevent overfitting by randomly deactivating a portion of the neurons during the training process. The model culminates with a Dense layer, housing two neurons, serving as the output layer.

The configuration for the optimizer, learning rate, loss function, and batch size mirrors that of the previously described MLP model, ensuring a consistent approach to learning. An additional early stopping mechanism is integrated into this training process, terminating training when the model ceases to improve, thus allowing the training to conclude after 344 epochs.

The hybrid network, which integrates elements of both CNN and LSTM architectures, comprises seven layers. The initial layer is a 1D convolutional layer (Conv1D) with 256 filters, a kernel size of 1, and employs TanH as the activation function. Following this, a MaxPooling1D layer is integrated, reducing the spatial dimensions of the output by a factor of 4.

After the pooling operation, a BatchNormalization layer is added to standardize the outputs, followed by a Dropout layer with a rate of 0.6 to help mitigate overfitting. The architecture then incorporates an LSTM layer with 128 memory units, providing the model with the ability to recognize temporal dependencies. To further guard against overfitting, another Dropout layer with a rate of 0.5 is introduced. The architecture concludes with a final Dense layer, outfitted with two neurons, serving as the output layer.

The model's compilation parameters of optimizer, learning rate, loss function, and batch size, remain consistent with the configurations used in the previously described LSTM and GRU models. An additional early stopping mechanism is utilized, consequently, the training process concluded after a total of 525 epochs.

Lastly, The CNN and GRU hybrid network model commences with a 1D convolutional layer, characterized by a kernel size of 2 and 256 filters, employing the SELU as the activation function. To prevent neuron saturation, the layer is initialized with a bias initializer of zeros and a kernel initializer of He initializer, which sets the initial values for the biases and weights respectively.Following this, the model incorporates a MaxPooling1D layer with a pool size of 1 to condense the feature maps, subsequently using a BatchNormalization layer, and then a Dropout layer with a rate of 0.3.

The architecture proceeds with a second 1D convolutional layer, now furnished with 128 filters and a larger kernel size of 4, using ReLU as its activation function and same initializers as the first CNN1D layer (EKMAN, 2021). This layer is succeeded by another MaxPooling1D layer with a pool size of 4, preparing the data for introduction to the GRU layer.

The GRU layer follows, consisting of 128 units and employing TanH as its activation function. This layer features a kernel regularizer, that is a technique intended to improve generalization, adding a penalty to the loss function. L1 regularization calculates the sum of the absolute activation values, promoting sparsity by allowing some activations to become zero. L2 regularization calculates the sum of the squared activation values, encouraging small activation values in general (BROWNLEE, 2018).

The kernal regularizer was L1 and L2 regularizers with regularizes factors, which determines the strength that the regularization is applied, set on 1e-5 and 1e-4, respectively. It is also equipped with a bias regularizer with a specified L2 value of 1e-4, and an activity regularizer assigned with a L2 value of 1e-5. To conclude the architecture, a final fully-connected Dense layer incorporating two neurons is included. For binary classification tasks, the softmax activation function is applied to provide output probabilities.

The architecture concludes with a final fully connected Dense layer, populated with two neurons and utilizing the softmax activation function for outputting probabilities for the binary classification task.

The compilation parameters for the model align with those used in the previously described LSTM, GRU, and hybrid models. The early stopping mechanism is also in place, concluding the training process after 303 epochs when the model performance plateaus.

3.3.1 Validation Criteria and Final Model Selection

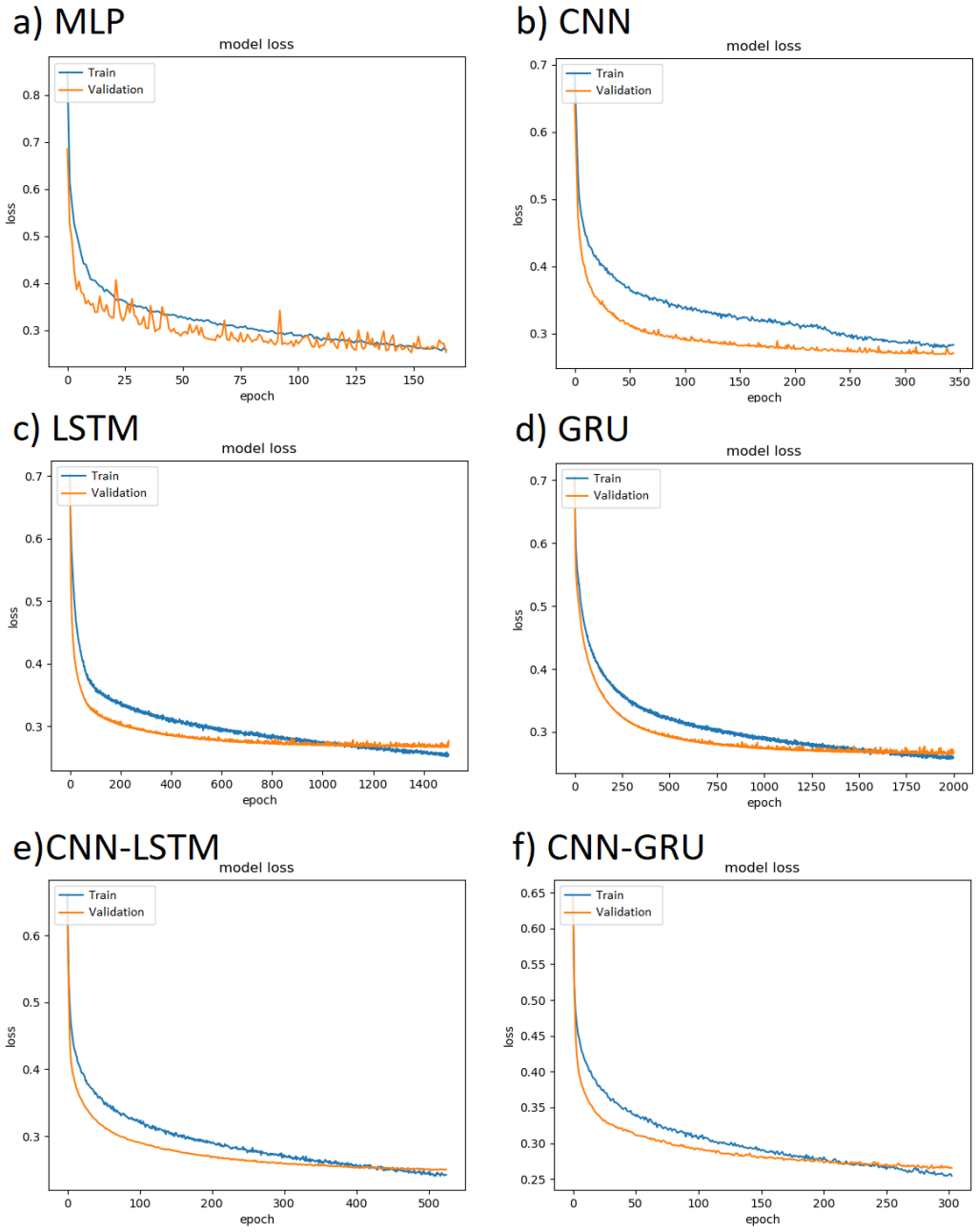
This study endeavored to assess the predictive power of various models on a chosen target variable, employing a series of carefully selected evaluation metrics to accomplish this goal. These metrics included Accuracy, Precision, Recall, F1-Score, AUC-ROC, and PRC-AUC. Comprehensive details about these metrics can be revisited in section 2.1.3.

To ensure a robust comparison across different models, the DeLong test, a method that is broadly used to compare two or more correlated binary classifiers on the basis of AUC-ROC, was employed. It provided a statistically significant assessment of their performances.

In the context of this study, the significance level (p-value) for the DeLong test was set to 0.05. This implies that if there is less than a 5% chance that the observed difference could occur purely by chance, it is considered to be statistically significant. This threshold is widely adopted in statistical analyses.

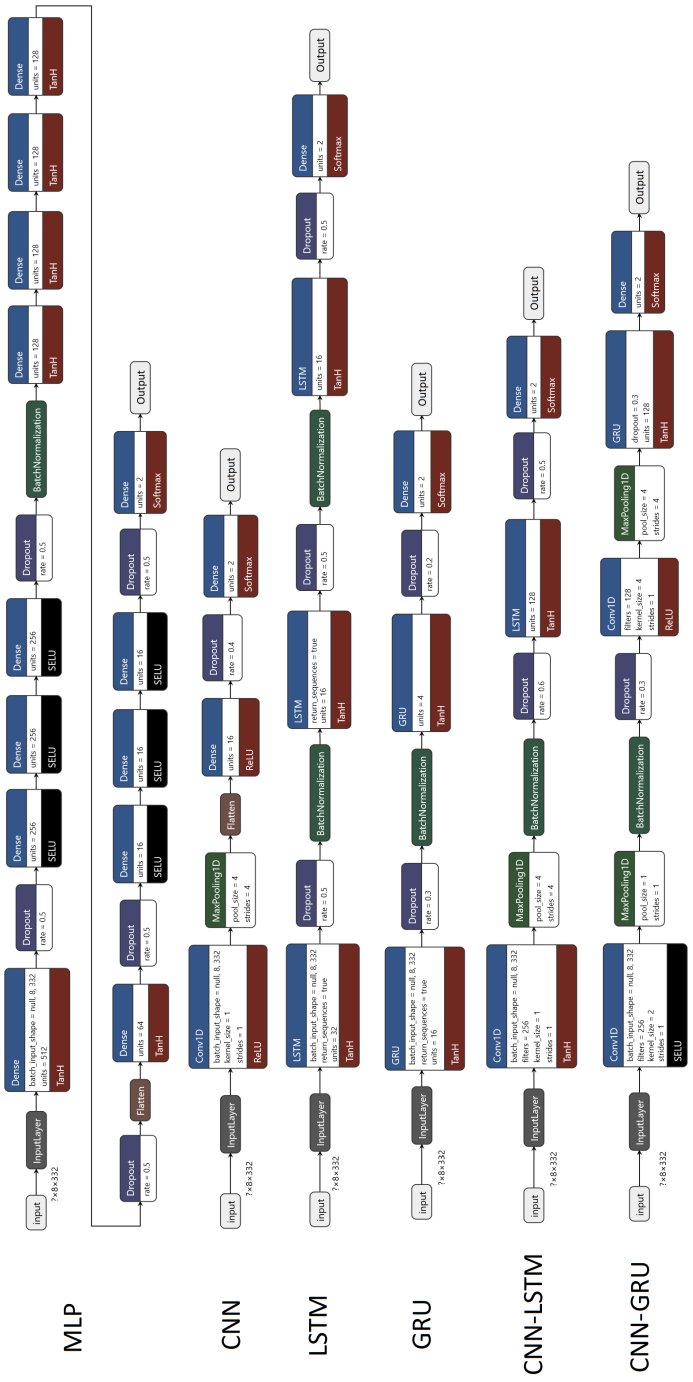
The specific implementation of the DeLong test used in this study was derived from the algorithm proposed in the paper by Sun e Xu (2014). This approach comprised several fundamental steps, which can be better understood through a thorough examination of the original paper by DeLong, DeLong e Clarke-Pearson (1988) and the subsequent work by Sun e Xu (2014). These references provide a comprehensive insight into the complexities of DeLong's method.

Figure 3.11: Loss Curves of The Neural Network Models - Panel (a) represents the loss curve for MLP, (b) for CNN, (c) for LSTM, (d) for GRU, (e) for CNN-LSTM, and (f) for CNN-GRU, respectively. The blue curve represents the training set, and the orange curve represents the validation set.



Source: Developed by the author.

Figure 3.12: Final Architectures for the Neural Network Models



Source: Developed by the author.

Chapter Four

Results and Discussion

4.1 Evaluation of models' performance

This section presents the metric results of the deep learning model's evaluation for predicting lightning activity up to 6 hours ahead. The performance analysis of various models and compare their statistical metrics. Additionally, the computational cost is associated with training and making predictions.

Figure 4.1 presents the normalized confusion matrix created based on the model's predictions. From the confusion matrix, it is evident that the models achieved a high hit rate in almost all cases, with 99.7% accuracy for the "No Lightning" class, except for the CNN, which recorded 99.6%. For the " Lightning" class, the models reached an accuracy ranging from 90.6% to 99.7%. The differences within the results are minimal; thus, the statistical metrics must be analyzed further to discern any significant variations.

Table 4.1 provides the metric values focusing on the "No Lightning Activity" class. Therefore, predicting the "Lightning" class correctly equates to predicting negative instances, thereby underscoring the significance of specificity as the key metric. Since, the specificity measures the proportion of actual "Lightning" instances, that are correctly identified.

All models demonstrated exceptional performance on the task, with F1-scores ranging from 0.9973 to 0.9981, indicating a strong balance between precision and recall.The F1 score is a particularly valuable metric when dealing with imbalanced datasets, as it considers both false positives and false negatives.

The models also exhibited high discriminative power, as evidenced by their robust ROC-AUC and PRC-AUC scores. The CNN-GRU model achieved the highest score of 0.9785 for the ROC-AUC, which reflects the ability to distinguish between positive and negative instances across various classification thresholds. While, all the models reached near-perfect score for PRC-AUC,ranging from 0.9982 to 0.9992, assessing the precision-recall trade-off. Consistently high scores in both metrics indicate excellent performance across different decision boundaries.

Precision values ranged from 0.9982 to 0.9992, and recall ranged from 0.9963 to 0.9970. Precision represents the proportion of true negatives among all predicted negatives, while recall measures the proportion of true negatives correctly identified. The highest specificity score was reached by the CNN-GRU model with 0.96, this metric specifically focuses

Table 4.1: Metrics results for the No Lightning activity class. Values in bold represent the best values for each metric.

Models	F1 Score	ROC AUC	PRC AUC	Precision	Recall	Specificity	Accuracy
MLP	0.9979	0.9650	0.9986	0.9987	**0.9970**	0.9330	0.9958
CNN	0.9973	0.9543	0.9982	0.9983	0.9963	0.9123	0.9947
LSTM	0.9978	0.9518	0.9981	0.9982	0.9974	0.9062	0.9956
GRU	0.9979	0.9769	0.9991	**0.9992**	0.9966	0.9573	0.9958
CNN-LSTM	0.9980	0.9764	0.9991	0.9991	0.9969	0.9559	0.9961
CNN-GRU	**0.9981**	**0.9785**	**0.9992**	**0.9992**	**0.9970**	**0.9600**	**0.9963**

on correctly predicting negative instances, which aligns with the objective of identifying "No Lightning Activity" cases accurately. The models achieved high scores in all these metrics, highlighting their ability to correctly identify negative instances.

Moreover, the models demonstrated remarkable accuracy in classifying instances into the "No Lightning Activity" class, with all models achieving accuracy scores ranging from 0.9947 to 0.9963, except for the CNN model with a score of 0.995.

The performance metrics for the "Lightning Activity" class are presented in Table 4.2. The key metric of interest in this context is recall, as correctly predicting the "Lightning" class corresponds to positive instances.

All models achieved near-perfect specificity scores, ranging from 0.9963 to 0.9974. The LSTM model stood out with the highest performance, followed by CNN-GRU(0.8704) and CNN-LSTM(0.8572), showcasing its superior capability in correctly classifying negative instances. In terms of accuracy, all models performed well, ranging from 0.9947 for the CNN model to 0.9963 for the others.

The GRU, CNN-LSTM, and CNN-GRU models exhibit the highest recall scores of 0.9573, 0.9559, and 0.96, respectively, along with ROC-AUC values ranging from 0.9764 to 0.9785. This indicates that these models are particularly effective at correctly identifying instances of the "Lightning" class and have excellent discriminative power. The high recall scores suggest a lower tendency to miss true "Lightning" instances.

In regards to precision, it indicated the ability to avoid false positives, i.e., misclassifying "No Lightning" instances as "Lightning." Among the models, the LSTM model demonstrates the highest precision score of 0.8704, with both hybrid models closely following, with precision scores of 0.8574 and 0.8604, signifying their effectiveness in reducing false positives.

Among the models, the CNN-GRU model achieved the top F1-score of 0.907 and PRC-

Table 4.2: Metrics results for the Lightning activity class. Values in bold represent the best values for each metric.

Models	F1 Score	ROC AUC	PRC AUC	Precision	Recall	Specificity	Accuracy
MLP	0.8941	0.9650	0.8022	0.8584	0.9330	0.9970	0.9958
CNN	0.8668	0.9543	0.7549	0.8256	0.9123	0.9963	0.9947
LSTM	0.8879	0.9518	0.7906	**0.8704**	0.9062	**0.9974**	0.9956
GRU	0.8969	0.9769	0.8085	0.8437	0.9573	0.9966	0.9958
CNN-LSTM	0.9039	0.9764	0.8203	0.8572	0.9559	0.9969	0.9961
CNN-GRU	**0.9074**	**0.9785**	**0.8267**	0.8604	**0.9600**	0.9970	**0.9963**

Table 4.3: Average Value of the Metric Results

Models	F1 Score	ROC AUC	PRC AUC	Precision	Recall	Specificity	Accuracy
MLP	0.9460	0.9650	0.9004	0.9286	0.9650	0.9650	0.9958
CNN	0.9320	0.9543	0.8766	0.9120	0.9543	0.9543	0.9947
LSTM	0.9428	0.9518	0.8944	**0.9343**	0.9518	0.9518	0.9956
GRU	0.9474	0.9769	0.9038	0.9214	0.9770	0.9770	0.9958
CNN-LSTM	0.9510	0.9764	0.9097	0.9282	0.9764	0.9764	0.9961
CNN-GRU	**0.9528**	**0.9785**	**0.9130**	0.9298	**0.9785**	**0.9785**	**0.9963**

AUC value of 0.827. This implies that it strikes a balance between precision and recall, indicating its effectiveness in capturing true "Lightning" instances while minimizing false positives.

It is worth noting that in situations where it is crucial to correctly identify every "Lightning" occurrence, prioritizing safety and preparedness, a model with a high recall for the "Lightning" class is preferred. Such a model excels at capturing the majority of true "Lightning" instances, even if it comes at the cost of misclassifying a few "No Lightning" instances as "Lightning". This prioritization reduces the risk of being unprepared for an actual lightning event.

Therefore, for tasks that require accurate detection of "Lightning" instances, such as weather monitoring systems or outdoor event management, the GRU model or one of the hybrid models, which demonstrate superior recall for the "Lightning" class, would be more suitable choices.

Table 4.3 presents the mean values for each metric. From this table, it becomes evident that the hybrid CNN-GRU model consistently outperforms in various metrics, followed closely by the CNN-LSTM and GRU models. However, there is an exception in the precision metric where the LSTM model outshines the rest. This is primarily due to its heightened precision in predicting the "Lightning Activity" class. For the "No Lightning activity" class, the other three models, again, exhibit superior performance

Table 4.4: Computational cost: Training and Inference Times for Each Model

Models	Training time	Inference time (Per data point)
MLP	50 min 34 s	1 milliseconds 627 µs
CNN	45 min 19 s	89 µs
LSTM	2 h 39 min 29 s	536 µs
GRU	2 h 2 min and 21 s	262 µs
CNN - LSTM	1 h 26 min 23 s	133 µs
CNN - GRU	1 h 20 min 7 s	156 µs

The obtained results align with the findings reported in the literature. In their study,Alves et al. (2021) achieved accuracy rates ranging from 82.14% to 86.49% when dealing with imbalanced data. However, by applying the SMOTE technique to balance the data, they were able to improve the accuracy to a range of 90.36% to 94.64%, slightly suppressing the accuracy found in this study in the second scenario.

Additionally,Mostajabi et al. (2019) also reported a lower accuracy rate of 76%. However, even though their study focused on ultra-short prediction horizons of only 30 minutes and attempted to predict lightning occurrences within a larger radius of 30 km.

When comparing the results of this study to the study conducted by Marope et al. (2023), it is observed that their best model was a logistic regression model, achieving a recall score of 93% and an ROC-AUC score of 90%. While, in this study, the CNN-GRU model, achieved the same ROC-AUC value of 98% for both classes. For the recall, when considering "No Lightning" instances as the positive class, it reached 100% and for the "Lightning" instances as the positive class 96%.

Consideration of computational cost is crucial in practical applications as it impacts the efficiency and usability of the models. Table 4.4 provides information on the computational cost for each model, including training and inference times. In terms of training times, the MLP and CNN models outperformed the LSTM and GRU models. The MLP and CNN models exhibited faster model convergence, possibly due to their simpler architectures. It is worth noting that the MLP model, despite having a greater depth and more neurons, still achieved faster convergence. On the other hand, the LSTM and GRU models required longer training times, likely due to their more complex structures, designed to capture long-term dependencies in the data.

The inference time was calculated with the total time it took to make the prediction upon the test dataset, and subsequently divided by the amount of data, the CNN model demonstrated the fastest prediction speed among the individual models. This efficiency can be attributed to the convolutional layers in the CNN architecture, which excel at parallel processing and feature extraction. Conversely, the MLP model showed the slowest

inference time, which can be attributed to both its fully connected characteristics and mainly due to its dept and number of neurons.

When considering the hybrid models, which combine different architectures, a reduction in both training and inference times is observed compared to their individual recurrent network components. This improvement can be attributed to the synergistic effects of combining the strengths of different models. The hybrid architectures leverage the efficiency of the CNN for feature extraction and the ability of the LSTM or GRU to capture temporal patterns.

In particular, among the models with preferable metric scores, such as the GRU and the hybrid models, the CNN-GRU model stands out with the lowest training time. For that reason, considering the computational cost alongside the performance metrics, the CNN-GRU model emerges as a compelling choice for real-time applications where both computational resources and prompt predictions are crucial. It is worth noting that, once usually numerical weather prediction models require much more computational power and hence time to perform their predictions, this solution provides an alternative yet viable approach that requires much less computational power to execute the forecasts, becoming a more energy-efficient and thus more sustainable solution.

4.2 Delong Test

Figure 4.2 represents the p-values and z-values obtained when comparing the developed models against each other. A positive z-score indicates that the observed difference is larger than the expected mean difference, while a negative z-score would indicate that the observed difference is smaller than the expected mean difference.

These z-scores obtained from the DeLong test indicate the magnitude of deviation between the observed differences and the expected differences under the null hypothesis. Larger absolute z-scores reflect more significant deviations, suggesting stronger evidence for differences between the compared groups. The p-value, on the other hand, represents the probability of observing the difference in performance between two models purely by chance, assuming that there is no true difference in their underlying performance. A low p-value suggests that the observed difference is unlikely to be due to chance and indicates evidence of a significant difference in performance between the models under comparison. Therefore, in general, as the z-value increases, the corresponding p-value tends to decrease. This is highlighted by the findings depicted in the Figure 4.2.

In hypothesis testing, if the p-value is below the chosen significance level (e.g., $p < 0.05$), we reject the null hypothesis in favor of the alternative hypothesis. This means that there

is a statistically significant difference between the compared models. Similarly, if the z-value exceeds the critical value associated with the chosen significance level (e.g., $z > 1.96$ for a 0.05 significance level in a two-tailed test), the null hypothesis is rejected. The critical value represents the threshold beyond which the observed difference is considered statistically significant.

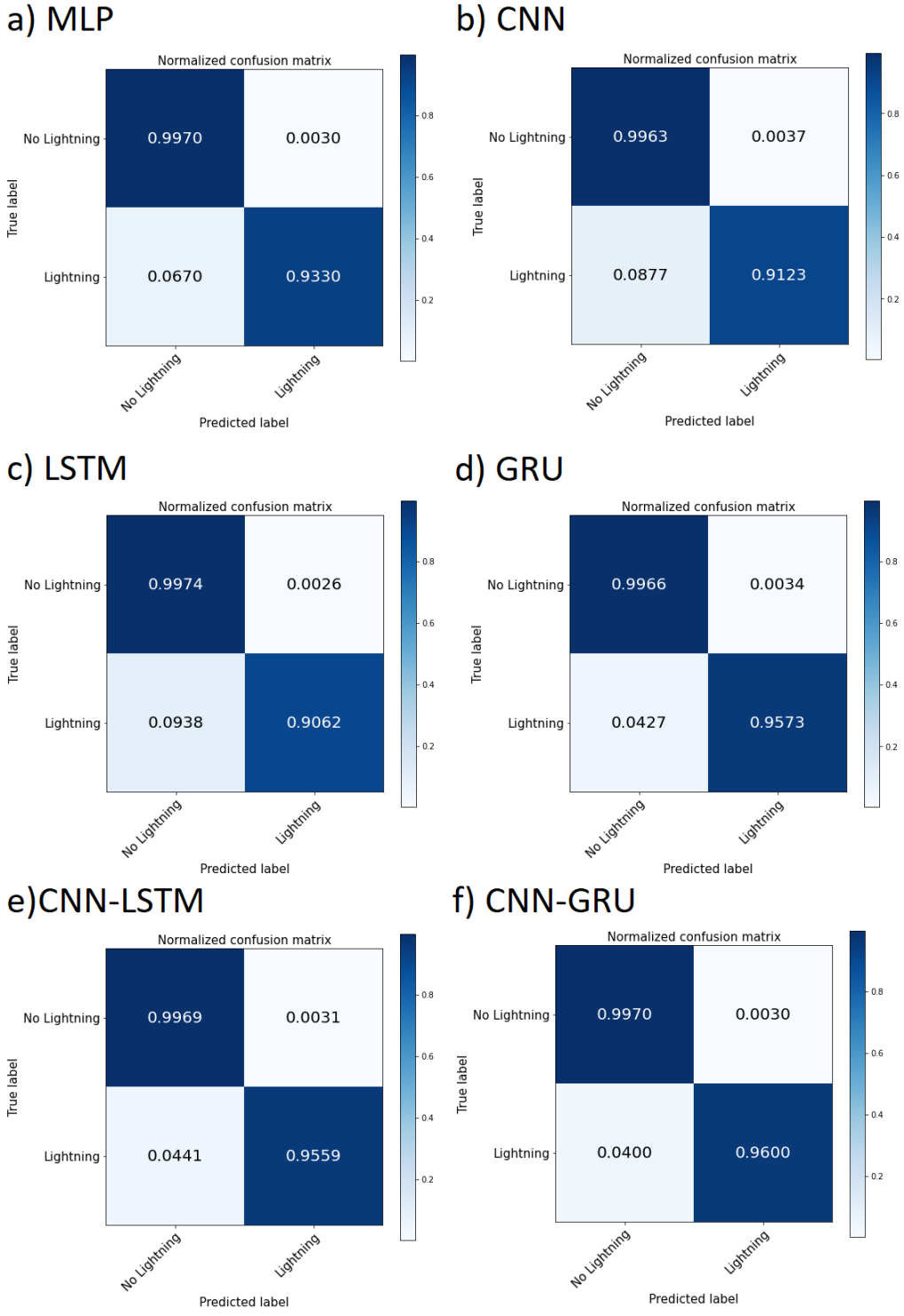
In the given results, the hybrid models (CNN-GRU and CNN-LSTM) showed the highest z-scores, surpassing 8.2727 in comparison to the other models, except for the comparisons between them and the GRU, and between themselves. These high z-scores indicate significant differences in performance between the hybrid models and the other models, such as MLP, CNN, and LSTM. However, the mentioned exceptions suggest that the performance differences between these specific pairs of models were less pronounced, indicating a relatively similar performance in those cases.

In the provided results, there were only three instances where the corresponding p-values exceeded the threshold of 0.05. This implies that the models being compared (LSTM with CNN, GRU with CNN-LSTM, and GRU with CNN-GRU) are not significantly different from each other.

Therefore, based on the Delong test results, it can be concluded that although the metrics scores presented in section 4.1 exhibit small ranges of variation, there are substantial disparities between the predictions of most of the models. Notably, the models which exhibit superior metric scores, the hybrid models, emerge as significantly different from each other but similar to the GRU.

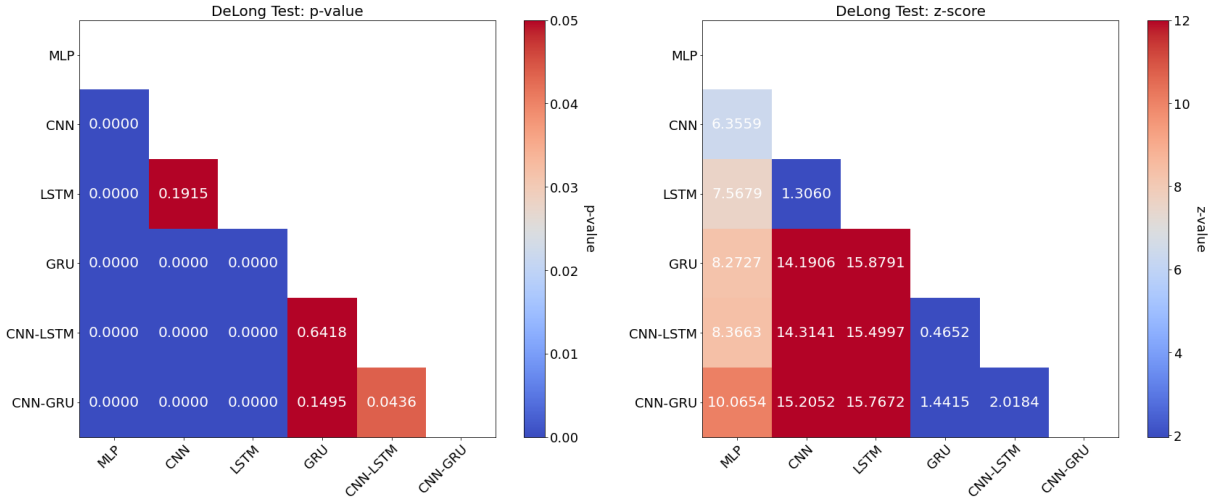
Hence, considering both the performance metrics and the computational cost, the CNN-GRU model is a compelling choice for real-time lightning forecasting applications where the superior accuracy, availability of computational resources and weather data, and prompt predictions are crucial factors to be considered. By opting for the CNN-GRU model, practitioners can strike a balance between superior performance and efficient resource utilization, making it an ideal candidate for real-time lightning forecasting applications that require both timely and accurate predictions through a more sustainable and energy-efficient approach.

Figure 4.1: Normalized Confusion Matrix The Neural Network Models - Panel (a) represents the loss curve for MLP, (b) for CNN, (c) for LSTM, (d) for GRU, (e) for CNN-LSTM, and (f) for CNN-GRU.



Source: Developed by the author.

Figure 4.2: DeLong Test:Statistical Comparison of Developed Models P-values and Z-values



Source: Developed by the author.

Chapter Five

Conclusion

5.1 Conclusion

This work explored the use of deep learning models for lightning prediction and evaluated their performance in predicting lightning activity up to 6 hours ahead. The results of the evaluation have provided valuable insights into the capabilities and limitations of different models in capturing key features and patterns related to lightning occurrence.

The models showed great overall performance predicting both the "No lightning activity" class and the "lightning activity" class. The GRU and hybrid models especially demonstrated superior performance across multiple metrics, highlighting the potential benefits of combining different architectures or techniques.

It is worth noting that even though variations in the metrics scores between the models may seem small, the Delong's test showed that the most models are indeed statistically significantly different from each other. Thus, while considering the overall performance across both classes, the CNN-GRU model in particular exhibited a consistently higher performance across various metrics while being statistically different from all the other models. These models showed a balanced ability to predict both "No Lightning" and "Lightning" instances accurately, making them well-suited for tasks that require accurate detection of lightning activity.

Furthermore, the evaluation of computational costs revealed that the MLP and CNN models had shorter training times, while the LSTM and GRU models required longer training times due to their complex architectures. In terms of inference time, the CNN model demonstrated the fastest prediction speed, while the MLP model had the longest inference time. The hybrid models generally showed reduced training and inference times compared to their individual components, highlighting the efficiency gained by combining different architectures.

Based on the performance metrics and computational cost considerations, the CNN-GRU model emerges as a promising choice for real-time applications that require accurate and prompt lightning predictions. Its balanced performance in detecting both "No Lightning" and "Lightning" instances, along with its relatively low training time and inference speed, make it a suitable model for weather monitoring systems.

In conclusion, this work makes a significant contribution to the field of lightning prediction

through its comprehensive evaluation of deep learning models that utilize globally available meteorological information as input. The findings from this research shed light on the effectiveness and potential of these models, to address the challenges posed by limited local data availability in certain regions. By leveraging advanced deep learning techniques and harnessing meteorological data on a global scale, this study opens up new avenues for facilitating lightning prediction and ultimately enhancing public safety in regions prone to lightning strikes, through a more energy-efficient and thus more sustainable approach.

Bibliography

ABDULLAH, N. H.; ADNAN, R.; RUSLAN, F. A. Lightning forecasting modelling using artificial neural network (ann): case study sultan abdul aziz shah airport or skypark subang. In: *IEEE Conference on Systems, Process and Control (ICSPC)*. Kuala Lumpur, Malaysia: [s.n.], 2018. p. 1–4. 2.3

ABREU, L. P. D.; GONÇALVES, W. A.; MATTOS, E. V.; ALBRECHT, R. I. Assessment of the total lightning flash rate density (frd) in northeast brazil (neb) based on trmm orbital data from 1998 to 2013. *International Journal of Applied Earth Observation and Geoinformation*, v. 93, p. 102195–102195, 2020. 1, 3.2.1

AGGARWAL, C. C. *Neural Networks and Deep Learning*. [S.l.]: Springer, 2018. v. 10. ISBN 978-3. 2.1.2.2, 2.1.2.2

ALMEIDA, H. A. *Climatologia Aplicada à Geografia*. Rua Baraúnas, 351 - Bairro Universitário - Campina Grande-PB - CEP 58429-500: Editora da Universidade Estadual do Paraíba, 2016. 2.2, 2.2.1, 3.1.1

ALVARES, C. A.; STAPE, J. L.; SENTELHAS, C. P.; GONÇALVES, J. L. M.; SPAROVEK, G. Köppen's climate classification map for brazil. *Meteorologische Zeitschrift*, v. 22, n. 6, p. 711–728, 2013. 3.2, 3.1.1, 3.3, 3.2, 3.2.2

ALVES, E.; LEAL, A.; LOPES, M.; FONSECA, A. Performance analysis among predictive models of lightning occurrence using artificial neural networks and smote. *IEEE Latin America Transactions*, v. 19, p. 755–762, 2021. 2.3, 4.1

ANA. *Mapa do Monitor de Secas*. accessed 2023–04–10. URL: https://monitordesecas.ana.gov.br/. 3.5

ARBIB, M. A. *Brains, Machines, and Mathematics*. Second. New York etc.: Springer-Verlag, 1987. XVI, 202 p. 63 figs. ISBN 3-540-96539-4. 2.1.2, 2.2, 2.3, 2.1.2

BRITO, Y. de; RUFINO, I.; BRAGA, C. et al. The brazilian drought monitoring in a multi-annual perspective. *Environmental Monitoring and Assessment*, Springer, v. 193, n. 31, 2021. Disponível em: <https://doi.org/10.1007/s10661-020-08839-5>. 3.1.2

BROWNLEE, J. *Long Short-Term Memory Networks With Python: Develop Sequence Prediction Models With Deep Learning*. [S.l.]: Machine Learning Mastery, 2017. 2.1.2.2

BROWNLEE, J. *Better deep learning: train faster, reduce overfitting, and make better predictions*. [S.l.]: Machine Learning Mastery, 2018. 2.1.1, 2.1.2.1, 2.1.2.1, 2.1.2.1, 2.6, 3.3

BROWNLEE, J. *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*. [S.l.]: Machine Learning Mastery, 2020. 2.1.1

CAI, S.; BILESCHI, S.; NIELSEN, E. *Deep Learning with JavaScript: Neural networks in TensorFlow.js*. Manning, 2020. ISBN 9781617296178. Disponível em: <https://books.google.com.br/books?id=N2dswgEACAAJ>. 2.1, 2.1.2.1

CHOLLET, F. *Deep learning with Python*. [S.l.]: Manning Publications, 2018. 2.1, 2.1, 2.1.1, 2.1.2.1, 2.1.2.2, 2.9, 2.1.2.2

COORAY, V. *An Introduction to Lightning*. Heidelberg: Springer, 2015. v. 201. 2.2

DELONG, E. R.; DELONG, D. M.; CLARKE-PEARSON, D. L. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, [Wiley, International Biometric Society], v. 44, n. 3, p. 837–845, 1988. ISSN 0006341X, 15410420. Disponível em: <http://www.jstor.org/stable/2531595>. 2.1.4, 3.3.1

DEMLER, O. V.; PENCINA, M. J.; D'AGOSTINO, R. B. S. Misuse of delong test to compare aucs for nested models. *Statistics in Medicine*, Wiley, v. 31, n. 23, p. 2577–2587, 2012. 2.1.4

DRAELOS, R. *Comparing AUCs of Machine Learning Models with DeLong's Test*. 2020. Website. Glass Box: Machine Learning and Medicine, by Rachel Draelos, MD, PhD. Disponível em: <https://glassboxmedicine.com/2020/02/04/comparing-aucs-of-machine-learning-models-with-delongs-test/>. 2.1.4

EKMAN, M. *Learning Deep Learning: Theory and Practice of Neural Networks Computer Vision. In Natural Language Processing, and Transformers Using TensorFlow*. [S.l.]: Addison-Wesley, 2021. 2.1.2.1, 2.1.2.2, 2.1.2.2, 2.1.2.2, 2.1.3, 3.3

ELSOM, D. M. *Lightning: Nature and Culture*. [S.l.]: Reaktion Books, 2015. 2.2.1

ESSA, Y.; AJOODHA, R.; HUNT, H. G. A lstm recurrent neural network for lightning flash prediction within southern africa using historical time-series data. In: *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. [S.l.: s.n.], 2020. p. 1–6. 2.3

ESSA, Y.; HUNT, H. G.; AJOODHA, R. Short-term prediction of lightning in southern africa using autoregressive machine learning techniques. In: *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. [S.l.: s.n.], 2021. p. 1–5. 2.3

ESSA, Y.; HUNT, H. G. P.; GIJBEN, M.; AJOODHA, R. Deep learning prediction of thunderstorm severity using remote sensing weather data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, v. 15, p. 4004–4013, 2022. 2.3

FAA. *FAQ: Weather Delay*. 2022. Last updated: Thursday, November 10, 2022; Accessed on 19/08/2023. Disponível em: <https://www.faa.gov/nextgen/programs/weather/faq>. 1

FOSTER, G.; GONÇALVES, R. *Delegação do Inter fica mais de 1h trancada em avião por conta de raios em Porto Alegre; outros 35 voos foram afetados*. 2023. Accessed on 24/08/2023. Disponível em: <https://g1.globo.com/rs/rio-grande-do-sul/noticia/2023/08/23/delegacao-do-inter-fica-mais-de-1h-trancada-em-aviao-por-conta-de-raios-em-porto-alegre-outros-35
ghtml>. 1

FURTADO, A. et al. Deep learning applied to chest radiograph classification - a covid-19 pneumonia experience. *Applied Sciences*, v. 12, n. 8, 2022. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/12/8/3712>. 2.1.4

GATEAU-REY, L.; TANNER, E. V.; RAPIDEL, B.; MARELLI, J.-P.; ROYAERT, S. Climate change could threaten cocoa production: Effects of 2015-16 El Niño-related drought on cocoa agroforests in Bahia, Brazil. *PloS one*, v. 13, n. 7, p. e0200454, 2018. 3.1.2

GENG, Y. A. et al. A heterogeneous spatiotemporal network for lightning prediction. In: *2020 IEEE International Conference on Data Mining (ICDM)*. [S.l.: s.n.], 2020. p. 1034–1039. 2.3

GIJBEN, M.; DYSON, L. L.; LOOTS, M. T. A statistical scheme to forecast the daily lightning threat over southern africa using the unified model. *Atmospheric Research*, v. 194, p. 78–88, 2017. 2.3

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016. 2.1, 2.1, 2.1.2, 2.1.2.2, 2.1.2.2

GULIYEV, H.; MUSTAFAYEV, E. Predicting the changes in the wti crude oil price dynamics using machine learning models. *Resources Policy*, v. 77, p. 102664, 2022. ISSN 0301-4207. Disponível em: <https://www.sciencedirect.com/science/article/pii/S030142072200112X>. 2.1.4

HAYKIN, S. *Neural networks and learning machines*. 3rd. ed. Hamilton: Pearson Education, Inc., McMaster University, 2009. 2.1.2, 2.1.2, 2.7, 2.1.2.2, 2.1.2.2

HUANG, J.; LING, C. X. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 17, n. 3, p. 299–310, 2005. 2.1.3

IBGE. *Cidades –Panorama da Bahia*. accessed 2023–04–10. URL: https://cidades.ibge.gov.br/brasil/ba/panorama. 3.1.1, 3.1.1

IBGE. *2019 Biomas e Sistema Costeiro-Marinho do Brasil - 1:250 000*. accessed 2023–04–13. URL: https://www.ibge.gov.br/geociencias/cartas-e-mapas/informacoes-ambientais/15842-biomas.html. 3.4

IBGE. *Biomas Continentais do Brasil*. accessed 2023–04–14. Disponível em: <https://geoftp.ibge.gov.br/informacoes_ambientais/estudos_ambientais/biomas/documentos/Sintese_Descricao_Biomas.pdf>. 3.1.1

Instituto Nacional de Pesquisas Espaciais. *Inpe avalia preju'izos causados por raios*. 2007. URL: <inpe.br/noticias/noticia.php?Cod_Noticia=936>. 1

K, V.; GAYATRI et al. Evaluation and usefulness of lightning forecasts made with lightning parameterization schemes coupled with the wrf model. *Weather and Forecasting*, v. 37, n. 5, p. 709–726, 2022. 2.3

KAPLAN, J. O.; LAU, K. H. K. The wglc global gridded lightning climatology and timeseries. *Earth System Science Data Discussions*, p. 1–25, 2021. 1

KILIÇARSLAN, S.; ADEM, K.; ÇELIK, M. *An overview of the activation functions used in deep learning algorithms*. [S.l.]: Tokat Gaziosmanpasa University, 2021. 75 - 88 p. 2.1.2.1, 2.1

KÖPPEN, W. Das geographische system de klimate. *Handbuch der klimatologie*, v. 1, p. 1–44, 1936. (document), 3.1.1, 3.3, 3.2

LIN, T. et al. Attention-based dual-source spatiotemporal neural network for lightning forecast. *IEEE Access*, v. 7, p. 158296–158307, 2019. 2.3

L'HEUREUX, M. Overview of the 2017-18 la niña and el niño watch in mid-2018. *Climate Prediction S&T Digest*, p. 97, 2019. 3.1.2

MAROPE, O.; TSHABALALA, B. G.; SCHUMANN, C.; HUNT, H. G. Johannesburg lightning nowcasting from meteorological data and electric field using machine learning. In: *2023 31st Southern African Universities Power Engineering Conference (SAUPEC)*. [S.l.]: IEEE, 2023. p. 1–6. 2.3, 4.1

MIAO, J.; ZHU, W. Precision–recall curve (prc) classification trees. *Evolutionary Intelligence*, Springer, 2021. 2.1.3

MOSTAJABI, A.; FINNEY, D. L.; RUBINSTEIN, M.; RACHIDI, F. Nowcasting lightning occurrence from commonly available meteorological parameters using machine learning techniques. *Npj Climate and Atmospheric Science*, v. 2, n. 1, p. 1–15, 2019. 2.3, 4.1

NACCARATO, K.; CARDOSO, M.; SANTOS, W. D.; CHAGAS, R. Monitoring lightning activity and other earth system variables from space using nanosatellite technology. In: *Proceedings of the International Conference on Grounding and Earthing and 5th International Conference on Lightning Physics and Effects*. [S.l.: s.n.], 2012. 3.2, 3.6

National Centers for Environmental Prediction/National Weather Service/NOAA/U.S. Department of Commerce. *NCEP GFS 0.25 Degree Global Forecast Grids Historical Archive*. 2015. Research Data Archive at the National Center for Atmospheric Research, Computational and Information Systems Laboratory. https://doi.org/10.5065/D65D8PWK. Accessed 26 Jan 2021. 3.2

OpenStreetMap contributors. *OpenStreetMap*. 2023. <https://www.openstreetmap.org>. Map data available under the Open Database License on the software Power BI. 3.1

PINTO JUNIOR, O. Thunderstorm climatology of brazil: Enso and tropical atlantic connections. *International Journal of Climatology*, v. 35, n. 6, p. 871–878, 2015. Disponível em: <https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/joc.4022>. 2.2.2

PINTO JUNIOR, O.; PINTO, C. d. A.; REGINA, I. *Brasil campeão mundial de raios*. São Paulo, SP: Artliber, 2021. 1, 2.2.1, 2.2.1, 2.2.2

PINTO JUNIOR, O.; PINTO, I. R. C. A. Brasildatdataset: Combining data from different lightning locating systems to obtain more precise lightning information. In: *International Conference on Lightning Detection*. Florida, USA: [s.n.], 2018. 3.2

RAHMAN, H.; AHMED, M. U.; BARUA, S.; FUNK, P.; BEGUM, S. Vision-based driver's cognitive load classification considering eye movement using machine learning and deep learning. *Sensors*, v. 21, n. 23, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/23/8019>. 2.1.4

RAKOV, V.; UMAN, M. *Lightning: physics and effects*. Cambridge: Cambridge university press, 2003. 2.2, 2.2.1

RONAGHAN, S. *Deep Learning: Overview of Neurons and Activation Functions*. 2018. <https://srnghn.medium.com/deep-learning-overview-of-neurons-and-activation-functions-1d98286cf1e4>. Accessed on May 1, 2023. 2.5, 2.1.2.1

SILVA, K. A.; ROLIM, G. de S.; VALERIANO, T. T. B.; MORAES, J. R. da Silva Cabral de. Influence of el niño and la niña on coffee yield in the main coffee-producing regions of brazil. *Theoretical and Applied Climatology*, v. 139, p. 1019–1029, 2020. 3.1.2

SUN, X.; XU, W. Fast implementation of delong's algorithm for comparing the areas under correlated receiver operating characteristic curves. *IEEE Signal Processing Letters*, v. 21, n. 11, p. 1389–1393, 2014. 2.1.4, 3.3.1

SZANDAŁA, T. Review and comparison of commonly used activation functions for deep neural networks. In: _____. *Bio-inspired Neurocomputing*. Singapore: Springer Singapore, 2021. (Studies in Computational Intelligence), cap. 11, p. 203–224. ISBN 978-981-15-5495-7. Disponível em: <https://doi.org/10.1007/978-981-15-5495-7_11>. 2.1.2.1, 2.1

The Royal Meteorological Society. *Types of Lightning*. 2017. <https://www.rmets.org/metmatters/types-lightning>. Accessed: 2023-05-30. 1, 2.11

TV Bahia. *Bahia é atingida por mais de 125 mil raios por causa de frente fria*. 2021. BATV news segment. Accessed on 27/05/2023. Disponível em: <https://globoplay.globo.com/v/10015798/>. 1

VUJOVI'c, Ž. Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, v. 12, n. 6, p. 599–606, 2021. 2.1.3

ZHOU, K.; ZHENG, Y.; DONG, W.; WANG, T. A deep learning network for cloud-to-ground lightning nowcasting with multisource data. *Journal of Atmospheric and Oceanic Technology*, v. 37, p. 927–942, 2020. 2.3

Short range lightning forecast based on deep learning models using globally available gridded meteorological data

Mirella Lima Saraiva Araujo

Salvador, October 2023.