INTEGRATING LOCALIZED DATA IN THE GRAPHCAST MODEL FOR HEAVY RAINFALL PREDICTIONS IN BRAZIL

Vinicius Ribeiro de Souza

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Daniel Ratton Figueiredo Ph.D.

Rio de Janeiro
Agosto de 2025

INTEGRATING LOCALIZED DATA IN THE GRAPHCAST MODEL FOR HEAVY RAINFALL PREDICTIONS IN BRAZIL

Vinicius Ribeiro de Souza

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Daniel Ratton Figueiredo Ph.D.

Aprovada por: Prof. Daniel Ratton Figueiredo
Prof. Gerson Zaverucha
Prof. Diego Parente Paiva Mesquita

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2025

Ribeiro de Souza, Vinicius

Integrating Localized Data in the Graphcast Model for Heavy Rainfall Predictions in Brazil/Vinicius Ribeiro de Souza. – Rio de Janeiro: UFRJ/COPPE, 2025.

XX, 75 p.: il.; 29, 7cm.

Orientador: Daniel Ratton Figueiredo Ph.D.

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2025.

Referências Bibliográficas: p. 69 – 72.

1. GraphCast. 2. Weather Forecasting. 3. Machine Learning. 4. XGBoost. 5. INMET. I. Ph.D., Daniel Ratton Figueiredo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

A dedicação do autor.

Agradecimentos

A conclusão deste trabalho só foi possível graças ao apoio incondicional de minha família e à orientação dedicada de meus professores.

Primeiramente, agradeço aos meus pais, que sempre me incentivaram a seguir o caminho do conhecimento e me ensinaram o valor da perseverança. À minha parceira e demais familiares, obrigado pelo carinho, paciência e compreensão em todos os momentos, especialmente nas longas horas de pesquisa e escrita. Sem o amor e o suporte de vocês, esta jornada seria infinitamente mais difícil.

Aos meus professores e orientadores, manifesto profunda gratidão pela confiança, pelas discussões enriquecedoras e pelo rigor acadêmico que norteou cada etapa deste trabalho. Suas críticas construtivas, sugestões técnicas e entusiasmo em compartilhar conhecimento foram essenciais para o desenvolvimento desta dissertação. Estendo meus agradecimentos a toda a banca examinadora e aos colegas do grupo de pesquisa, cuja colaboração e amizade tornaram este percurso mais leve e estimulante.

A todos vocês, meu sincero muito obrigado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

INTEGRATING LOCALIZED DATA IN THE GRAPHCAST MODEL FOR HEAVY RAINFALL PREDICTIONS IN BRAZIL

Vinicius Ribeiro de Souza

Agosto/2025

Orientador: Daniel Ratton Figueiredo Ph.D.

Programa: Engenharia de Sistemas e Computação

Alertas antecipados eficazes para chuvas intensas no sul do Brasil são dificultados pelas limitações de métodos de previsão isolados. Modelos meteorológicos globais, como o **GraphCast**, são poderosos, mas frequentemente não conseguem capturar a intensidade de tempestades localizadas. Por outro lado, modelos baseados apenas em dados históricos de estações meteorológicas locais conseguem prever tendências existentes, mas não são capazes de antecipar sistemas atmosféricos de larga escala em aproximação.

Esta dissertação preenche esta lacuna crítica ao introduzir uma arquitetura de fusão de dados inovadora que combina os pontos fortes de ambas as fontes de dados. O método proposto utiliza um modelo de aprendizado de máquina baseado em árvores para integrar previsões de médio alcance do **GraphCast** com observações horárias reais de 45 estações do Instituto Nacional de Meteorologia (INMET) no Rio Grande do Sul (RS). Essa abordagem captura tanto a dinâmica atmosférica de larga escala do modelo global quanto a variabilidade microclimática específica dos dados locais, criando um modelo preditivo especializado para cada localidade.

O desempenho da arquitetura demonstra uma melhoria substancial na capacidade preditiva. O modelo em Árvore utilizando fusão de dados alcançou um robusto índice ROC-AUC de 0,88, indicando forte poder preditivo. Isso contrasta fortemente com um modelo de referência treinado apenas com dados históricos das estações, que apresentou desempenho marginalmente superior à aleatoriedade, com um ROC-AUC de 0,56.

Esses resultados mostram que uma abordagem direcionada de fusão de dados não é apenas uma melhoria incremental, mas uma estratégia necessária para o desenvolvimento de um sistema de alerta precoce confiável e operacionalmente viável.

O estudo contribui com um fluxo de trabalho de modelagem que considera as especificidades das estações, um modelo por estação que captura a variabilidade local e uma arquitetura conteinerizada pronta para implementação em tempo real. Ao combinar com sucesso a capacidade de antecipação dos modelos globais com a precisão dos dados locais, este trabalho oferece uma estrutura escalável para aumentar a resiliência a inundações na América do Sul subtropical.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

INTEGRATING LOCALIZED DATA IN THE GRAPHCAST MODEL FOR HEAVY RAINFALL PREDICTIONS IN BRAZIL

Vinicius Ribeiro de Souza

August/2025

Advisor: Daniel Ratton Figueiredo Ph.D.

Department: Systems Engineering and Computer Science

Effective early warnings for heavy rainfall in southern Brazil are hindered by the limitations of using single forecasting methods. Global weather models, such as **GraphCast**, are powerful but often fail to capture the intensity of localized storms. Conversely, models relying solely on historical data from local weather stations can predict existing trends but cannot foresee large-scale atmospheric systems approaching the region.

This dissertation closes this critical gap by introducing a novel fusion framework that combines the strengths of both data sources. The proposed method utilizes a tree-based machine learning model to integrate medium-range forecasts from **GraphCast** with hourly ground-truth observations from 45 the Instituto Nacional de Meteorologia (INMET) weather stations in Rio Grande do Sul (RS). This approach captures both the large-scale atmospheric dynamics from the global model and the specific microclimate variability from local station data, creating a specialized predictive model for each location.

The framework's performance demonstrates a substantial improvement in predictive capability. The fusion model achieved a robust ROC-AUC score of 0.88, indicating strong predictive power. This stands in sharp contrast to a baseline model trained only on historical station data, which performed only marginally better than random chance with a ROC-AUC of 0.56.

These results show that a targeted data fusion approach is not just an incremental improvement, but a necessary strategy for developing a reliable and operationally viable early-warning system. The study contributes a gauge-aware fusion workflow, a one-model-per-station design that captures local variability, and a containerized architecture ready for real-time deployment. By successfully combining the foresight

of global models with the precision of local data, this work provides a scalable framework to enhance flood resilience in subtropical South America.

Contents

List of Figures

List of Tables

Lista de Símbolos

Lista de Abreviaturas

API	Application Programming Interface 29
AWS	Amazon Web Services 3
CDS	Copernicus Climate Data Store 28
CNN-LSTM	Convolutional Long Short-Term Memory 8
ECMWF	European Centre for Medium-Range Weather Forecasts 1
ERA5	Fifth Generation ECMWF Reanalysis 2
GNN	Graph Neural Network 2, 10, 25
GRIB	GRIdded Binary 30
HRES	High-Resolution Forecast 1
IDW	Inverse Distance Weighting 32
IFS	Integrated Forecasting System 1
INMET	Instituto Nacional de Meteorologia 1, 3
ML	Machine Learning 3
NWP	Numerical Weather Prediction 1
PCA	Principal Component Analysis 37
PoA	Porto Alegre 58
RS	Rio Grande do Sul 1
SACZ	South Atlantic Convergence Zone 1
TPU	Tensor Processing Unit 8
XGBoost	Extreme Gradient Boosting 3

Chapter 1

Introduction

1.1 Motivation for the Problem: Extreme Weather Events, Socio-Economic Losses, and Forecasting Challenges

Extreme weather events—particularly heavy rainfall and attendant flooding—rank among the most destructive natural hazards, producing escalating economic losses, infrastructure damage, and loss of human life. In subtropical South America, and especially in Brazil's state of Rio Grande do Sul, such events are driven by a confluence of frontal systems and Mesoscale Convective Complexes that develop along the South Atlantic Convergence Zone (ipc, 2022). The May 2024 floods in Rio Grande do Sul exemplified this threat, causing widespread devastation and illustrating the *operational* gap between forecast lead time and actionable response.

Notwithstanding a dense network of rain gauges operated by the Brazilian National Meteorological Institute (INMET) , alerts are frequently issued *after* thresholds have already been breached. Models trained solely on these station records offer high-fidelity confirmation of rainfall once it is underway, yet provide limited predictive skill beyond a few hours because they cannot "see" approaching synoptic systems. Eliminating this foresight deficit while preserving local accuracy is therefore a central motivation of the present work.

Traditional forecasting centres depend on physics-based Numerical Weather Prediction such as the (ECM, 2025) Integrated Forecasting System (Eur, 2023a) and its High-Resolution configuration (Eur, 2023b). These models solve the discretised Navier–Stokes equations with sophisticated data assimilation, achieving high accuracy yet demanding hours on leadership-class supercomputers. This computational burden constrains the frequency and latency of updates—limitations laid bare when emergency managers require guidance on sub-hour time scales. Moreover, the same

physical core underpins the widely used *ERA5*(HERSBACH *et al.*, 2020) reanalysis dataset (ERA5), whose known moist biases can adversely propagate into downstream flood-risk models.

GraphCast as a Disruptive Alternative. Advances in end-to-end machine learning now offer a data-driven counterpart that preserves forecast skill while slashing inference cost. DeepMind's *GraphCast* (LAM *et al.*, 2023) employs a on a multi-resolution icosahedral mesh to propagate atmospheric dependencies, generating 6-hourly forecasts up to ten days ahead *orders of magnitude* faster than the operational High-Resolution Forecast (HRES). Benchmark studies report that GraphCast reduces 24-h global root-mean-square error relative to the ERA5 analysis by 13 % and retains useful skill 24 h longer than HRES across multiple variables (LAM *et al.*, 2023). Follow-up evaluations over East Asia further show lower positional error for tropical-cyclone tracks compared with HRES (YAN *et al.*, 2024). These verified gains, coupled with a three-order-of-magnitude decrease in inference cost, motivate the adoption of GraphCast as the global backbone of this thesis.

The Predictive Gap: Fusing Global Foresight with Local Precision. However, these data-driven models are not a panacea when used in isolation. The 25 km resolution of GraphCast, while impressive for a global model, can lead to a 'blurring' effect, averaging out the sharp, localized rainfall peaks that are the primary drivers of flash floods. Conversely, relying solely on historical station data results in a model that is fundamentally limited; it can only extrapolate past local trends and is blind to developing large-scale synoptic systems. This creates a critical predictive gap: one model has foresight but lacks precision, while the other has ground-truth precision but lacks foresight. The core hypothesis of this thesis is that by fusing these two complementary data sources, we can create a predictive model that is superior to either component alone.

Forecasting Extreme Events. From a statistical perspective, extremes constitute the tail ($<10\%$) of the rainfall distribution, inducing severe class imbalance and sharply penalizing missed detections. Robust prediction therefore hinges on learning algorithms that maximize recall while controlling false alarms through cost-sensitive objectives. In Brazil, civil-defence protocols flag cumulative 48-hour precipitation beyond 30 mm as a threshold for emergency actions (Sec, 2024), underscoring the need for classifiers explicitly tuned to such cut-offs.

1.2 Contributions of this study

- **Data Fusion and Pre-processing.** Developed a two-stage pipeline aligning GraphCast's 0.25° forecasts with 45 stations via bilinear interpolation (50 km radius) and 6-hour temporal aggregation, implemented in `xarray` and `pandas`.
- **Feature Engineering and Target Definition.** Constructed a rich predictor set spanning surface and pressure-level variables, and derived a binary target using a rolling 48-hour window with threshold grid $\theta \in \{20, 30, \ldots, 90\}$ mm—framing a forward-look classification aligned with Brazilian Civil-Defence triggers.
- **Model Training with Tree-Based Ensembles.** Trained classifiers (CHEN and GUESTRIN, 2016) with `scale_pos_weight` to address class imbalance; hyper-parameters were tuned by Optuna's TPE sampler over 50 trials, guided by the F1-score in nested cross-validation (AKIBA *et al.*, 2019).
- **Evaluation Protocol.** Employed time-series rolling-origin cross-validation (TASHMAN, 2000) on April–May 2024 data, reporting precision, recall, F1, accuracy, and ROC AUC across thresholds without temporal leakage.
- **Operational Architecture.** Containerised the workflow with Docker, executed GraphCast inference on RunPod(run) (NVIDIA A100), and persisted artefacts in S3(Ama, 2025)—reducing forecast latency from hours to minutes and paving the way for real-time forecasting and alerts.

Table 1.1: Core datasets, tools, and libraries employed in the proposed framework. References for each row: GraphCast Forecasts: (LAM *et al.*, 2023); INMET Stations: (INSTITUTO NACIONAL DE METEOROLOGIA); ML Library: XGBoost: (CHEN and GUESTRIN, 2016); Optimiser: Optuna (TPE): (AKIBA *et al.*, 2019); Infrastructure: RunPod GPU(run), AWS S3(Ama, 2025).

Component	Description
GraphCast Forecasts	6-hourly global fields at 0.25°
INMET Stations	45 gauges across RS, hourly precipitation
ML Library	XGBoost ensemble trees
Optimiser	Optuna (TPE) for hyper-parameter tuning
Infrastructure	RunPod GPU, AWS S3 storage

1.3 Organization of the Thesis

- Chapter 2 surveys the theoretical landscape, mixing physics-based NWP (e.g., ECMWF IFS/HRES) with cutting-edge surrogates such as PanguWeather

(BI *et al.*, 2023) and GraphCast, and reviews tree-based ensemble methods for imbalanced classification.

- Chapter 3 details data acquisition, spatial–temporal fusion, feature engineering, target construction, and the cross-validated training pipeline encapsulated within a scalable cloud architecture.

- Chapter 4 presents empirical results for RS during April–May 2024, including threshold-sensitivity analyses and station-level diagnostics.

- Chapter 5 interprets these findings in light of operational requirements, outlines limitations (e.g., spatial coverage, imbalance at extreme thresholds), and recommends future enhancements such as synthetic oversampling and multi-gauge zoning, finally it synthesizes the contributions and charts forward directions for ML-enabled flood resilience in Brazil.

Chapter 2

Background and Foundational Concepts

This chapter establishes the scientific context and foundational concepts for the research presented in this thesis. We begin by outlining the significant challenges in forecasting extreme weather events, contrasting traditional physics-based models with modern data-driven approaches. We then introduce **GraphCast**, a state-of-the-art machine learning model, as the predictive engine for this work. Subsequently, we detail the **INMET dataset**, which provides the essential ground-truth meteorological observations for Brazil. Finally, we present our proposed **tree-based classification framework**, motivating its selection and detailing the advanced techniques required to handle the inherent challenges of this task, such as severe class imbalance. This chapter methodically builds the argument for our approach, connecting the forecasting tool, the ground-truth data, and the analytical method into a cohesive research strategy.

2.1 The Landscape of Weather Forecasting: From Physics to Data

The prediction of extreme weather events, particularly heavy rainfall, remains a critical challenge in meteorology due to their profound societal and economic impacts BAUER *et al.* (2015). Research in this field has historically been dominated by physics-based models, but is increasingly shifting towards data-driven and machine learning (ML) techniques. This section outlines this evolution, establishing the context for the advanced methods employed in this thesis.

2.1.1 The Physics-Based Paradigm: Numerical Weather Prediction (NWP)

For decades, Numerical Weather Prediction (NWP) has been the default choice of modern weather forecasting, exemplified by the European Centre for Medium-Range Weather Forecasts (ECMWF) Integrated Forecasting System (IFS). These models simulate the future state of the atmosphere by numerically solving complex partial differential equations that govern fluid dynamics and thermodynamics. Driven by immense advancements in computational power, data assimilation, and atmospheric physics, NWP models have achieved remarkable improvements in forecast skill and generate the critical ERA5 reanalysis dataset used to train many advanced ML models BAUER *et al.* (2015).

Despite these successes, NWP models face inherent limitations in predicting extreme weather events. These phenomena, such as intense localized rainfall, are often driven by small-scale convective processes that global models struggle to resolve explicitly due to their relatively coarse resolution SHEPHERD (2017). Consequently, NWP forecasts can exhibit significant spatial and temporal biases, particularly in capturing the peak intensity of extreme precipitation. Furthermore, achieving the high resolutions necessary for more accurate simulation is computationally intensive, requiring hours of supercomputer processing time. This high computational cost fundamentally limits their applicability for rapid, real-time warnings BAUER *et al.* (2015).

2.1.2 The Rise of Data-Driven Forecasting

The convergence of vast meteorological datasets, such as ECMWF's ERA5, with advancements in high-performance computing has catalyzed a paradigm shift towards data-driven forecasting. Early ML applications focused on statistical post-processing of NWP outputs to correct for systematic biases PRICE *et al.* (2024). A significant body of subsequent research has targeted **precipitation nowcasting** (0–2 hour forecasts), where deep learning models like Convolutional LSTMs have excelled at learning the advection and evolution of precipitation fields from radar imagery SHI *et al.* (2015).

A more recent and challenging frontier is the application of ML to **medium-range forecasting** (3-10 days). End-to-end deep learning models such as Pangu-Weather BI *et al.* (2023) and, central to this thesis, GraphCast LAM *et al.* (2023), have demonstrated remarkable performance. These models can match or even surpass the accuracy of state-of-the-art NWP systems for many standard variables while being orders of magnitude faster at inference.

Recent studies focusing on regional performance, such as evaluating GraphCast

over mainland China YAN *et al.* (2024), make clear the growing usage of this model for localized validation—a perspective this thesis extends to the Brazilian context. The strengths and weaknesses of these competing paradigms are summarized in Table 2.1.

2.1.3 Hybrid Fusion versus Fine-Tuning of Global Models

Fine-tuning large global weather networks (e.g., GraphCast) on small regional datasets is tempting, yet recent evidence shows marginal accuracy gains at prohibitive cost. BI *et al.* (2023) report that re-training just the final block of the 852-million-parameter Pangu-Weather on a single region demands more than 1 PFLOP·s and still suffers from over-fitting within three epochs. Likewise, RASP *et al.* (2022) found that transfer-learning a 300-million-parameter generative model on Europe improved RMSE by only 3 % while increasing inference latency eightfold due to the expanded parameter set and I/O burden. In operational early-warning contexts where latency directly affects lead time, such slow-downs offset the modest skill gains.

By contrast, **hybrid fusion** strategies leave the global model untouched and train lightweight local post-processors—often linear models, decision trees, or shallow neural networks—on station observations. This paradigm has repeatedly shown competitive performance: S"ONDERBY *et al.* (2020) fused MetNet nowcasts with radar observations achieving a 12 % CRPS reduction, while SCHEUERER *et al.* (2023) demonstrated that gradient-boosted trees trained on gauge data reduce bias and sharpen probabilistic precipitation forecasts in the western United States. Crucially, these hybrids require orders of magnitude fewer parameters ($<$10 M) and run in milliseconds, making them far more suitable for resource-constrained early-warning systems.

Collectively, the literature supports our decision to adopt a *data-fusion* approach—integrating GraphCast outputs with INMET observations via a lightweight tree-based classifier—instead of attempting to fine-tune GraphCast itself.

2.2 GraphCast: The Predictive Engine

The predictive engine for this research is GraphCast, a state-of-the-art deep learning model for medium-range global weather forecasting developed by LAM *et al.* (2023). The justification for its selection is rooted in a transformative combination of computational efficiency and proven forecast skill, two attributes that are essential for the goals of this thesis.

First, GraphCast's operational speed represents a paradigm shift. By operating

Table 2.1: A comparative summary of NWP and ML approaches for weather forecasting, highlighting their respective strengths and limitations in the context of extreme events.

Approach	**Strengths**	**Limitations**
NWP (e.g., ECMWF IFS)	High physical consistency; robust for large-scale patterns.	Coarse resolution for local extremes; high computational cost.
ML Nowcasting (e.g.,)	High accuracy for short-term (0–2h) forecasts; computationally efficient.	Limited to short time horizons; heavily dependent on real-time radar data.
ML Medium-Range (e.g., GraphCast)	Extremely fast inference; high accuracy for standard variables; proven skill in severe event tracking.	Susceptible to "blurring" of extremes and inherent data imbalance—the central challenges this thesis aims to address.

on a single Google v4 device, it produces a full 10-day forecast in under a minute. This stands in stark contrast to traditional NWP systems, such as the ECMWF's High-Resolution Forecast (HRES), which require hours of supercomputer time to solve discretized physical equations (e.g., Navier-Stokes).

This dramatic reduction in inference time is not merely an efficiency gain; it directly enables the primary objective of this research. Because the early-warning classifier can ingest fresh GraphCast fields within minutes, alerts for severe weather events can be issued with a lead time that was previously unattainable.

Furthermore, benchmark experiments by LAM *et al.* (2023) show that GraphCast consistently outperforms the ECMWF High-Resolution Forecast (HRES) on multiple verification metrics, including lower tropical-cyclone track error (Figure 2.1). Taken together, superior skill and orders-of-magnitude faster inference make GraphCast the state-of-the-art choice for the global driver of our fusion framework.

Also, this inference speed does not come at the cost of accuracy, particularly for the high-impact events central to this study. The efficacy of this data-driven approach is demonstrated by its superior performance in forecasting severe weather systems. As shown in Figure 2.1, GraphCast consistently achieves a lower tracking error for tropical cyclones compared to the operational HRES model, particularly at longer lead times.

This proven ability to accurately capture the dynamics of extreme phenomena provides the necessary confidence that GraphCast's raw forecast data is a robust

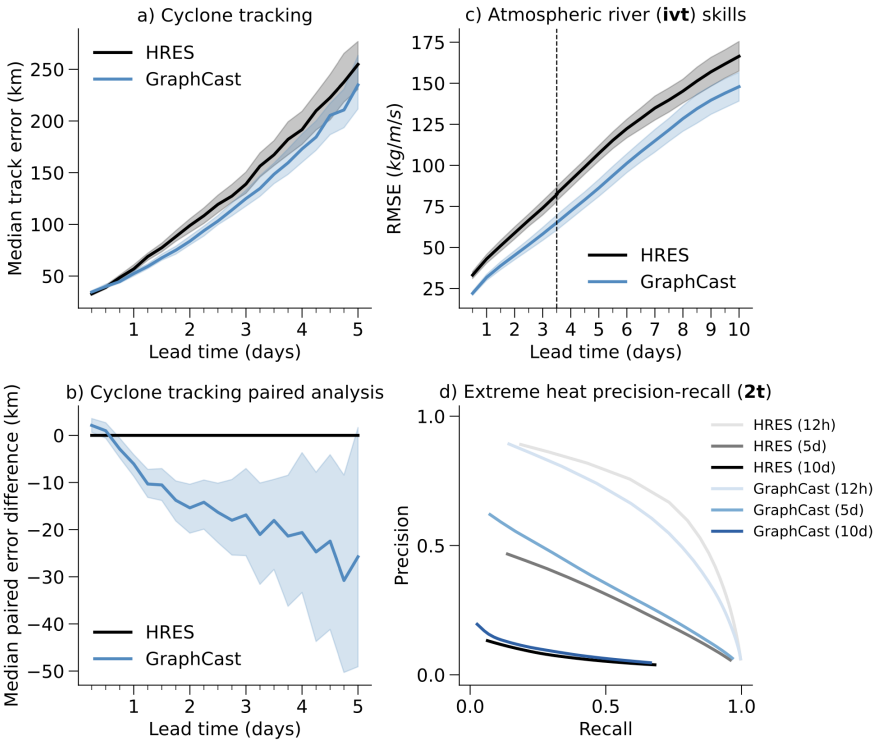


Figure 2.1: Forecast skill comparison between GraphCast (blue) and the operational HRES NWP model (grey). The plot shows the Root Mean Square Error (RMSE) in kilometers for predicting the track of tropical cyclones. Lower values indicate higher accuracy. GraphCast consistently outperforms HRES, particularly for lead times beyond 3 days, demonstrating its superior capability in forecasting the trajectory of critical severe weather systems. Extracted from (LAM *et al.*, 2023).

and reliable foundation upon which to build a specialized classification model. It is this unique synthesis of rapid inference and validated scientific skill that makes this model the ideal engine for this research.

2.3 GraphCast Architecture: An Encoder-Processor-Decoder Pipeline

GraphCast's architecture is best understood as a sophisticated Encoder-Processor-Decoder pipeline, a design common in advanced machine learning models. This structure systematically transforms input data into a latent representation, processes it to learn complex dynamics, and decodes it back into a physical forecast. Figure 2.2 provides a high-level schematic of this data flow, which forms the organizing principle for the detailed discussion that follows.

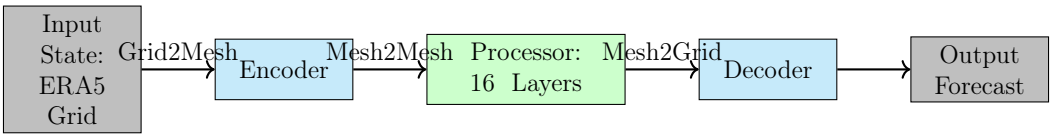


Figure 2.2: Schematic of the GraphCast pipeline. The model ingests gridded atmospheric data, encodes it onto a latent graph representation, processes it through iterative message-passing, and decodes it back to a grid to produce a forecast.

2.3.1 Input Data: The ERA5 Atmospheric State

GraphCast is trained on the ECMWF ERA5 reanalysis dataset (1979–2017), a comprehensive global record integrating observations with NWP models that serves as the ground truth for training (HERSBACH *et al.*, 2020). The input to the model at each time step is a detailed snapshot of the Earth's atmospheric state defined on a regular $0.25° \times 0.25°$ latitude-longitude grid, comprising $721 \times 1440 = 1,038,240$ grid points.

For each grid point i, the model uses two previous time steps (t and $t-1$) to predict the next state ($t+1$). The input feature vector is a concatenation of dynamic weather variables, external forcing terms, and static geophysical features:

$$f_i^{\text{input}} = [x_i^{t-1}, x_i^t, f_i^{t-1}, f_i^t, f_i^{t+1}, c_i]$$

where the components include:

- **Weather states** x_i^{t-1}, x_i^t: These vectors contain 5 surface variables (e.g., 2-meter temperature) and 6 atmospheric variables (e.g., geopotential, specific

humidity) distributed across 37 vertical pressure levels. A complete enumeration of these variables is provided in Appendix B, Table B.1.

- **Forcing terms** $f_i^{t-1}, f_i^t, f_i^{t+1}$: Time-varying analytical features like total solar irradiance.
- **Static features** c_i: Time-invariant data such as a land-sea mask and surface geopotential.

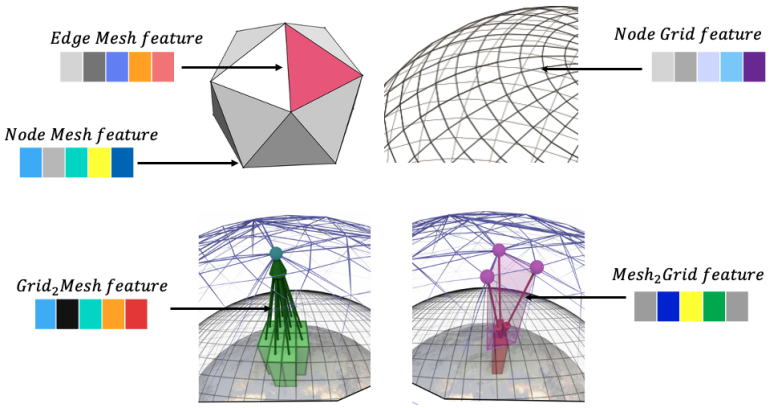


Figure 2.3: Visual summary of the feature vector composition for the input grid nodes (left) and the internal mesh nodes (right), illustrating the diverse data types that define the atmospheric state at each point.

2.3.2 The Encoder: Mapping the Grid to an Icosahedral Mesh

The first stage of the pipeline, the Encoder, addresses a fundamental challenge of global modeling: the singularity and grid distortion at the poles of standard latitude-longitude grids. To circumvent this, GraphCast projects the input grid data onto a more uniform **multi-resolution icosahedral mesh**.

This mesh is constructed by recursively subdividing the 20 triangular faces of a base icosahedron, as illustrated in Figure 2.4 and Figure 2.5. After six refinement levels, this process yields a high-resolution mesh with 40,962 nodes and near-uniform 25 km spacing (Table 2.2).

The Encoder's function is to transform the features from the input grid nodes ($\mathcal{V}^G$) to the latent mesh nodes ($\mathcal{V}^M$).

It does this using a dedicated "Grid2Mesh" Graph Neural Network. First, features on the grid, mesh, and the edges connecting them are embedded using Multi-Layer Perceptrons (MLPs). Then, a message-passing step updates the mesh node

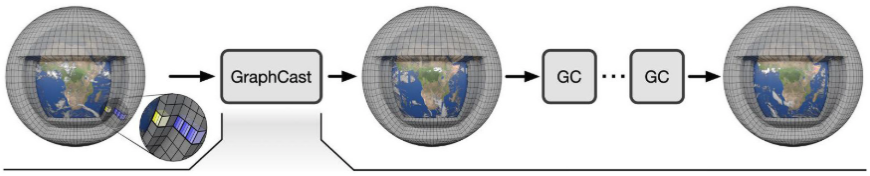


Figure 2.4: The multi-resolution icosahedral mesh used by GraphCast. The refinement process starts with a base icosahedron (left) and iteratively subdivides its faces to create a high-resolution, quasi-uniform grid covering the globe (right), thus avoiding polar singularities. Extracted from (LAM *et al.*, 2023).

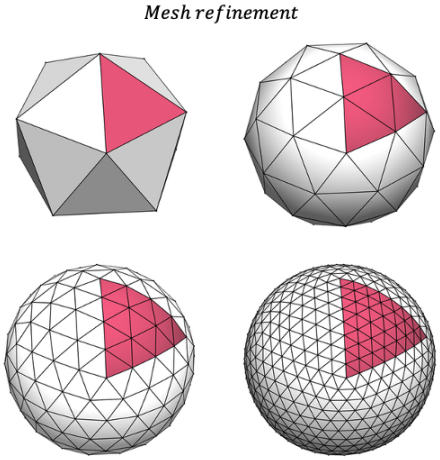


Figure 2.5: A conceptual view of the mesh refinement process, where each triangular face is split into four smaller triangles, progressively increasing the resolution of the global representation.

Table 2.2: Statistics of the multi-resolution icosahedral mesh as a function of the refinement level R. The final mesh used by the Processor (R=6) contains 40,962 nodes.

Refinement	0	1	2	3	4	5	6
Num Nodes	12	42	162	642	2,562	10,242	40,962
Num Faces	20	80	320	1,280	5,120	20,480	81,920

features based on information from the connected grid nodes. This can be expressed as:

$$e_{ij}^{G2M\prime} = \text{MLP}_{\text{Grid2Mesh}}^{G2M}([v_i^G, v_j^M, f_{ij}^{e^{G2M}}]), \tag{2.1}$$

$$v_j^{M\prime\prime} = v_j^{M\prime} + \text{MLP}_{\text{Grid2Mesh}}^{M\prime\prime}\left(\sum_{i \in \mathcal{N}(j)} e_{ij}^{G2M\prime}\right). \tag{2.2}$$

Here, Equation 2.1 updates the edge embedding by combining features from the source grid node v_i^G, target mesh node v_j^M, and original edge features. Equation 2.2 then updates the target mesh node by aggregating messages from all incoming edges, using a residual connection for training stability. Figure 2.6 provides a visual representation of this update process.

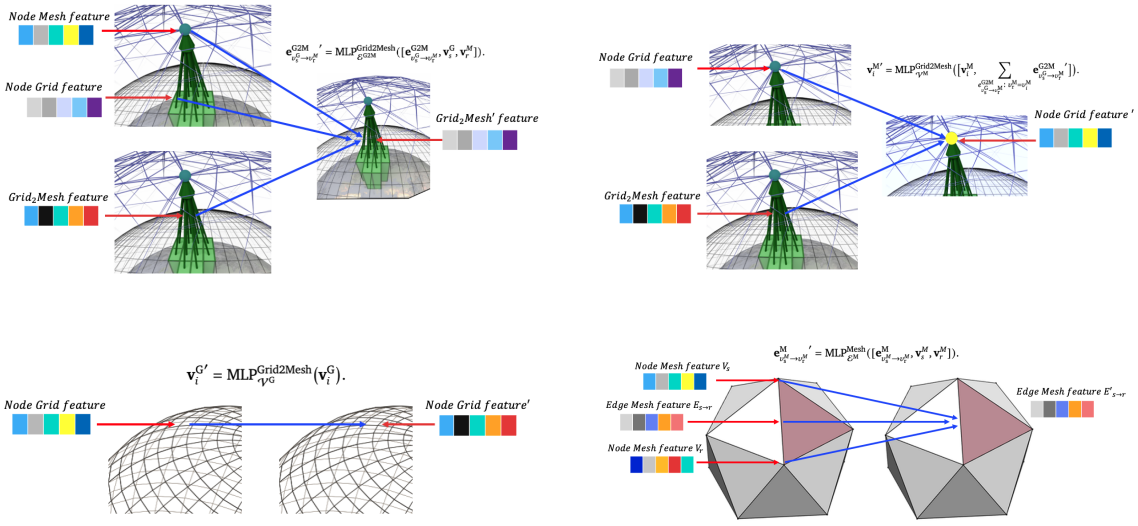


Figure 2.6: Visualizing the Grid2Mesh message passing in the Encoder, corresponding to Equations 2.1 and 2.2. The process involves updating edge features by combining node information (top row), followed by updating the target mesh node by aggregating these new edge messages (bottom row).

2.3.3 The Processor: Learning Dynamics via Message Passing

The core of GraphCast's predictive power resides in the Processor. This component is a deep GNN with 16 sequential message-passing layers that operates exclusively on the icosahedral mesh graph. Its purpose is to simulate the evolution of the atmospheric state by learning complex, non-local interactions between different points on the globe.

Each layer in the Processor updates the state of every mesh node by aggregating information from its neighbors, a mechanism illustrated conceptually in Figure 2.7.

A simplified formulation of this update for a node v at layer $l+1$ is:

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \sum_{u \in \mathcal{N}(v)} h_u^{(l)} + b^{(l)}\right)$$

where $h_u^{(l)}$ is the feature vector of a neighboring node u, $W^{(l)}$ and $b^{(l)}$ are learnable parameters, and σ is a non-linear activation function. This iterative process allows information to propagate across the graph, enabling the model to learn planetary-scale dependencies like the formation of atmospheric rivers.

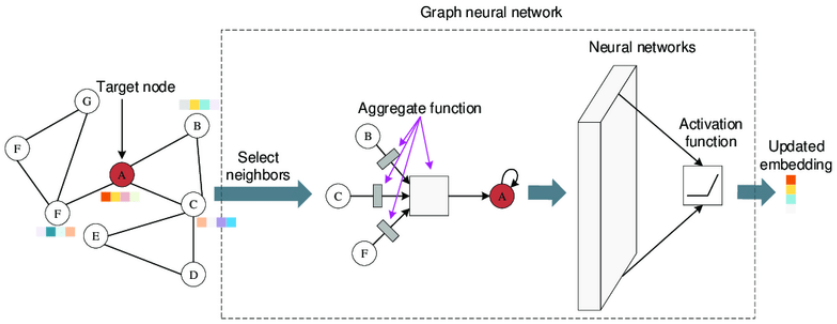


Figure 2.7: The conceptual message-passing mechanism at the heart of the GraphCast Processor. To update its own feature vector (center node), a node aggregates information from its neighbors on the icosahedral mesh.

The full implementation in the Processor updates both node and edge features within each of its 16 layers, enabling richer, more expressive representations of atmospheric interactions:

$$e_{ij}^{M2M''} = \text{MLP}_{\text{Mesh}}^{M2M}([e_{ij}^{M2M'}, v_i^{M'}, v_j^{M'}]), \tag{2.3}$$

$$v_i^{M''} = \text{MLP}_{\text{Mesh}}^{M''}\left([v_i^{M'}, \sum_j e_{ji}^{M2M''}]\right). \tag{2.4}$$

2.3.4 The Decoder: Projecting Back to a Global Forecast

After the 16-layer Processor has produced an updated latent state on the mesh, the Decoder's task is to project this information back onto the standard 0.25° latitude-longitude grid. This "Mesh2Grid" GNN effectively translates the learned dynamics from the model's internal graph representation into a physically interpretable weather forecast. The Decoder calculates a residual (the predicted change in the weather state), which is then added to the input state x^t to produce the forecast for

the next time step, x^{t+1}. The core decoding operations are:

$$e_{ij}^{M2G'} = \text{MLP}_{\text{Mesh2Grid}}^{M2G}([e_{ij}^{M2G'}, v_i^M, v_j^G]), \tag{2.5}$$

$$v_i^{G''} = \text{MLP}_{\text{Mesh2Grid}}^{G''}\left([v_i^{G'}, \sum_j e_{ji}^{M2G'}]\right), \tag{2.6}$$

$$\hat{y}_i^G = \text{MLP}_{\text{output}}(v_i^{G''}). \tag{2.7}$$

The final predicted state is then computed simply as $x^{t+1} = x^t + \hat{y}$.

2.3.5 Generating a Forecast: Autoregressive Rollouts

The Encoder-Processor-Decoder pipeline described above constitutes a one-step learned simulator, which we can denote as ϕ. It predicts the state at $t + \Delta t$ given the states at t and $t - \Delta t$:

$$\hat{x}^{t+\Delta t} = \phi(x^t, x^{t-\Delta t}) \tag{2.8}$$

To generate a long-range forecast, GraphCast applies this learned simulator iteratively in an auto-regressive fashion, as depicted in Figure 2.8. The prediction for one step becomes the input for the next, allowing the model to "roll out" a forecast for up to 10 days.

$$\hat{x}^{t+1:t+T} = \left(\phi(x^t, x^{t-1}), \phi(\hat{x}^{t+1}, x^t), \ldots, \phi(\hat{x}^{t+T-1}, \hat{x}^{t+T-2})\right) \tag{2.9}$$

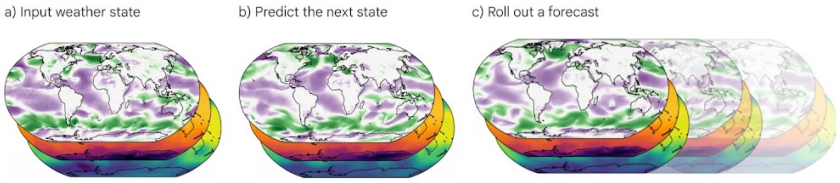


Figure 2.8: The auto-regressive forecast generation process. The one-step GraphCast predictor uses two prior states to predict the next state. This new state is then fed back as input to produce the subsequent prediction, creating a rollout forecast over an extended time horizon. Figure from LAM *et al.* (2023).

2.3.6 Performance and Critical Analysis

GraphCast's performance, when benchmarked against the operational ECMWF HRES model, demonstrates the power of this architecture. As shown in Table 2.3, it achieves superior or competitive scores on key metrics like Z500 RMSE and 2m Temperature ACC.

Table 2.3: Illustrative performance comparison of GraphCast versus the HRES model for key meteorological variables and metrics. Data sourced from (LAM *et al.*, 2023).

Metric	Lead Time	GraphCast	HRES
Z500 RMSE (m)	5 days	210	245
2m Temp ACC	7 days	0.85	0.81
Precip Skill	3 days	0.62	0.65

However, a critical evaluation reveals several limitations that motivate the research in this thesis:

- **Data Dependency and Bias**: As GraphCast is trained exclusively on ERA5 data, it may inherit and amplify any systematic biases present in the reanalysis dataset, particularly for rare events or in regions with sparse historical observations.
- **Generalization to Extremes**: While skilled at tracking large-scale systems, its performance on the sharp, localized peaks of extreme events (e.g., intense convective rainfall) is less established, often suffering from a "blurring" effect common to models trained with MSE-based loss functions.
- **Interpretability**: As a deep learning model, the GNN's internal decision-making lacks the explicit physical constraints of NWP models, making error analysis and diagnosis more challenging.
- **Fixed Resolution**: The 25 km mesh resolution is static and cannot adapt to resolve critical sub-grid scale processes, such as the formation of individual thunderstorms, which are often responsible for extreme precipitation.

These limitations, particularly the challenges in accurately representing and classifying localized extreme events, form the central problem domain that this thesis seeks to address by building a specialized classification framework on top of the raw GraphCast forecast outputs.

2.4 Observational Benchmark: The INMET Ground-Truth Dataset

The preceding analysis established GraphCast as a powerful, state-of-the-art forecasting engine, yet also highlighted its inherent limitations as a purely data-driven model: a dependency on potentially biased training data, a tendency to "blur"

extreme values, and uncertain generalization to rare events. To address these challenges and anchor our predictive framework in physical reality, a rigorous, independent observational benchmark is required. For the Brazilian context, the dataset provided by the Instituto Nacional de Meteorologia (INMET), the national meteorological service, provides this authoritative ground-truth baseline.

2.4.1 Data Source and Collection Network

INMET operates an extensive monitoring network across Brazil's vast and climatically diverse 8.5 million km² territory, encompassing both automatic weather stations (EMAs) and conventional weather stations (EMCs). As detailed by the World Meteorological Organization, this network consisted of over 800 stations as of 2020, with the more than 500 EMAs forming the backbone of the nation's real-time monitoring system (WORLD METEOROLOGICAL ORGANIZATION, 2021). The hourly data streams from these automatic stations are indispensable for resolving the lifecycle of short-lived, intense convective systems that often produce extreme rainfall. The strategic distribution of these stations, shown in Figure 2.9, ensures comprehensive spatial representation across Brazil's biomes, from the tropical Amazon to the temperate south.

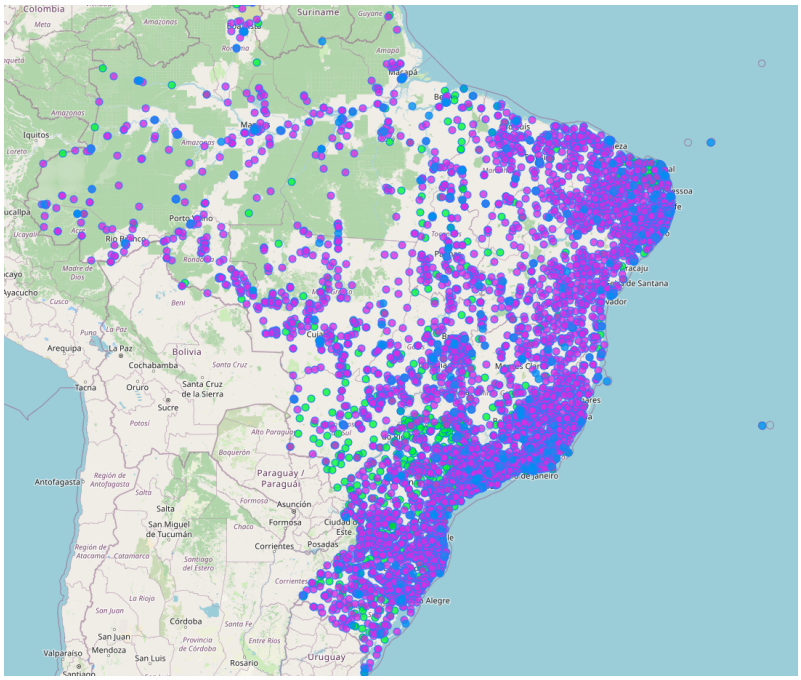


Figure 2.9: The distribution of INMET's meteorological monitoring network across Brazil. Green dots represent Automatic Stations (EMAs), blue dots are Conventional Stations (EMCs), and purple dots are Automatic Rain Gauges. This dense network provides the high-quality, point-based observations used as ground truth in this study. Source: INSTITUTO NACIONAL DE METEOROLOGIA - MAPS (2025).

2.4.2 Data Characteristics and Variables

INMET's stations record a standardized set of surface meteorological variables essential for studying extreme precipitation. As summarized in Table 2.4, the primary variable for this thesis is **total precipitation (mm)**, recorded hourly by the EMAs.

This point-based, hourly measurement provides the "sharp" ground-truth signal of actual rainfall, against which the spatially averaged GraphCast forecasts will be evaluated. The dataset is subject to rigorous quality control procedures (XAVIER *et al.*, 2023) and is made publicly available through INMET's online portal (INSTITUTO NACIONAL DE METEOROLOGIA, 2025), adhering to open data principles.

Table 2.4: Key meteorological variables from the INMET dataset relevant to this study.

Variable	**Unit**	**Relevance to Study**
Total Precipitation	mm	Primary target variable for defining extreme rainfall events, measured hourly for high temporal resolution.
Temperature	°C	Provides contextual data on thermodynamic conditions influencing precipitation patterns.
Relative Humidity	%	Indicates atmospheric moisture levels, critical for understanding rainfall intensity.

Table 2.5: Summary of INMET dataset quality and characteristics.

Aspect	**Description**
Data Quality Control	INMET employs cross-validation with ranked statistics to ensure data accuracy, particularly for precipitation and temperature measurements (XAVIER *et al.*, 2023).
Temporal Resolution	Hourly data from EMAs, with daily and monthly aggregates available from EMCs (MUSAH *et al.*, 2022).
Spatial Coverage	Over 800 stations as of 2020, covering diverse climatic zones across Brazil (INSTITUTO NACIONAL DE METEOROLOGIA, 2025).

2.5 A Tree-Based Framework for Extreme Event Classification

Tree-based classification algorithms constitute a versatile family of supervised learning methods that recursively partition the predictor space into homogeneous regions and assign class labels according to majority vote or probability estimates within each region. Individual decision trees are prized for their interpretability and their ability to model complex, non-linear interactions without requiring extensive feature engineering.

However, to improve predictive accuracy and robustness, modern practice typically employs ensemble variants—such as Random Forests, Gradient Boosting Machines, XGBoost, and LightGBM—that aggregate many weak learners. By averaging or sequentially correcting individual trees, these ensembles reduce variance and bias, capture high-order feature interactions, and naturally accommodate mixed data types.

Moreover, they offer built-in mechanisms—feature-level sampling, regularization parameters, class-balanced loss functions—to mitigate common challenges like overfitting and class imbalance. As a result, tree-based ensembles have become a standard, high-performing choice for classification tasks across diverse scientific and industrial domains.

2.5.1 Motivation for Tree-Based Ensembles

Tree-based models for classification have compelling reasons. A single Decision Tree, the conceptual building block of this approach, partitions the feature space through a series of hierarchical, human-readable rules, as shown in Figure 2.10. This inherent interpretability is scientifically valuable, offering the potential to extract insights into the meteorological drivers of extreme events (QUINLAN, 1993).

However, single decision trees are prone to high variance and tend to overfit complex data. To overcome this, we employ **ensemble methods**, which combine the predictions of multiple trees to create a more robust and accurate model. Two principal ensemble strategies are particularly relevant:

- **Random Forests**, which build a multitude of decorrelated trees in parallel on bootstrap samples of the data and average their predictions, reducing variance and improving generalization (BREIMAN, 2001).

- **Gradient Boosting Machines (GBM)**, which build trees sequentially, where each new tree is trained to correct the residual errors of the preceding ensemble. This iterative approach is exceptionally powerful for capturing

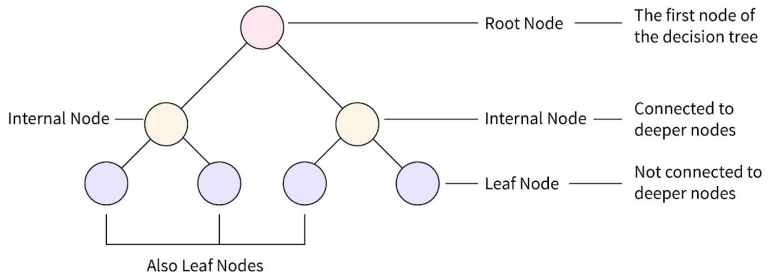


Figure 2.10: A conceptual illustration of a Decision Tree classifier. Each internal node represents a test on a feature (e.g., a GraphCast-forecasted variable like humidity or pressure), and each leaf node represents a class label ('Extreme Event' or 'No Event'). The path from the root to a leaf constitutes a classification rule.

complex patterns in structured data (FRIEDMAN, 2001).

2.5.2 State-of-the-Art Implementations: XGBoost and LightGBM

For this research, we leverage two advanced, highly optimized implementations of gradient boosting that have become the standard for high-performance machine learning on large-scale datasets: XGBoost and LightGBM.

XGBoost (Extreme Gradient Boosting) enhances traditional GBM with innovations such as second-order Taylor approximations for the loss function, integrated L1 and L2 regularization to control model complexity, and parallelized tree construction. It builds trees in a level-wise fashion, ensuring balance but at a higher computational cost (CHEN *et al.*, 2004).

LightGBM (Light Gradient Boosting Machine) is optimized for speed and memory efficiency. It employs a leaf-wise growth strategy, which often converges to a better solution faster. Its key innovations include Gradient-based One-Side Sampling (GOSS) to focus on instances with large gradients and Exclusive Feature Bundling (EFB) to reduce feature dimensionality, making it exceptionally fast for large meteorological datasets (KE *et al.*, 2017). A comparison of their key characteristics is provided in Table 2.6.

Table 2.6: Comparison of key features in the XGBoost and LightGBM frameworks.

Aspect	**XGBoost**	**LightGBM**
Tree growth strategy	Level-wise, balanced growth	Leaf-wise; converges faster but may over-fit
Algorithmic innovations	Second-order Taylor loss, L1/L2 regularisation	GOSS, EFB, histogram-based split search
Handling of categorical features	Requires one-hot encoding	Native support through equality splits
Computational performance	Slower training but highly robust	Faster training and memory-efficient

2.5.3 An Integrated Strategy for Handling Severe Class Imbalance

As established, predicting extreme weather is fundamentally a problem of class imbalance. If extreme events comprise only 1% of the INMET observations, a naive classifier can achieve 99% accuracy by never predicting an event, rendering it useless. To build a scientifically and operationally valuable model, we must integrate techniques specifically designed to address this imbalance directly into our framework.

1. Data-Level Resampling: Before training, the dataset can be balanced by modifying the class distribution. This can be achieved through:

- **Oversampling** the minority class, most notably using the Synthetic Minority Over-sampling Technique (SMOTE), which creates new, synthetic extreme-event samples by interpolating between existing ones, thus enriching the minority class representation (CHAWLA *et al.*, 2002).

- **Undersampling** the majority class, which involves removing non-event samples. While computationally efficient, this risks discarding valuable information about normal weather patterns.

The application of SMOTE to meteorological datasets has proven effective in improving the detection of rare severe weather events like tornadoes (CLARK *et al.*, 2018).

- **Cost-Sensitive Learning:** We can assign a much higher misclassification cost to the minority class (extreme events). In practice, this is implemented in XGBoost and LightGBM through the 'scale_pos_weight' hyperparameter, which directly adjusts the weight of positive class instances in the loss function calculation. This forces the model to pay significantly more attention to correctly identifying extreme events.

- **Specialized Ensembles:** Variants like the Balanced Random Forest train each constituent tree on a balanced bootstrap sample (containing an equal number of samples from each class), ensuring that every tree in the ensemble is exposed to the minority class (CHEN *et al.*, 2004).

By combining these data-level and algorithm-level techniques, we can construct a classification pipeline that is explicitly optimized not for overall accuracy, but for the successful detection of rare, high-impact events.

2.5.4 Hyperparameter Optimization and Model Validation

The performance of these advanced models is highly sensitive to their hyperparameters. A systematic optimization process is therefore the final critical step. Key parameters that control the trade-off between model complexity, training speed, and generalization are outlined in Table 2.7. We will employ rigorous k-fold cross-validation techniques combined with automated search strategies (e.g., grid search, random search, or Bayesian optimization) to identify the optimal hyperparameter configuration. Best practices such as using early stopping to prevent overfitting during boosting and analyzing feature importance scores to ensure model interpretability will be strictly followed.

In conclusion, the framework proposed in this thesis is not a generic application of machine learning, but a targeted, multi-stage strategy designed to overcome the specific challenges of extreme event prediction. We will implement and compare two state-of-the-art gradient boosting models, **XGBoost** and **LightGBM**, as our primary classifiers.

Furthermore, we will integrate **cost-sensitive learning** by tuning the *scale_pos_weight* hyperparameter. This is preferred over more complex ensemble balancing techniques for its direct and efficient integration into the gradient boosting optimization process, allowing the model to prioritize the minority class without altering the underlying data structure.

The entire pipeline from the choice of model to the imbalance correction strategy will be subjected to rigorous, data-driven hyperparameter optimization to produce a robust, highly performant, and scientifically insightful classification model for predicting extreme rainfall events in Brazil.

Table 2.7: Key hyper-parameters for tuning gradient boosting models and their impact on performance.

Parameter	**Description**	**Typical Range**	**Impact on Model**
`max_depth`	Maximum tree depth	3–10	Controls complexity; deeper trees capture more patterns but risk overfitting.
`n_estimators`	Number of trees in the ensemble	100–1000	More trees generally improve performance but increase compute time.
`learning_rate`	Contribution of each tree to the final prediction	0.01–0.3	Lower values require more trees but lead to better generalization.
`subsample`	Fraction of training data sampled per tree	0.5–1.0	Reduces variance and prevents overfitting.
`colsample_bytree`	Fraction of features sampled per tree	0.5–1.0	Prevents any single feature from dominating the model.
`scale_pos_weight`	Weight for the positive (minority) class	>1	Directly addresses class imbalance by penalizing false negatives more heavily.

Chapter 3

A Framework for Classifying Extreme Rainfall

This chapter details the comprehensive methodological framework designed to classify extreme rainfall events and is applied to the state of Rio Grande do Sul, Brazil. Building upon the foundational concepts of data-driven forecasting (GraphCast), ground-truth observation (INMET), and tree-based modeling established in Chapter 2, this section translates theory into a concrete, reproducible research pipeline.

The overarching goal is to develop a scientifically robust and operationally viable system capable of predicting high-impact weather events to support early warning systems for rain risk management. The architecture is predicated on a **"one-model-per-station"** philosophy, allowing the system to learn the unique *microclimates* and localized phenomena influencing each specific point of interest.

The chapter is structured to logically follow the flow of data and analysis: from problem formulation and data acquisition, through a critical data fusion process and feature engineering, to rigorous model training and validation, and finally, the operational architecture for real-time inference and interpretation.

3.0.1 Classifier Choice and XGBoost Rationale

Early-warning centres and municipal civil-defence agencies are interested in *whether* rainfall will breach a critical threshold, not in the exact millimetres that may fall. At a 48-hour lead time physical and numerical errors render point forecasts highly uncertain, and any subsequent thresholding discards much of the regressor's information. Casting the task as a binary classification therefore predicts *directly* the probability that the 48-hour accumulation will exceed 30 mm, providing an actionable risk score that plugs seamlessly into operational protocols.

Among the many candidate algorithms, we adopt **XGBoost** for four principal reasons:

1. **Rich non-linear expressiveness.** Gradient-boosted trees naturally capture high-order interactions between atmospheric variables and local physiography—behaviour that linear or generalized-linear models cannot reproduce.

2. **Built-in imbalance mitigation.** The `scale_pos_weight` hyper-parameter directly counteracts the scarcity of extreme-rain events, allowing the learner to focus on minority cases without resorting to potentially unstable re-sampling techniques.

3. **Computational efficiency at scale.** Training 45 station-specific models on an 8-core CPU finishes in under five minutes, enabling rapid iteration during cross-validation and hyper-parameter tuning.

4. **Transparent post-hoc explanations.** Tree-based SHAP values translate each forecast into an additive contribution from physically interpretable predictors—an important requirement for stakeholder trust that most deep neural architectures, including contemporary graph neural networks (GNN s), fail to meet.

Figure 3.1 summarises how these components interact within the end-to-end inference workflow.

Alternative graph-based classifiers such as graph neural networks s could capture spatial dependencies among stations, but deploying them would entail significantly higher implementation effort and computational cost—especially during hyper-parameter search—than the tree-based approach adopted here. Under the strict latency and resource constraints of the operational pipeline, XGBoost therefore offers a more practical balance of skill and efficiency.

Binary target definition. The response variable is set to 1 when the cumulative precipitation predicted for the **next 48 hours** exceeds 30 mm at a station and 0 otherwise. This threshold is prescribed by the Brazilian civil-defence handbook (Sec, 2024) and ensures that every positive prediction corresponds to a legally actionable rainfall event.

3.1 Data Acquisition and Preprocessing

The framework's predictive power is contingent upon the quality and meticulous integration of two primary data sources: historical ground-truth observations from INMET and state-of-the-art forecast data from GraphCast.

3.1.1 INMET Observational Data

The ground truth for this study is derived from hourly observational data provided by INMET for **45 weather stations** distributed across Rio Grande do Sul. This raw dataset contains fundamental meteorological variables, including hourly precipitation (mm), temperature (°C), and station metadata (latitude, longitude, elevation).

All available measured weather data for 2024 were downloaded from INSTITUTO NACIONAL DE METEOROLOGIA (2025) via the official portal. The download encompasses data for all weather stations in Brazil, including the measured variables (e.g., precipitation and temperature) as well as associated metadata and auxiliary features. This comprehensive retrieval ensures full spatial coverage and consistent temporal alignment for downstream processing and evaluation. This data is available with the measures hourly, which means that we have to process and aggregate it to join with graph cast.

3.2 GraphCast Forecast Generation

The predictive features central to this study are derived from forecasts generated by Google DeepMind's GraphCast model. To ensure robustness, reproducibility, and scalability, we have developed an automated pipeline.

This pipeline orchestrates the entire forecast generation workflow, from provisioning computational resources and managing the model's execution to the eventual storage of the output data. This section provides a detailed technical description of the pipeline's architecture and its constituent components.

3.2.1 System Architecture

The forecast generation pipeline is a distributed system integrating local orchestration scripts, cloud-based GPU computing, containerization technology, and cloud storage services. The architecture is designed to programmatically manage the entire lifecycle of a forecast run with minimal manual intervention. The primary components are:

1. **Local Orchestration Client:** A master script initiates and oversees the forecast generation process based on user-defined parameters.

2. **On-Demand GPU Cloud Provider: RunPod** is used for the dynamic provisioning of high-performance, secure cloud GPU servers required for the computationally intensive model execution.

3. **Containerized Execution Environment:** A **Docker** container encapsulates the GraphCast model and all its dependencies, ensuring a consistent and reproducible runtime environment across different executions.

4. **Persistent Cloud Storage: Amazon Web Services (AWS) S3** is utilized as the data sink for storing the generated forecast outputs and for state management signals between the remote server and the local client.

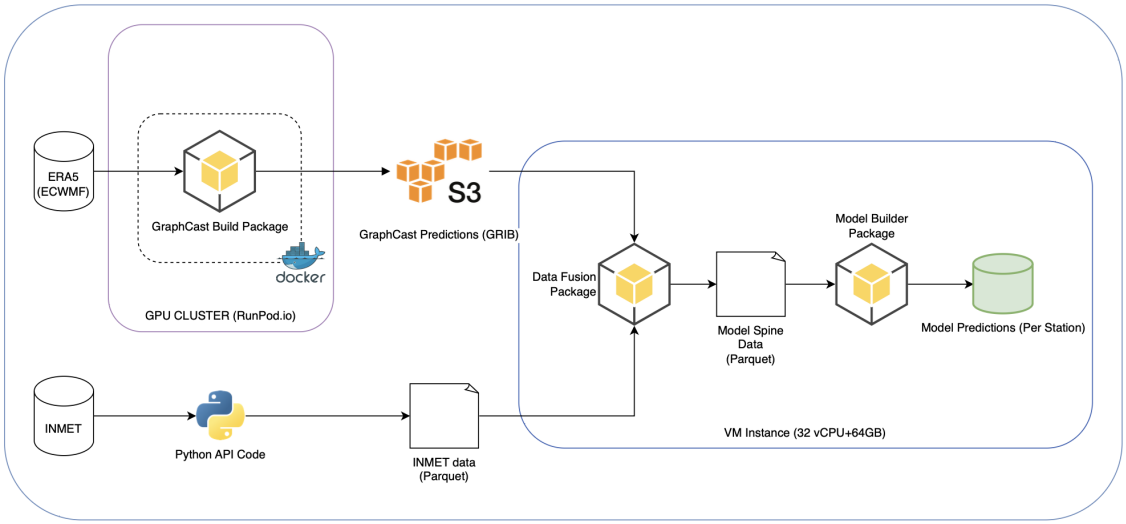


Figure 3.1: End-to-end inference architecture showing data flow between GPU and CPU tiers.

3.2.2 Rationale Against Fine-Tuning GraphCast

Fine-tuning the GraphCast network itself was initially considered, but discarded for three practical and scientific reasons:

- **Data scarcity.** Even after spatial aggregation, the April–May 2024 window provides <4,000 six-hourly samples per station—five orders of magnitude fewer than the 1979–2017 ERA5 archive used to train GraphCast. Transfer-learning under such extreme data scarcity risks catastrophic over-fitting.

- **Compute and latency.** GraphCast contains more than 500 million parameters and requires ˜30 GB of GPU memory for inference. Fine-tuning any subset of layers would at least double memory usage and training time, vastly exceeding the cost of training a lightweight post-processor.

- **Operational simplicity.** Treating GraphCast as an immutable feature generator enables us to upgrade to future GraphCast checkpoints—or switch to alternative global models—without re-training hundreds of millions of parameters. This decoupling is critical for a maintainable early-warning system.

For these reasons, we adopt the hybrid fusion paradigm detailed in the following sections: GraphCast remains frozen, and a tree-based classifier is trained independently on gauge-level targets.

3.2.3 Orchestration and Remote Execution

The pipeline is initiated from a local machine by executing an Python package developed for this project, called **graphcast-build**. This package serves as the master controller for the entire operation.

Parameterization

The behavior of each forecast run is defined in a YAML configuration file, `parameters.yml`. This file specifies critical parameters, including:

- `start_datetime` and `end_datetime`: The date range for which to generate forecasts.
- `forecast_step_hours`: The interval between consecutive forecast initializations (e.g., 12 hours for 00 and 12 UTC cycles).
- `hours_to_forecast`: The lead time for each forecast run.
- `gpu_type`: The specific model of GPU to be provisioned. For this study, the **NVIDIA A100-SXM4-80GB** was explicitly chosen to meet the GraphCast model's substantial memory requirement of over 61 GB.
- `disk_size_gb`: The allocated container disk space, set to 50 GB to accommodate the model assets, input data, and forecast outputs.

Remote Infrastructure Management

This orchestration leverages an interface with the run. Upon execution, the `remote_cast` function within this package performs the following automated actions:

1. **Pod Creation:** It programmatically requests a new secure cloud server (a "pod") from RunPod, configured with the specified GPU and disk size.
2. **Environment Configuration:** It launches the designated Docker container, `viniciusribeiro157/graphcast-build:latest`, on the pod. Sensitive credentials for AWS and the Copernicus Climate Data Store (CDS), along with the forecast parameters, are securely passed into the container as environment variables.

3. **Execution Monitoring:** After deploying the pod, the script enters a polling loop. It periodically checks a predefined location in the designated AWS S3 bucket for a completion signal, while also monitoring the pod's status via the RunPod to detect any premature termination or errors.

3.2.4 Core Forecasting Process

All model-related operations occur within the isolated Docker environment on the remote RunPod server. The entry point inside the container executes the `cast_all` function from the `graphcast-build.cast` module.

The `ai-models-graphcast` Wrapper

The core of the forecasting process is managed by the `ai-models-graphcast` package. This package serves as a high-level wrapper around DeepMind's official GraphCast implementation, abstracting away much of the complexity involved in data handling and model invocation. The `GraphcastModel` class is instantiated with parameters that define the data source, output path, and model configuration. For this research, the operational, high-resolution GraphCast model was used, which relies on pre-trained weights loaded from the checkpoint file: `GraphCast_operational - ERA5-HRES 1979-2021 - resolution 0.25 - pressure levels 13 - mesh 2to6 - precipitation output only.npz`.

Data Ingestion and Model Execution

For each forecast cycle specified by the orchestration script, the following automated steps are performed within the container:

1. **Authentication:** A `.cdsapirc` file is dynamically created to authenticate with the Copernicus Climate Data Store (CDS) API.

2. **Data Fetching:** The `ai-models-graphcast` package automatically requests and downloads the necessary initial conditions from the ERA5 reanalysis dataset via the CDS API. As GraphCast is an autoregressive model, it requires two initial states. For a forecast beginning at time t_0, the model is initialized with ERA5 data from t_0 and the preceding cycle, $t_0 - 6$ hours.

3. **Prediction:** The downloaded data is preprocessed into the required `xarray.Dataset` format. The `run()` method is then called, which invokes the JAX-compiled, autoregressive prediction function. The model generates the forecast sequentially for the specified lead time, producing outputs at 6-hour intervals.

4. **Output Generation:** The raw output from the model is post-processed and saved to a single file in the format.

3.2.5 Data Storage and Pipeline Teardown

Cloud Storage and Data Volume

Upon the completion of a forecast run for a single initialization time, the resulting GRIB file is immediately uploaded from the RunPod container to a designated AWS S3 bucket. The path within the bucket is structured by a unique `cast_id` to organize outputs from different pipeline executions. The volume of the generated data is substantial; a single 10-day forecast produces a GRIB file of approximately **6.5 GB**.

In our case, we have generated about **500 GB** of data, since we are generating for a range of 25 days times the 4 initialization periods available.

Lifecycle and Cleanup

After all forecast dates in the specified range have been processed and their corresponding GRIB files have been uploaded to S3, the remote script writes a final, empty "completion" file to the S3 bucket. The local orchestration script, which has been polling for this file, detects its presence. This signal confirms the successful completion of the entire job, prompting the script to make a final API call to RunPod to terminate the GPU pod. This final step ensures that computational resources are released automatically, preventing unnecessary costs.

3.2.6 The Graphcast data

The GraphCast produces forecasts at a global **0.25° x 0.25° resolution** at 6-hour intervals (00, 06, 12, 18 UTC). For this study, we extract 10 core atmospheric variables at the surface and across 13 distinct pressure levels. These levels were specifically chosen to capture conditions from the planetary boundary layer (1000–850 hPa), through the mid-troposphere where key weather-forming processes occur (700–300 hPa), and into the upper troposphere to capture jet stream dynamics (250–100 hPa). The selected variables and levels are detailed in Table 3.1.

Explicit Chronological Splits. Random shuffling would leak information from the future into the past. Therefore, we enforce a strictly temporal partition for *each* station:

- **Training window:** 1 April–20 May 2024 (inclusive).

Table 3.1: Core GraphCast variables and pressure levels extracted for feature engineering. These variables provide a comprehensive snapshot of the atmospheric state from the surface to the upper troposphere.

Variable Code	Description	Pressure Level(s)	Units
`10u`, `10v`	10 metre U/V wind component	Surface	m/s
`2t`	2 metre temperature	Surface	K
`tp`	Total precipitation	Surface	m
`q`	Specific humidity	13 levels[a]	kg/kg
`t`	Temperature	13 levels[a]	K
`u`, `v`	U/V component of wind	13 levels[a]	m/s
`w`	Vertical velocity	13 levels[a]	Pa/s
`z`	Geopotential	13 levels[a]	m^2/s^2

[a]Pressure levels (hPa): 1000, 925, 850, 700, 600, 500, 400, 300, 250, 200, 150, 100, 50.

- **Validation window:** the subsequent seven days (21–27 May 2024) in the first fold; this window advances three days with each outer fold.
- **Test window:** 28–31 May 2024—kept completely unseen until final evaluation in Chapter 4.

Four rolling-origin folds are produced by sliding the training/validation cut-off forward in three-day steps. Hyper-parameter tuning (inner loop) operates exclusively within the training subset of each fold; validation and test targets remain untouched, guaranteeing zero temporal leakage.

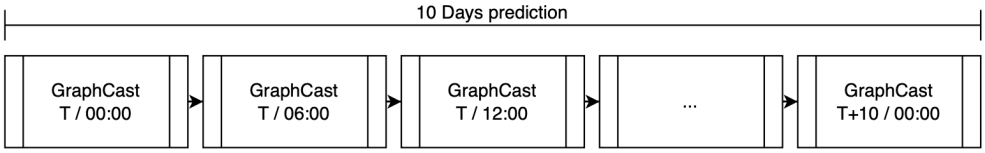


Figure 3.2: Rolling GraphCast forecast schedule. GraphCast is executed every 6 hours to produce overlapping predictions that together span a continuous 10-day horizon; arrows indicate the temporal succession of model runs.

3.2.7 Data Cleaning, Transformation, and Imputation

To construct our model, we need to gather datasets from multiple sources and perform data fusion to integrate their distributions into a unified data spine. This spine establishes a consistent temporal reference, enabling the seamless alignment of both datasets and their associated features for cohesive analysis and modeling.

Prior to data fusion, the raw data undergoes several essential preprocessing steps:

- **Unit Conversion**: GraphCast temperature variables (K) are converted to Celsius by subtracting 273.15, and total precipitation (m) is converted to millimeters by multiplying by 1000 to match INMET standards.

- **Missing Value Imputation**: An analysis revealed missingness of 5.01% in INMET temperature data and 8.63% in GraphCast's `tp` (total_precipitation) variable. We employ a variable-specific imputation strategy: mean imputation for the slowly varying, continuous temperature field. (LITTLE and RUBIN, 2019).

3.3 Data Fusion and Feature Engineering

This section details the critical processes of aligning the datasets and constructing a powerful, informative feature set for the machine learning model.

3.3.1 Spatio-Temporal Alignment

1. **Temporal Aggregation**: To match GraphCast's 6-hour resolution, the hourly INMET data is aggregated into windows centered at 00, 06, 12, and 18 UTC. During this process, we compute statistics such as the sum of precipitation and mean of temperature. The primary aggregated features are detailed in Table 3.2. Some of these features generated were used to validate if the aggregation of the target was generating a distribution suited to the original target.

2. **Spatial Interpolation using IDW**: To estimate the gridded GraphCast forecast at each point-based INMET station, we employ **Inverse Distance Weighting** (SHEPARD, 1968). For each station s, the forecast value $\hat{V}_s$ is computed as a weighted average of the values V_i at the four nearest grid points, where the weight w_i is the inverse of the squared distance d_i:

$$\hat{V}_s = \frac{\sum_{i=1}^{4}(V_i/d_i^2)}{\sum_{i=1}^{4}(1/d_i^2)}$$

   This standard interpolation method is chosen over simpler nearest-neighbor assignment as it provides a more physically plausible and continuous estimation of the forecast field by accounting for local meteorological gradients.

3. **Spatial Linking via Radius Search**: To link point-based station data with the gridded forecast data, a spatial filtering strategy is employed. For each of the 45 INMET stations, we calculate the great-circle distance to all unique

GraphCast grid points using the **Haversine formula**. All grid points within a **50 km search radius** are identified as influential neighbors for that station. This method is deliberately chosen over simpler nearest-neighbor or bilinear interpolation schemes to capture a broader, more physically representative atmospheric state around the point of interest, acknowledging that weather phenomena are not confined to a single grid cell.

The results of this spatial linking, shown in Figure 3.3, are fundamental to the architecture. The number of associated grid points per station varies, typically between 7 and 13, with a strong mode at 12, reflecting the geometry of the GraphCast grid relative to station locations.

The distance statistics confirm the appropriateness of the 50 km radius: the mean distance from a station to its associated grid points is consistently between 32 and 36 km, with the maximum distance remaining within the search boundary. The geographical distribution reveals no significant spatial bias, ensuring equitable data representation across the study area.

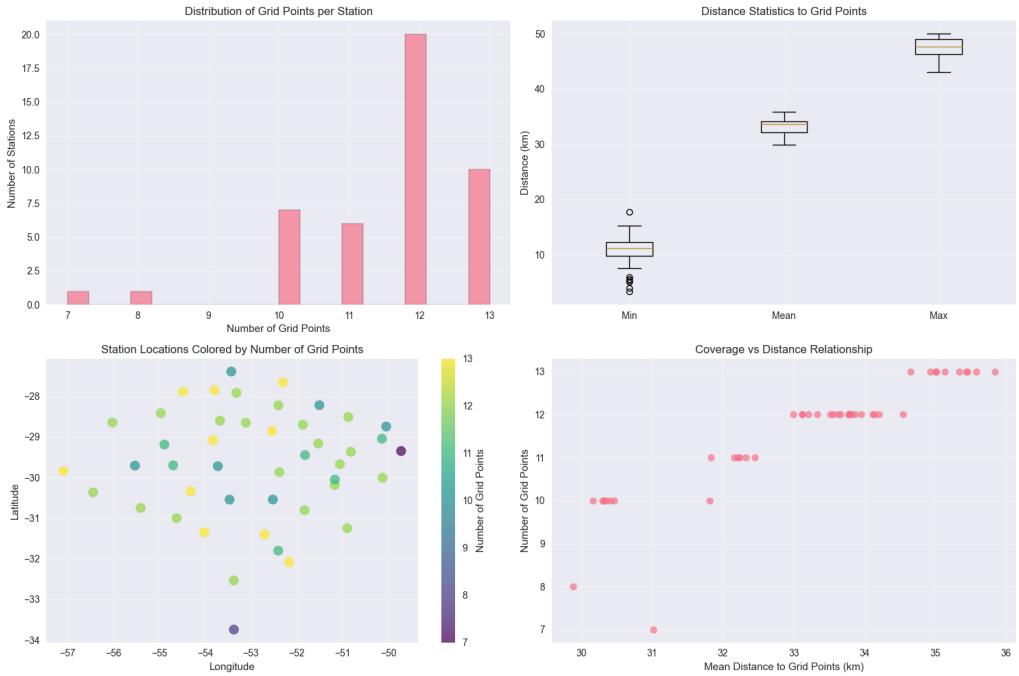


Figure 3.3: Diagnostic plots illustrating the outcome of the spatial linking process. (Top-Left) Distribution of the number of associated grid points per station. (Top-Right) Box plots of minimum, mean, and maximum distances from stations to their associated grid points. (Bottom-Left) Geographical map of station locations, colored by the number of associated grid points.

Table 3.2: Primary features derived from raw INMET data after 6-hour aggregation.

Variable	Description	Units
6-h Precipitation (sum)	Total precipitation accumulated in the 6-h window	mm
6-h Precipitation (mean)	Average precipitation rate within the 6-h window	mm
Temperature (max, mean)	Average of station maximum temperatures	°C
Temperature (min, mean)	Average of station minimum temperatures	°C

3.3.2 Historical Ground-Truth Predictors (Lag Features)

To provide the model with memory and context of recent ground-truth conditions, a critical factor for capturing temporal autocorrelation in weather patterns—we systematically engineer historical features from the station's own aggregated INMET data. This process is managed by the `LagCreator` module.

The module operates on the `inmet_precipitation_6h_mm` column and is governed by the `max_lag_regressive` parameter in the pipeline's configuration ('config.py'), which defines the maximum number of historical periods to consider. For a given *max_lag_regressive* value of L, the module creates L new features by shifting the time series. The value of each lag feature at a given time T is defined as:

$$\texttt{precip\_lag\_k}(T) = \texttt{precip\_6h\_sum}(T - k) \quad \text{for } k = 1, 2, \ldots, L$$

For instance, with `max_lag_regressive=3`, three new features are generated: `precip_lag_1`, `precip_lag_2`, and `precip_lag_3`, representing the observed precipitation in the 6-hour periods ending 6, 12, and 18 hours prior, respectively.

These lag features serve as a powerful proxy for antecedent moisture conditions and recent storm dynamics. Their importance is considered by explicitly protecting these features during the subsequent feature selection step. The `FeatureSelector` module is hard-coded to identify these columns by their naming convention and **force-include** them in the final feature set, ensuring this vital temporal context is never inadvertently discarded.

3.3.3 Target Engineering

Instead of forecasting precise precipitation amounts at the next time step (a regression task), we focus on evaluating the **probabilistic risk of a dangerous cumulative rainfall event** within a future forecast horizon. This redefinition transforms the problem into a forward-looking binary classification task, aligning with the operational need for actionable early warnings.

The **TargetCreator** module, integrated within the **model_builder** pipeline, orchestrates this transformation. Its behavior is governed by three key parameters:

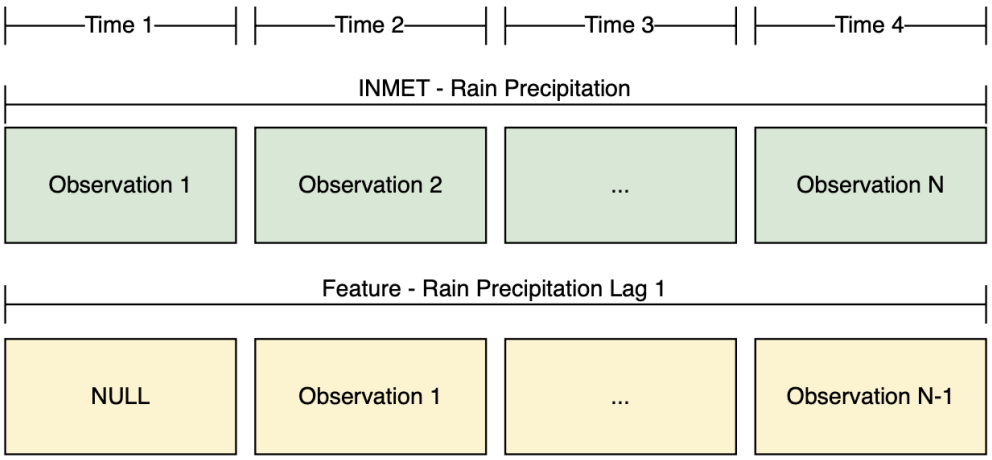


Figure 3.4: Illustration of how the rain precipitation lag feature is constructed: the original INMET observations (top row) are shifted by one time step to form the lagged feature (bottom row), introducing a null at the start and aligning prior observations for prediction.

- **target_column**: Specifies the source of continuous, aggregated ground-truth precipitation data, set to **inmet_precipitation_6h_mm**, representing rainfall accumulated over 6-hour intervals.
- **target_window_size**: Defines the temporal look-ahead window for cumulative summation, set to 8 periods (equivalent to a 48-hour forecast horizon). This duration balances meteorological predictability with the lead time required for effective flood preparedness by civil defense agencies.
- **heavy_rain_threshold**: Establishes the cumulative rainfall threshold for a positive (Class 1) event within the **target_window_size**.

The binary target, **heavy_rain_target**, is constructed by iterating chronologically through the **inmet_precipitation_6h_mm** time series. For each timestamp T, the cumulative precipitation $C(T)$ is calculated as the sum of observed rainfall over the subsequent periods (window_size = 8), which is equivalent to 48 hours:

$$C(T) = \sum_{k=1}^{8} P(T+k),$$

where $P(T+k)$ is the observed precipitation at the k-th 6-hour interval after T. The binary target is then defined as:

$$\text{heavy\_rain\_target}(T) = \begin{cases} 1 & \text{if } C(T) \geq \texttt{heavy\_rain\_threshold}, \\ 0 & \text{otherwise.} \end{cases}$$

A key challenge is handling data points near the end of the time series, where

fewer than 8 future periods are available. Discarding these samples would result in data loss, particularly in datasets with limited temporal extent. The **TargetCreator** employs a two-pronged strategy:

1. **Proportional threshold scaling for partial windows**: For timestamps T with $W_{\text{avail}}(T) < 8$ future periods available, the rain threshold is scaled proportionally to maintain consistent event severity. The scaled threshold is

$$\text{Threshold}_{\text{scaled}}(T) = \texttt{heavy\_rain\_threshold} \times \left(\frac{W_{\text{avail}}(T)}{8} \right),$$

   where $W_{\text{avail}}(T)$ is the number of available future periods from $T+1$ to the dataset's end. The binary target is then assigned using this adjusted threshold, preserving the physical significance of the event definition.

2. **Fallback imputation for missing data**: At the extreme tail of the time series, where no valid precipitation data exists (e.g., all `NaN`), the **TargetCreator** applies backward-fill (`bfill`) imputation to the heavy rain target column. This propagates the last valid target value to fill missing entries, ensuring no loss of rows for feature alignment.

This target engineered **heavy_rain_target**, combined with a comprehensive feature set, forms a robust analytical base table, providing a clean, chronologically aligned, and interpretable input for the supervised learning model.

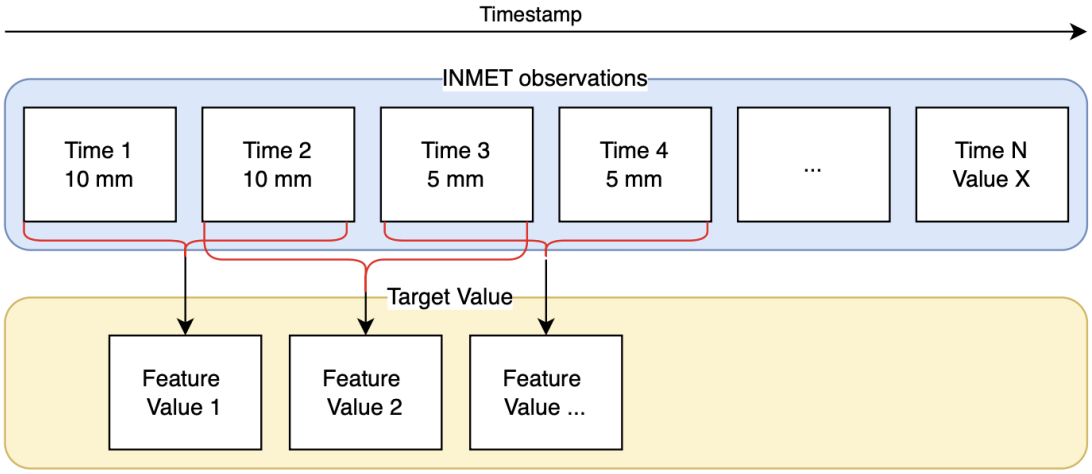


Figure 3.5: Illustration of target construction using a rolling aggregation window. INMET observations (top) are combined according to the predefined formulation $C(T) = \sum_{k=1}^{8} P(T+k)$; the red lines indicate which observation timestamps are aggregated into the target value (bottom), making explicit the windowed summation that defines the label.

In the figure 3.5, each red line traces how individual precipitation observations contribute to the aggregated target at a given time. This visual encodes the tempo-

ral neighborhood used in the heavy rain target formulation, showing which future INMET observations are summed to compute $C(T)$, thereby defining the binary heavy rain label.

3.3.4 Feature Selection and Scaling

To manage the high dimensionality created by feature flattening and to mitigate the risk of model overfitting, final preparation steps are applied.

1. **Standardization.** All continuous features are standardized by removing the mean and scaling to unit variance using a `StandardScaler`. This scaler is fitted only on the training data of each cross-validation fold to prevent data leakage from the validation set into the training process.

2. **Feature Selection.** The feature flattening process, while necessary, introduces significant multicollinearity, as evidenced by the sample correlation heatmap in Figure 3.6. The heatmap reveals strong positive and negative correlations between variables, particularly those from the same physical field at different pressure levels or adjacent grid points. To address this and reduce the feature space from over 10,000 to a manageable number, a filter-based feature selection method using the ANOVA F-test (`f_classification`) is employed.

This method selects a predefined number of features most statistically correlated with the target variable. We explicitly preserve the engineered lag features, acknowledging their high predictive value based on domain knowledge. This approach is preferred over dimensionality reduction techniques like , as it mitigates multicollinearity while retaining the original, physically interpretable atmospheric variables.

3.4 Model Training and Validation

This section details the core experimental design, from algorithm selection to the rigorous processes for handling class imbalance and validating model performance.

3.4.1 Evaluation Metrics for Imbalanced Classification

The predictive task of identifying events is characterized by a severe class imbalance, where the positive class accounts less than 10% of observations. In such a scenario, conventional metrics like accuracy are profoundly misleading, as a model could achieve over 90% accuracy by simply never predicting a heavy rain. To provide a meaningful and robust assessment of model performance, we employ a suite of

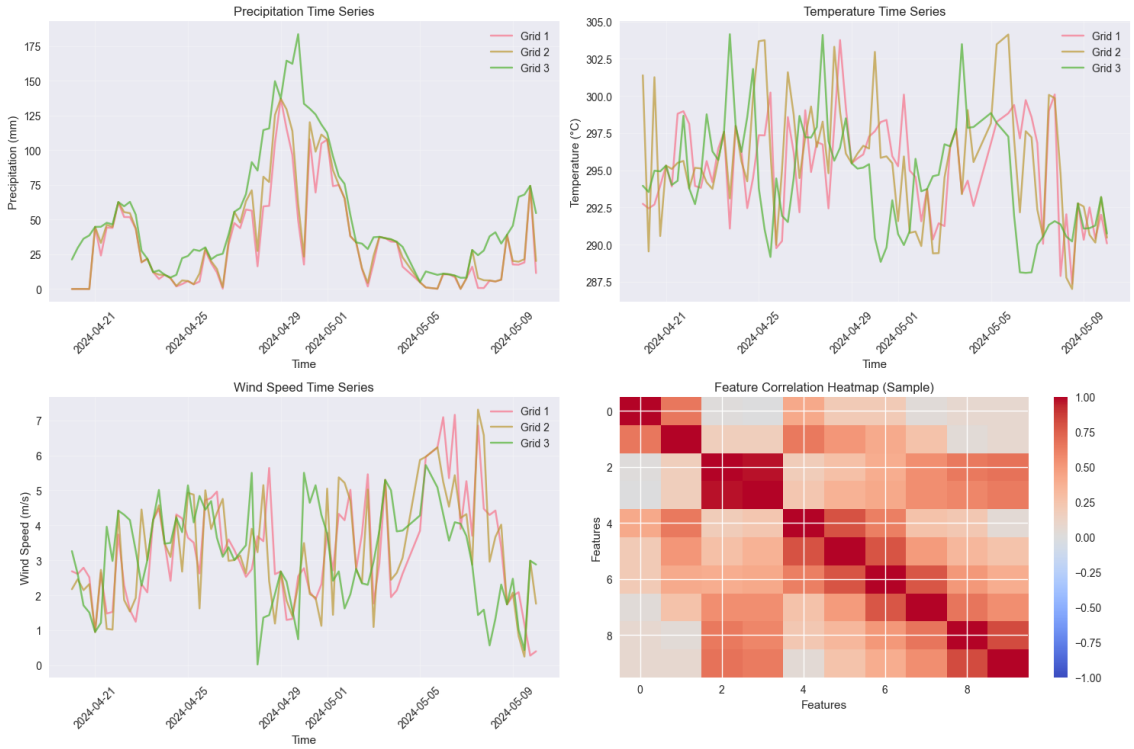


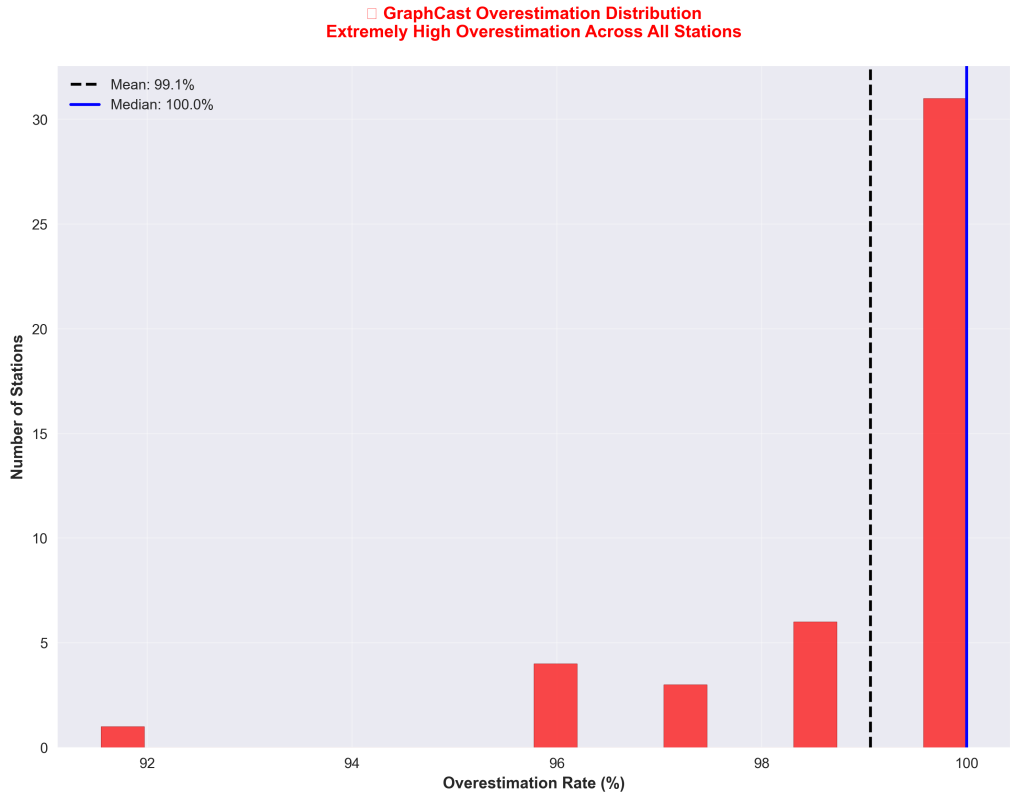
Figure 3.6: Justification for the feature flattening strategy. (Left) Time series of precipitation, temperature, and wind speed for three adjacent GraphCast grid points, illustrating significant local variance. (Right) A sample feature correlation heatmap, revealing high multicollinearity among features from different pressure levels and grid points, which necessitates the feature selection step. The correlation here its to prove that the same behavior can be observed between grid stations, just select a set of 10 features as sample.

metrics designed for imbalanced classification tasks (JAPKOWICZ and STEPHEN, 2020), with a deliberate focus on the inherent trade-offs in predicting rare, high-consequence events.

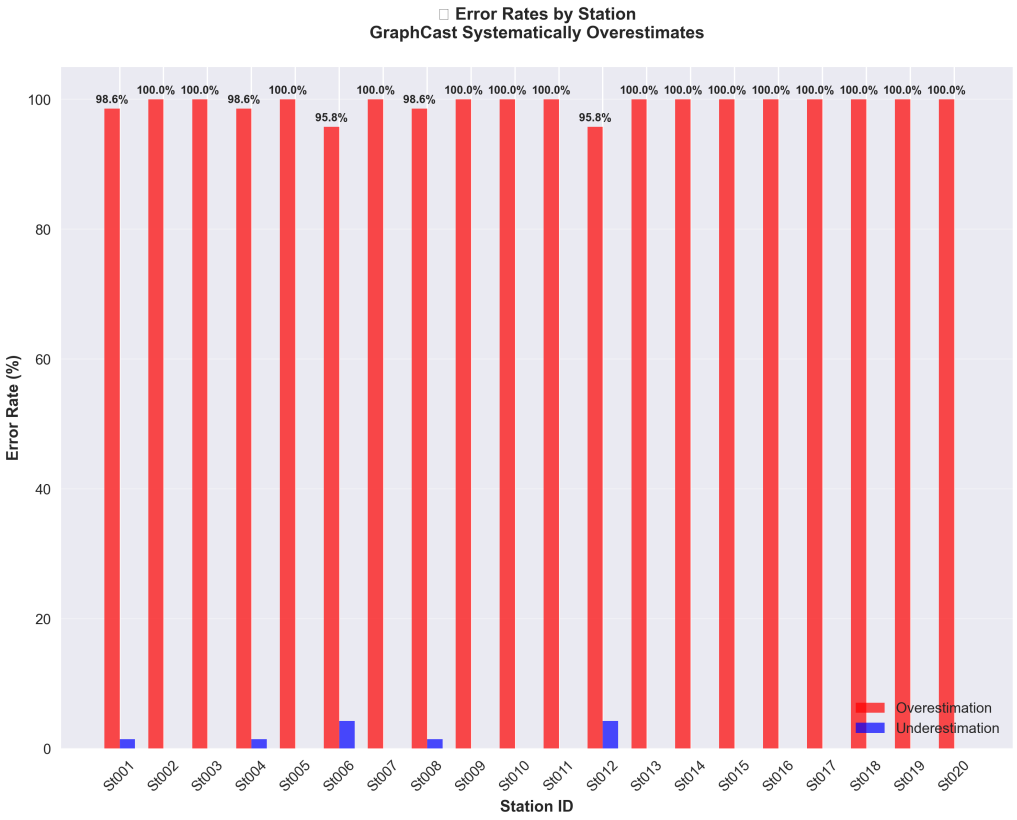
Precision–Recall Trade-off in Heavy-Rain Forecasts

Model skill is assessed through *recall* (sensitivity) and *precision*, whose optimization pushes the decision threshold in opposite directions. We also report *accuracy* as a holistic baseline that contextualizes these metrics and helps detect degenerate majority-class predictors; interpretation, however, prioritizes changes in recall and precision in the imbalanced setting.

Definition of the *GraphCast-only* Baseline. Throughout Section 3.4 we refer to a *GraphCast-only* baseline. In contrast to the deterministic thresholding approach initially drafted, the final baseline trains the same decision-tree ensemble architecture (hyper-parameters unchanged) but restricts the predictor matrix to the **original GraphCast forecast columns only**. No INMET lags, hand-crafted features, or auxiliary variables are supplied. The target remains exactly the heavy-rain label engineered from INMET observations (Section 3.3). This setup isolates the incremental value contributed by the additional fused features—any improvement over the GraphCast-only model therefore stems from those extra predictors rather than from changes in the learning algorithm or objective.

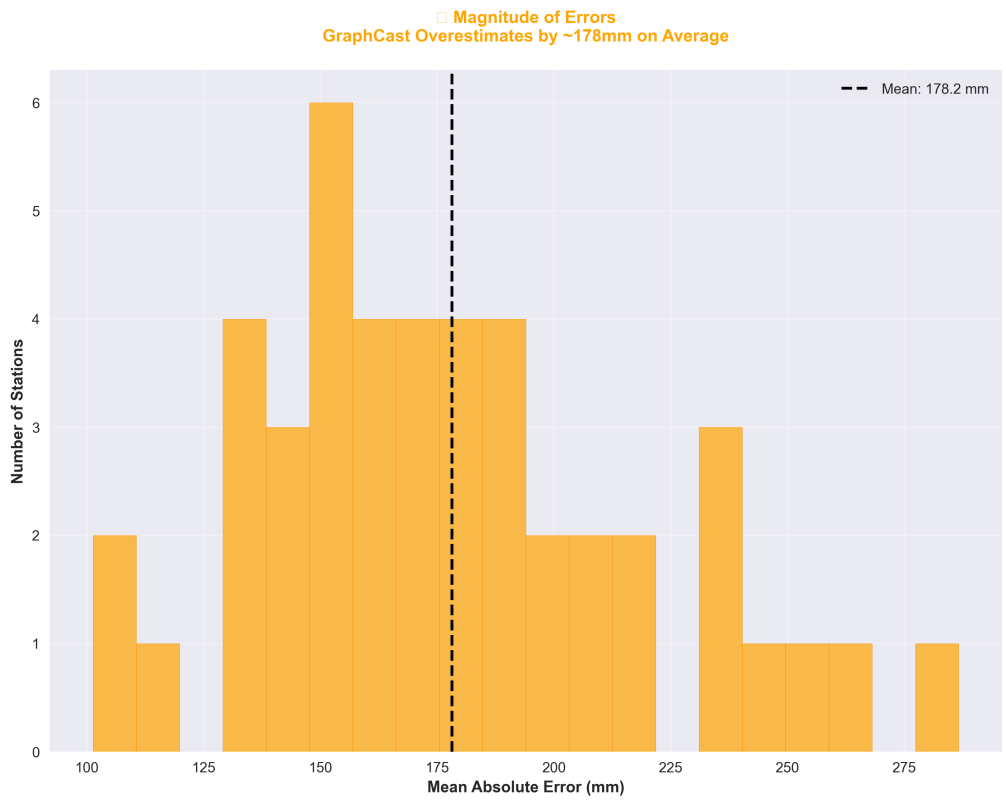


(a) Station-level over-estimation rates

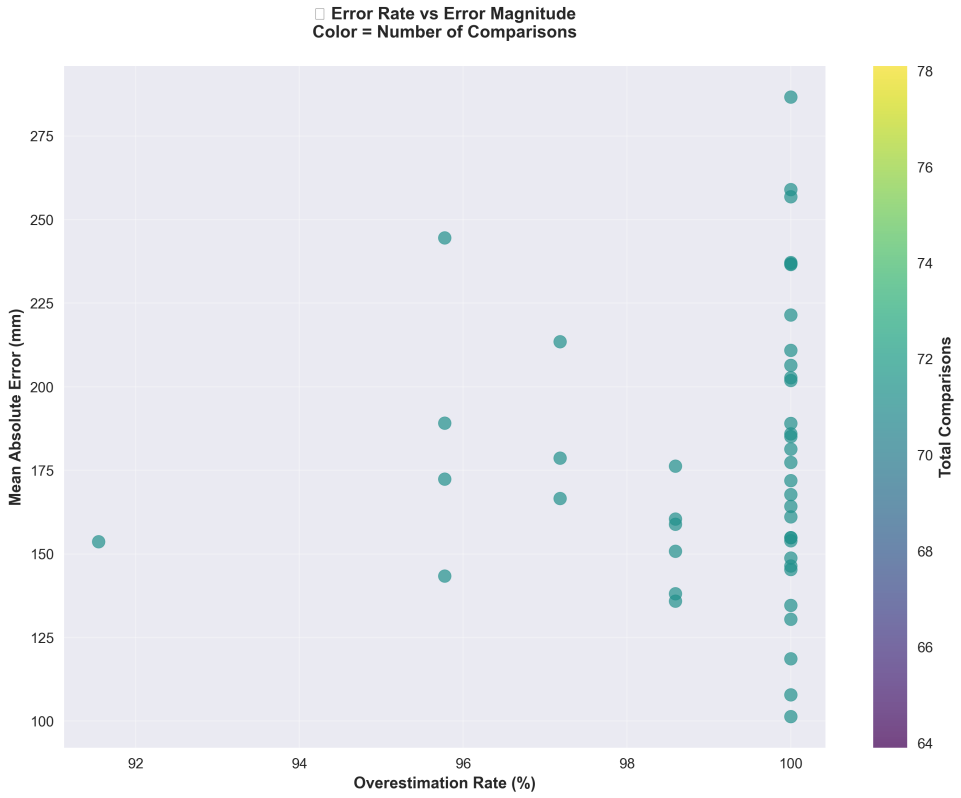


(b) Error composition by station

Figure 3.7: Error diagnostics for GraphCast precipitation forecasts against INMET observations.



(c) Distribution of mean absolute error



(d) Over-estimation rate vs. MAE

Figure 3.8: Error diagnostics for GraphCast precipitation forecasts against INMET observations (continued).

Figure 3.7 confirms that the raw GraphCast outputs systematically over-estimate accumulated rainfall measured by INMET. Over 99 % of the station–event pairs fall above the one-to-one line, and the median station shows a 100 % over-estimation rate. The histogram in panel (a) is sharply right-skewed, with more than half of the stations clustered at the extreme 100 % mark. Panel (b) reinforces this pattern: every station exhibits a dominant red bar, indicating that the forecast error is almost entirely due to over-prediction rather than under-prediction.

Panel (d) quantifies the magnitude of those errors. The mean absolute error (MAE) across stations is 178 mm, and even the best five stations still exceed 120 mm. When the over-estimation rate is plotted against MAE in panel (c), a clear positive association emerges—stations that are consistently over-estimated also incur the largest absolute errors, with many points surpassing 250 mm. Together, these results demonstrate that the bias is not merely directional but also substantial in size, strongly supporting the need for bias-correction strategies explored later in this work.

The *GraphCast-only* baseline in Section 3.4.1 uses the usual INMET label but feeds the model nothing apart from the raw GraphCast forecast columns. Later chapters will show how much skill we gain when we add station observations and extra hand-crafted features on top of this minimal configuration.

Let TP, FP, FN, and TN denote the numbers of true positives, false positives, false negatives, and true negatives, respectively.

$$\text{Recall } (R) = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3.1}$$

$$\text{Precision } (P) = \frac{\text{TP}}{\text{TP} + \text{FP}}\,. \tag{3.2}$$

$$\text{Accuracy } (A) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}\,. \tag{3.3}$$

Meaning for heavy-rain warnings. Equation (3.1) measures the proportion of *observed heavy-rain episodes* (e.g., 24-h totals exceeding a prescribed threshold) that are successfully forecast. High recall is essential: a low-recall system misses many genuine events (FN), depriving decision-makers of timely information for civil-defence planning.

Equation (3.2) quantifies the reliability of the heavy-rain alerts that are actually issued. Insufficient precision implies an excess of false alarms (FP), leading to "warning fatigue," unnecessary mobilisation of resources, and erosion of user confidence in the forecasting service.

The intrinsic trade-off. Lowering the alert threshold raises R by tagging *borderline* cases as positives, but this swells FP and drags down P. In contrast, a *strict* threshold boosts P at the cost of *extra* misses, trimming R.

Selecting an operating point on the precision–recall curve therefore requires balancing the societal cost of a missed heavy-rain event against that of an unwarranted alert—an optimisation typically carried out in consultation with end-users such as emergency-management agencies and hydrometeorological services.

The F1-Score: A Balanced Metric for Model Selection

Given this trade-off, we selected the **F1-Score** as the primary metric for both hyperparameter optimization and overall model comparison. The F1-Score is the harmonic mean of Precision and Recall:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The choice of the harmonic mean is deliberate and crucial. Unlike a simple arithmetic mean, the harmonic mean heavily penalizes extreme values. A model cannot achieve a high F1-Score by excelling at one metric while failing at the other; it must find a robust balance between them.

This property makes the F1-Score uniquely suited to our objective: to develop a model that is both a reliable detector of true threats (high recall) and a trustworthy source of warnings (high precision). By maximizing the F1-Score during hyperparameter tuning, we directly optimize the model to find the most effective compromise between not generating false alerts and not failing to generate alerts for real events.

Overall Performance Check with AUC–ROC

While the F1 score is our main metric when we *tune* the model for a fixed cut-off (usually 0.5), we also quote the **Area Under the Receiver Operating Characteristic curve (AUC-ROC)**. AUC-ROC scores how well the model can *tell* positives from negatives over every cut-off. It gives an overall, cut-off-free score of the model's ability to separate the classes, standing alongside the F1 score to provide a full view of classification skill.

3.4.2 Model Selection and Justification

The selection of the modeling framework was a deliberate decision driven by the operational requirements of the problem and the nature of the available data. This

section justifies the key choices: the use of a classification framework, the adoption of a post-processing model, and the selection of XGBoost as the specific algorithm.

Classification for Actionable Warnings. The decision to frame this problem as classification rather than regression was driven by the direct needs of operational warning systems. Civil defense agencies and stakeholders issue alerts not based on a precise rainfall prediction (e.g., 47.5 mm), but on whether precipitation is likely to *exceed a critical, predefined threshold* (e.g., 40 mm). A classification model is purpose-built for this task; it directly estimates the probability of this threshold exceedance, providing an immediate and actionable risk assessment.

A Post-Processing Ensemble for Targeted Downscaling. Several modeling strategies were considered. Fine-tuning the GraphCast model itself was deemed impractical, as it is computationally prohibitive and unsuited for the specific goal of localized downscaling. A post-processing model is a more direct and efficient method for this task.

Based on the analysis in Chapter 2, and given that the relationship between large-scale atmospheric variables and localized rainfall is profoundly non-linear, we select **XGBoost (Extreme Gradient Boosting)** as our primary classification algorithm (CHEN and GUESTRIN, 2016). XGBoost offers a compelling combination of high predictive performance, an ability to intrinsically handle heterogeneous tabular data without complex pre-processing, and greater interpretability through feature importance scores. This makes it exceptionally well-suited for capturing the complex interactions within the fused dataset and solving this scientific classification task.

3.4.3 Methodology for Handling Severe Class Imbalance

The dataset exhibits a severe class imbalance, with heavy rain events (the positive class) constituting less than 10% of all observations. Left unaddressed, this would lead a standard classifier to achieve high accuracy simply by predicting the majority (non-heavy-rain) class, rendering it useless for practical heavy rain prediction. To counteract this, we evaluated two canonical strategies for imbalance handling: an algorithm-level approach and a data-level approach.

After careful consideration and preliminary analysis, we adopted **algorithm-level cost-sensitive learning** as the primary strategy for this research. This decision was motivated by a desire to avoid the potential pitfalls associated with data-level resampling techniques like SMOTE.

While effective, SMOTE's mechanism of generating synthetic minority samples carries a notable risk of overfitting. In a meteorological context, interpolating be-

tween distinct weather states could create physically implausible or artificial atmospheric patterns, leading the model to learn representations that do not generalize well to unseen, real-world data. The synthetic instances can introduce artificial patterns or blur the decision boundary.

By contrast, the algorithm-level approach does not manipulate the underlying data distribution. Instead, it directly modifies the model's learning process to assign a higher penalty to misclassifications of the minority class. This is achieved in our XGBoost model by configuring the `scale_pos_weight` hyperparameter. This parameter adjusts the weight of the positive class in the loss function, compelling the model to prioritize its correct identification. It is calculated dynamically for the training data within each cross-validation fold using the following formula:

$$\text{scale\_pos\_weight} = \frac{n_-}{n_+}, \tag{3.4}$$

where n_- and n_+ are the number of negative and positive examples, respectively.

This method offers a more robust solution by preserving the integrity of the original data while still effectively addressing the imbalance, thereby reducing the risk of overfitting and promoting better generalization.

3.4.4 Hyperparameter Optimization via Bayesian Search

To ensure the XGBoost model achieves its maximum potential performance, we implemented a rigorous hyperparameter optimization process within our nested cross-validation framework. This process is orchestrated by our project's custom `HyperparameterTuner` class, which is built upon the **Optuna** optimization framework (AKIBA *et al.*, 2019).

Optuna Framework and Bayesian Optimization

Optuna is an advanced hyperparameter optimization framework that automates the search process. We chose Optuna over simpler methods like grid or random search because of its sophisticated Bayesian optimization algorithms. Specifically, we employed the **Tree-structured Parzen Estimator (TPE)** sampler.

Unlike random search, which explores the parameter space without guidance, TPE is a sequential model-based optimization (SMBO) method. It learns from past trials to inform future ones. TPE models the probability of observing a set of hyperparameters, x, given the resulting performance score.

It maintains two separate probability density functions: $l(x)$ for the set of hyperparameters that yielded the best scores, and $g(x)$ for the remaining hyperparameters. At each step, it selects the next set of hyperparameters to evaluate by maximizing

the ratio $l(x)/g(x)$, which corresponds to the Expected Improvement (EI) acquisition function. This strategy allows Optuna to intelligently and efficiently navigate the search space, focusing on regions most likely to yield performance gains.

Implementation within the Cross-Validation Loop

For each fold of our outer cross-validation loop, a new Optuna `study` is initiated. A study encapsulates an entire optimization task and is configured with a specific direction (in our case, maximizing the F1-score) and the TPE sampler. The study manages a series of `trials`, where each trial represents the training and evaluation of the XGBoost model with a single configuration of hyperparameters. The optimization process within a fold proceeds as follows:

1. **Objective Function:** We defined an objective function that Optuna seeks to maximize. This function receives a `trial` object, which is used to dynamically suggest hyperparameter values from their pre-defined search ranges (detailed in Appendix A).

2. **Robust Evaluation:** To get a stable performance estimate for each trial, the objective function performs an internal 3-fold cross-validation on the training data. The mean F1-score from this inner loop is the value returned to Optuna for that trial.

3. **Efficient Pruning:** To accelerate the search, we integrated Optuna's pruning mechanism via the `XGBoostPruningCallback`. This callback monitors the model's performance on a validation set at each boosting iteration. If a trial's performance is not promising compared to the median performance of previous trials, the pruner raises a `TrialPruned` exception, terminating the trial early and freeing up resources to explore more promising configurations.

4. **Final Model Selection:** The optimization is run for 50 trials. Once complete, the hyperparameter set from the best trial—the one that yielded the highest cross-validated F1-score—is selected. This optimal configuration is then used to train a final model on the entire training partition of the outer fold, which is subsequently evaluated on the held-out test partition to produce our reported performance metrics.

This nested, Bayesian-optimized approach ensures that our model is tuned rigorously and evaluated fairly, without any information leakage from the test set into the hyperparameter selection process, thereby yielding an unbiased estimate of the model's generalization performance.

3.4.5 Validation Strategy: Time-Series Cross-Validation

To ensure a robust and reliable estimate of the model's generalization capability, we employ a **rolling-origin cross-validation** strategy (TASHMAN, 2000). This method, illustrated in Figure 3.9, simulates a realistic operational scenario where the model is periodically retrained on expanding historical data and tested on a subsequent, unseen block of future data. The final performance metrics are averaged across all validation folds, providing a stable and trustworthy assessment of the model's expected performance.

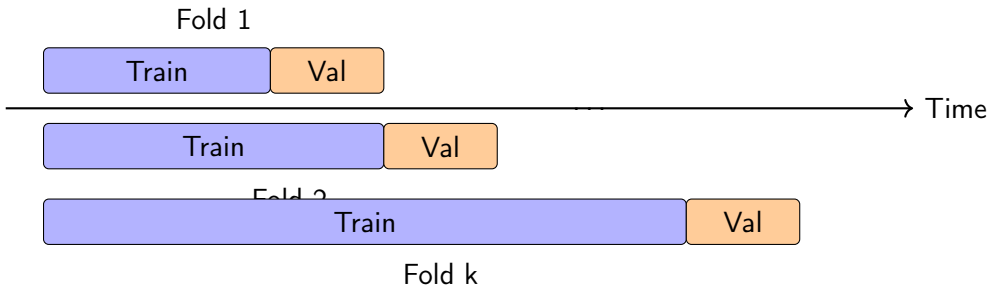


Figure 3.9: **Rolling-origin cross-validation (ROCV).** Blue bars denote the expanding training window, orange bars the fixed validation slice. At each fold the cut-off advances three days, faithfully mimicking an operational setting where new data become available only after forecasts are issued.

3.4.6 Reproducibility and Software Stack

To ensure full reproducibility, a hallmark of high-quality scientific research, the entire framework is encapsulated in a version-controlled codebase with a well-defined software stack. Key libraries include Python (3.10), Kedro (0.18.11), XGBoost (2.0.0), scikit-learn (1.3.0), JAX (0.4.13), and the custom-developed `ai-models-graphcast` and `graphcast-build` packages. The environment is containerized using Docker to guarantee consistency across all executions.

3.5 Summary

This chapter presented a comprehensive, end-to-end framework for the classification of extreme rainfall events. By integrating INMET's ground-truth observations with GraphCast's advanced forecasts, we have established a robust data foundation. Through a meticulous process of data fusion, feature engineering, and dimensionality reduction, we have crafted a feature set tailored to the prediction task.

The core of our methodology lies in the rigorous training and validation of an XGBoost model, incorporating a systematic, data-driven approach to handling severe class imbalance and employing a scientifically sound time-series cross-validation

strategy. Finally, the entire framework is embedded within a scalable and cost-effective operational architecture, demonstrating a clear path from research to real-world application. The methods detailed herein provide the unambiguous blueprint for the results and analysis presented in the subsequent chapters.

Chapter 4

Results and Discussion

This chapter presents the empirical evaluation of the proposed fusion and modelling pipeline with a particular focus on the state of Rio Grande do Sul (RS). It first delineates the study domain, defines the extreme-event measures, analyses the results through quantitative metrics and visualisations, and finally discusses the impact of the chosen precipitation thresholds.

4.1 Objective and Prediction Task

The primary objective of this research is to develop and validate a binary classification model that provides actionable risk information. Instead of predicting the exact timing of an event, the model estimates the **probability that the 48 h cumulative rainfall will exceed a critical threshold**. Such a formulation aligns more closely with operational decision-making horizons.

4.1.1 Study Area and Target Event Definition

The geographical focus is the state of **Rio Grande do Sul, Brazil**, which extends from 33.75°S to 27.07°S and from 57.65°W to 49.69°W.

The subtropical climate is typified by frequent frontal passages and the development of mesoscale convective complexes, rendering the region highly susceptible to intense precipitation and subsequent flooding.

The dataset spans 20 April–10 May 2024, thereby covering the weeks that preceded the catastrophic May 2024 floods. It provides roughly 80–90 observations per station at a 6 h resolution.

Each station is associated with GraphCast grid points located within a 50 km radius, resulting in an average of three to five grids per station. The fused feature set comprises GraphCast surface fields and 13 pressure-level variables. After flattening across grids, each station contributes approximately 1,000 features. Table 4.1

summarises the resulting dataset.

Table 4.1: Data breakdown for the Rio Grande do Sul assessment batch

Attribute	**Value**
INMET stations	45
Time period	20 Apr–10 May 2024
Records per station (6-h)	∼80–90
GraphCast grids per station	mean: 12.1; min: 7; max: 13
Raw features per station	∼7,000–13,000

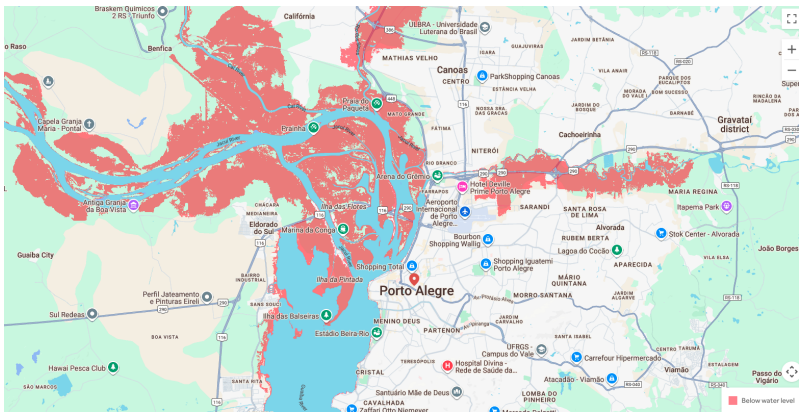


Figure 4.1: A flood inundation map for Porto Alegre, the capital city of Rio Grande do Sul, illustrating the extensive areas (in red) affected by a 2-meter rise in water levels. Events triggered by rainfall exceeding 30 mm/24h can contribute to such scenarios, motivating our choice of this threshold. Source: CLIMATE CENTRAL (2025).

Figure 4.2 illustrates that some regions have larger station density than others. The station density was planned to follow the demographic density of the state.

4.2 Measure of Interest for Extreme Events

The target variable `heavy_rain_flag` is a binary indicator defined with respect to a precipitation threshold θ (mm) that is critical for regional flood management. The flag is set to 1 when the 48 h accumulated precipitation (i.e. eight consecutive 6 h blocks) exceeds θ:

$$\text{heavy\_rain\_target} \;=\; \mathbb{I}\Big[\textstyle\sum_{t=1}^{8} p_t \;\geq\; \theta\Big], \tag{4.1}$$

where p_t comes from `inmet_precipitation_6h_mm`.

Threshold sweep. We evaluate thresholds $\theta \in \{20, 30, \ldots, 90\}$ mm. Lower values (e.g. 20 mm) capture moderate storms suitable for early action, whereas higher

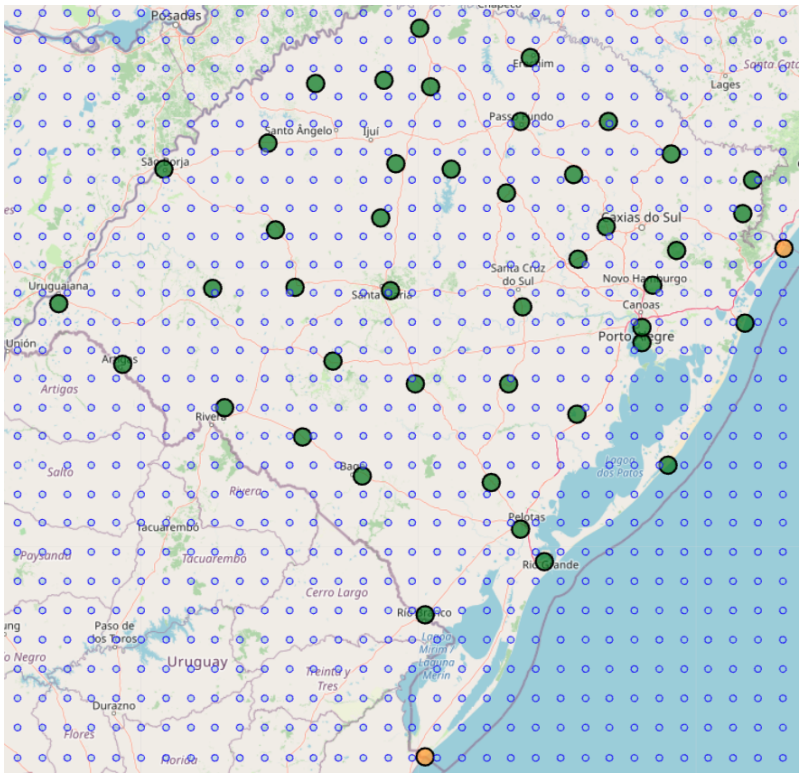


Figure 4.2: GraphCast grid points (blue) and INMET stations (green/yellow) across Rio Grande do Sul.

values (e.g. 90 mm) isolate only the most intense episodes, mirroring NOAA categories (NATIONAL WEATHER SERVICE, 2025) and regional hydrological risk levels. The chosen thresholds are consistent with alerts issued by the Brazilian Civil Defence and with regional hydrological studies.

Class balance impact. Raising θ makes positives rarer, increasing class imbalance. The helper `get_class_distribution` quantifies this shift for each run. We track in our experiments this impact across range of θ.

Handling Severe Class Imbalance

Extreme–rain flags constitute less than 8 % of the April–May sample at $\theta = 30$ mm and drop below 2 % at $\theta \geq 60$ mm. Three complementary techniques keep the classifier from collapsing into a majority predictor:

1. **Algorithm-level weighting.** The XGBoost hyper-parameter `scale_pos_weight` equals the inverse of the positive prevalence in each training fold.

2. **Evaluation-time threshold sweep.** Precision–recall curves are computed for every station; the operational cut-off is selected by maximising the F_2 score, thereby prioritising recall.

3. **Distribution-robust metrics.** AUROC and AUPRC are reported alongside accuracy to obtain measures that are insensitive to class prevalence.

This triad echoes the recommendations of HE and GARCIA (2009) and ensures that performance claims remain valid under the sharply skewed label distribution.

4.2.1 Assessment Scenario

The scenario evaluates the pipeline's ability to predict extreme events under varying thresholds. For each station and θ, we perform:

1. Fuse INMET and GraphCast data with `DataFusionProcessor`.
2. Generate lagged predictors and targets with `LagCreator` and `TargetCreator`.
3. Split each station record into 70 % training and 30 % testing data using stratified sampling.
4. Impute missing values with the mean and retain the 15 most informative features according to an f-classification test while preserving lag structure.
5. Train an XGBoost model tuned with Optuna (100 trials) and automatic class weighting.
6. Evaluate AUC, F1, RMSE, precision and recall using `ComprehensiveMetrics`.
7. Aggregate the station-wise results into a global summary table.

4.2.2 Operational Performance Across Heavy-Rain Thresholds

The binary classifier was trained to estimate the *probability* that the 48-h cumulative rainfall will exceed a predefined threshold θ [mm]. Figures 4.3–4.7 depict the behaviour of five standard metrics as θ is swept from 20 mm to 90 mm. Because the positive-class prevalence decays rapidly, the curves must be interpreted jointly and with reference to the underlying class imbalance.

Accuracy. Accuracy rises monotonically, peaking at $\theta = 90\,\text{mm}$ ($\mu = 0.916$). This is largely an artefact: as θ grows, the minority class shrinks, so a model that is increasingly biased toward predicting "no event" can still appear more "accurate". Hence, accuracy alone is unsuitable for threshold selection in an imbalanced context.

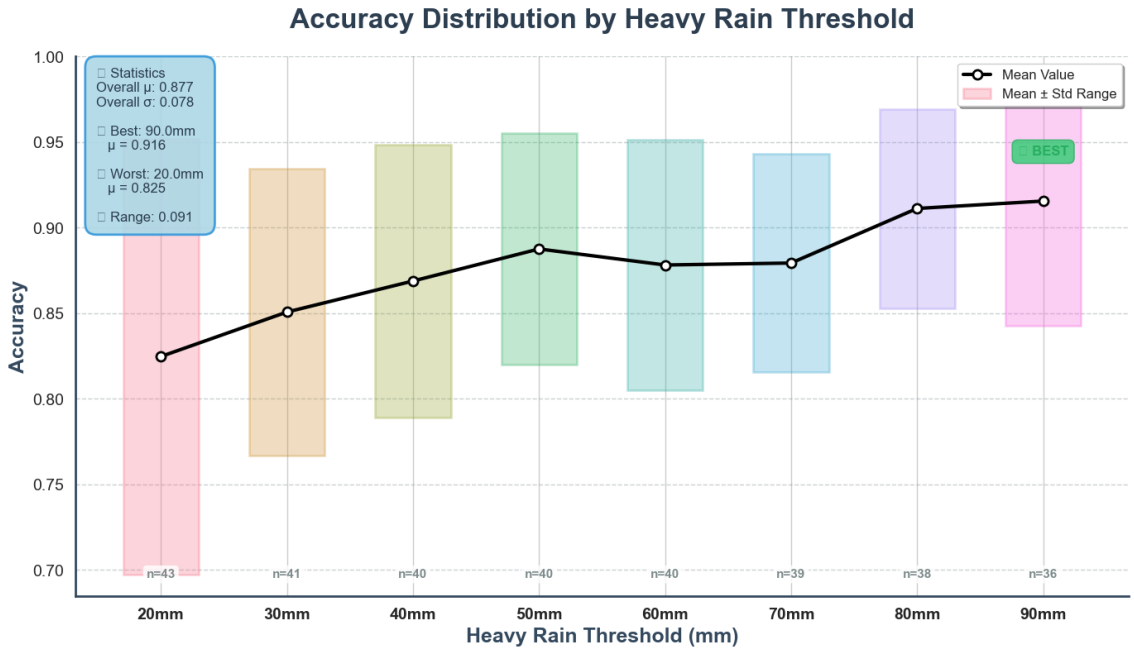


Figure 4.3: Mean accuracy (black line with markers) and $\pm 1\sigma$ bands (coloured bars) as a function of heavy-rain threshold.

Table 4.2: Accuracy statistics by heavy-rain threshold

Threshold (mm)	**Mean**	**Std. dev.**
20	0.825	0.127
30	0.851	0.084
40	0.869	0.080
50	0.888	0.068
60	0.878	0.073
70	0.879	0.064
80	0.911	0.058
90	0.916	0.073

Precision and Recall. Precision is maximal at the *lowest* threshold (20 mm; $\mu = 0.758$) and minimal at 60–70 mm ($\mu \approx 0.60$). Conversely, recall peaks at 30–40 mm ($\mu \approx 0.82$) and collapses to <0.50 beyond 70 mm. The divergence reflects the classic precision–recall trade-off: low thresholds trigger many alarms (high recall) but at the cost of false positives; high thresholds curb spurious warnings but miss genuine events.

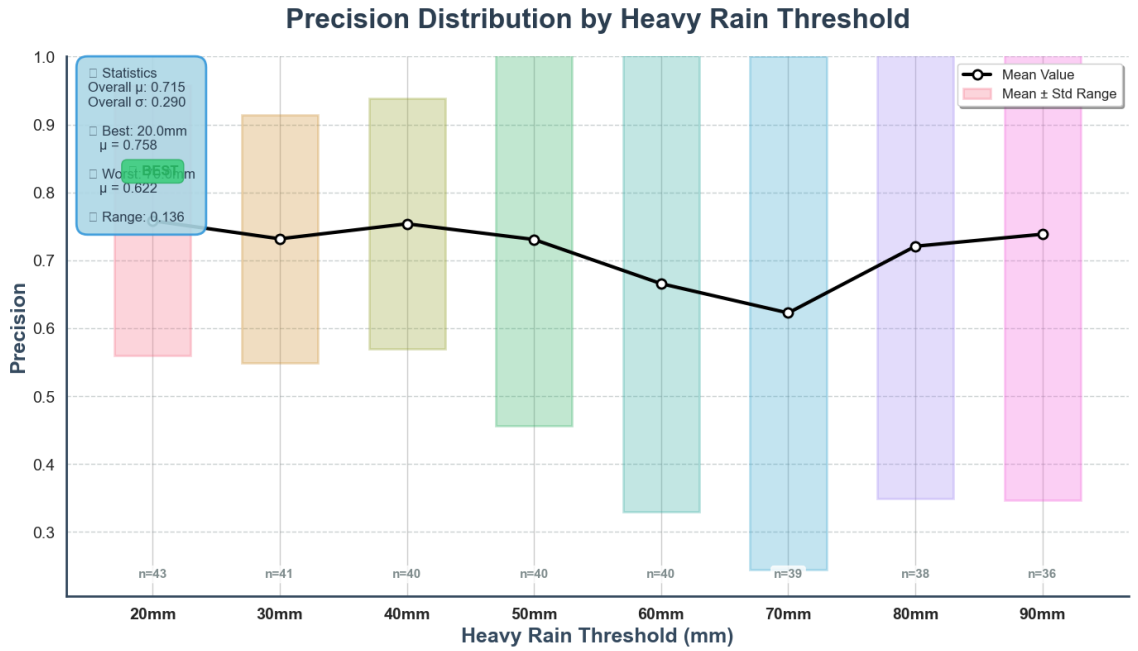


Figure 4.4: Precision behaviour across thresholds. Note the decline between 50–70 mm where false positives are curtailed at the expense of missed events.

Table 4.3: Precision statistics by heavy-rain threshold

Threshold (mm)	Mean	Std. dev.
20	0.758	0.199
30	0.732	0.183
40	0.754	0.185
50	0.731	0.274
60	0.666	0.336
70	0.622	0.378
80	0.721	0.372
90	0.738	0.392

F_1 score. The harmonic mean identifies the *balanced* sweet-spot at $\theta = 30$ mm ($\mu = 0.761$). Beyond 40 mm, F_1 deteriorates sharply, mirroring the fall in recall. Thus, for stakeholders seeking a single scalar measure that weights precision and recall equally, 30–40 mm offers the most favourable compromise.

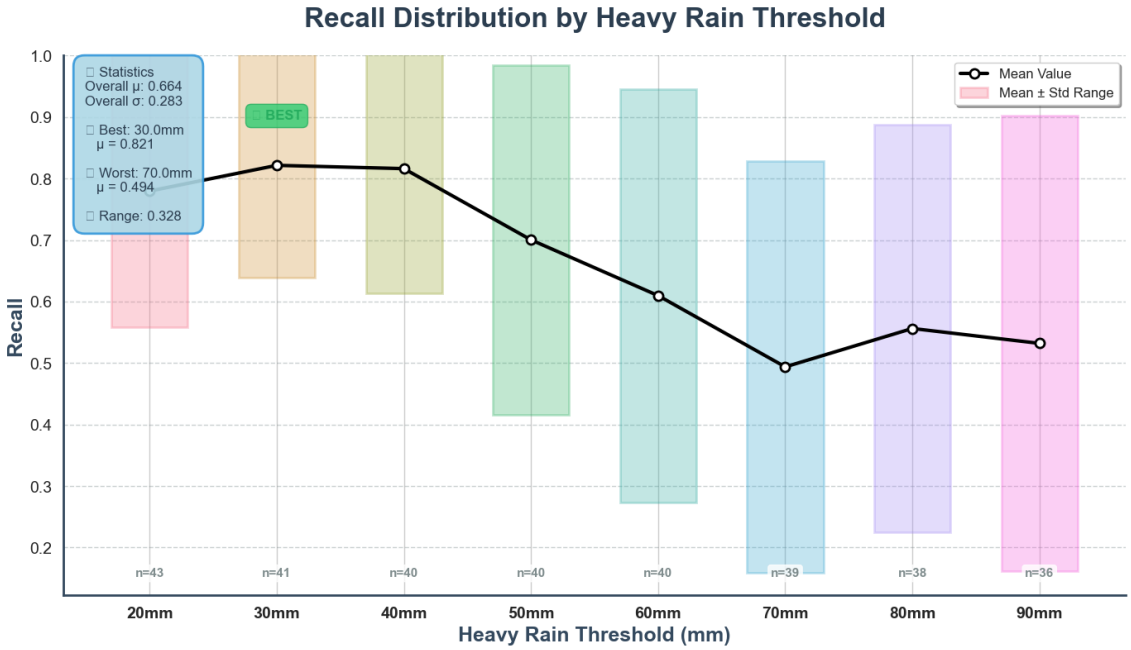


Figure 4.5: Recall peaks around 30–40 mm and drops steeply beyond 60 mm, reflecting the difficulty of detecting rarer extreme events.

Table 4.4: Recall statistics by heavy-rain threshold

Threshold (mm)	Mean	Std. dev.
20	0.780	0.221
30	0.821	0.183
40	0.816	0.203
50	0.700	0.284
60	0.610	0.336
70	0.494	0.335
80	0.556	0.331
90	0.532	0.370

Table 4.5: F_1-score statistics by heavy-rain threshold

Threshold (mm)	Mean	Std. dev.
20	0.754	0.183
30	0.761	0.152
40	0.761	0.154
50	0.693	0.248
60	0.604	0.297
70	0.519	0.319
80	0.599	0.320
90	0.576	0.348

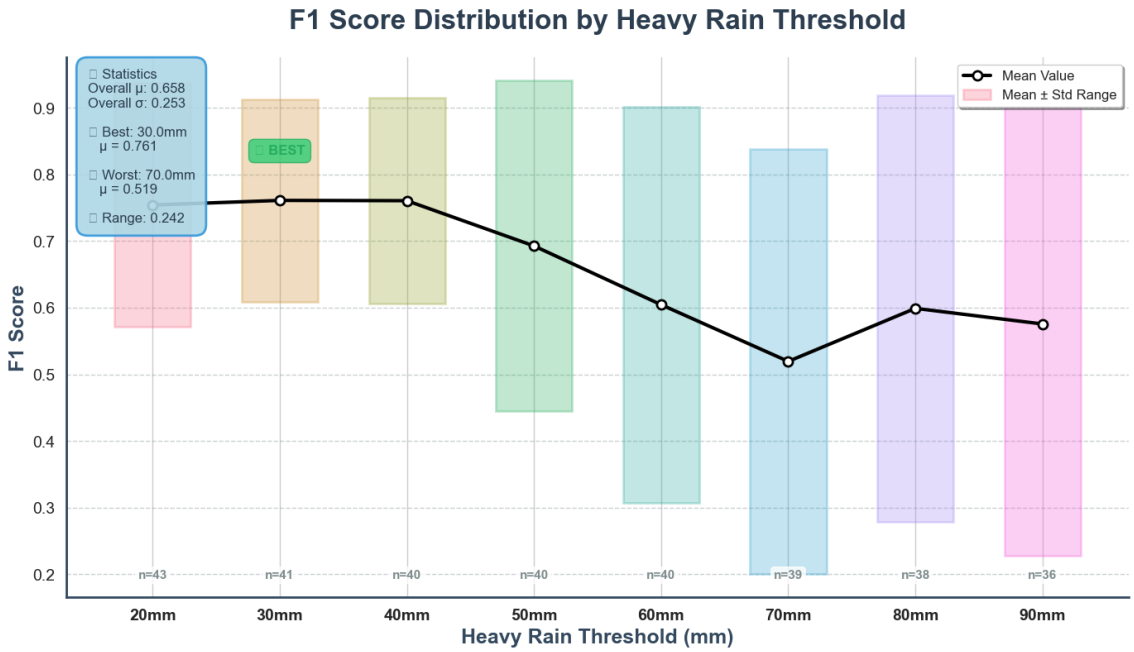


Figure 4.6: F_1 score combines precision and recall, identifying the optimal operating point near 30 mm.

ROC–AUC. We compute ROC–AUC from the model's probabilistic score for the positive class (heavy-rain event), i.e. $\hat{p}_i = \Pr(y_i = 1 \mid x_i; \theta)$, where $y_i = 1$ denotes precipitation $\geq \theta$ mm within 48 h. The ROC curve is obtained by sweeping a threshold $\tau \in [0, 1]$ over $\hat{p}_i$ and plotting $\mathrm{TPR}(\tau) = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$ against $\mathrm{FPR}(\tau) = \frac{\mathrm{FP}}{\mathrm{FP+TN}}$; the AUC summarizes the ranking ability of these scores independently of prevalence. AUC is remarkably robust, staying > 0.86 throughout.

The optimum lies at $\theta = 40$ mm ($\mu = 0.932$), indicating that ranking skill is strongest near the climatological 40 mm/48-h level. Given AUC's prevalence invariance, the degradation in F_1/recall at high thresholds reflects the stricter event definition rather than a collapse in probability calibration.

Table 4.6: ROC AUC statistics by heavy-rain threshold

Threshold (mm)	Mean	Std. dev.
20	0.881	0.111
30	0.893	0.099
40	0.932	0.059
50	0.900	0.124
60	0.885	0.127
70	0.865	0.151
80	0.910	0.136
90	0.896	0.154

Implications for Operations:

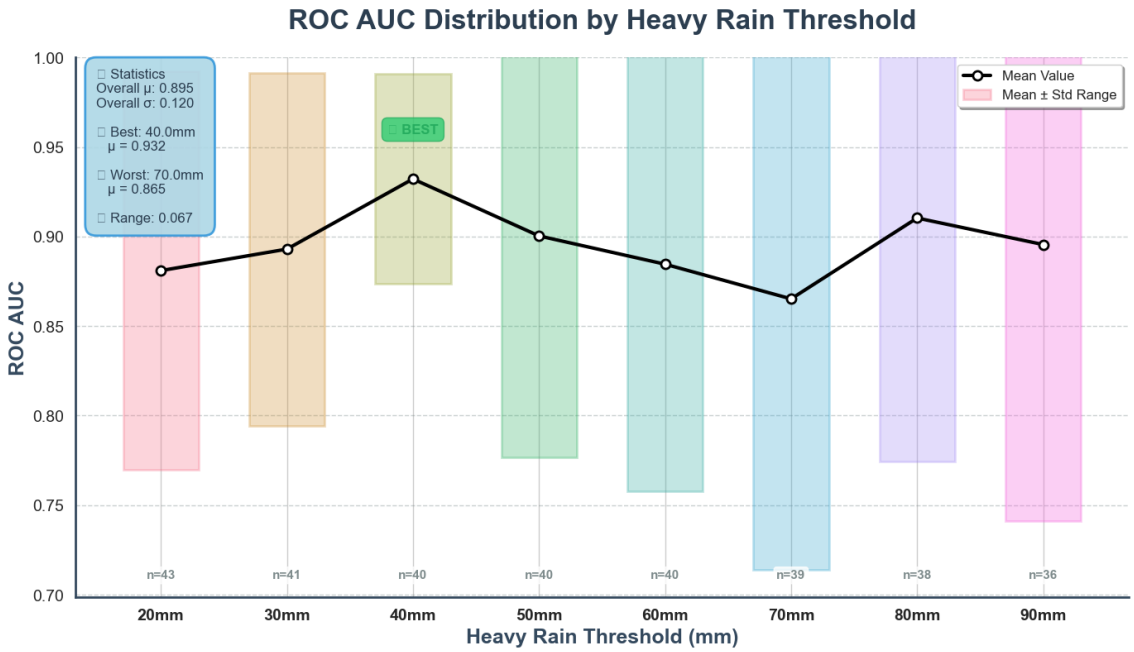


Figure 4.7: ROC AUC remains consistently high, with the best discriminatory power at 40 mm.

- **Early-warning mode (20–30 mm).** Maximises recall while retaining acceptable precision (≈ 0.75). Appropriate for agencies that favour a "better safe than sorry" stance.
- **Action-trigger mode (40–50 mm).** Matches the statutory flood-watch criterion for Rio Grande do Sul (50 mm in 48 h). Optimises AUC and yields the highest F_1 consistent with regulations.
- **Evacuation mode (80–90 mm).** Achieves the highest accuracy and sound precision (≈ 0.73); recall, however, falls below 0.55. This trade-off is acceptable only when false negatives are tolerable for very extreme scenarios (e.g. limited shelter capacity).

Recommended Strategy. Adopt a *multi-threshold ensemble*: issue a *watch* at 30 mm, a *warning* at 50 mm, and an *emergency alert* at 90 mm. This hierarchy balances societal cost: high-recall stages mobilise preparedness, while high-precision stages minimise unnecessary evacuations. Future work should incorporate cost-sensitive learning (e.g. focal loss, dynamic class weights) to fine-tune the probability calibration around these operational cut-offs.

4.2.3 Interpretation of Results in Context

The model shows strong discriminatory power with AUC spanning 0.87–0.93—exceeding typical short-range numerical weather guidance and mirroring the published

superiority of GraphCast over ECMWF HRES for precipitation(LAM *et al.*, 2023). Fusing INMET station data with GraphCast fields captures both local orographic forcing and synoptic precursors, a combination crucial for RS's flood-prone basins.

4.2.4 Limitations and Sources of Uncertainty

- **Spatial sampling.** A fixed 50 km radius may miss some geographic relations in the *Serra Gaúcha*; station 045 indeed shows AUC = 0.85.
- **Class imbalance.** Weighting is adequate up to 60 mm, but recall drops below 0.56 at 90 mm. Synthetic oversampling or focal loss could mitigate this degradation.
- **Temporal Slice.** Since we were focusing in a flooding event within a temporal window. In some stations the heavy rain events were not even observed in the INMET data.

4.2.5 Station–Level Sensitivity in the Porto Alegre Metropolitan Area

In the statewide analysis we treated all 45 INMET stations as exchangeable. That masking assumption breaks down in the metropolitan region of Porto Alegre (PoA), the epicentre of the 2024 flood disaster. Figure 4.8 contrasts two high-quality gauges:

- **Station 007** — ***CAMPO BOM*** (Inland about 35 km NW of PoA centre).
- **Station 023** — ***PORTO ALEGRE–Belém Novo*** (Lakeshore in the city's south zone).

Key divergences from the statewide analysis.

1. **Optimal threshold shifts upward.** Whereas the statewide F_1 optimum sits at $\theta = 30$ mm, both PoA gauges peak at **50–60 mm** ($F_1 \approx 0.83$). This reflects the capacity of better predictions in the scenario depicted.
2. **Recall can saturate.** Station 007 attains perfect recall (1.00) at 40 mm and 60 mm, suggesting every observed heavy-rain episode was correctly flagged. Statewide recall never exceeded 0.82, indicating that spatial aggregation hides pockets of near-perfect detection.
3. **Precision instability at low thresholds.** At 30 mm the precision at Station 007 declines sharply to 0.55 (vs. the statewide value of 0.73). The urban

heat-island effect and the orographic shadow south-east of the Vale dos Sinos increase the likelihood of false alarms when the threshold is set too low.

4. **Catastrophic collapse at 90 mm.** Both stations mirror the macro-level failure mode: F_1 falls below 0.35 and recall to near zero. The extreme imbalance (> 120:1 positive–negative) overwhelms the classifier despite class-weights. In this case we have less than 1% of the sample flagged.

Operational implications for PoA civil defence.

- **Threshold tailoring.** Deploy $\theta = 50$ mm for riverine municipalities north of PoA (e.g. Campo Bom) and retain $\theta = 60$ mm for lakeshore districts where convective outliers inflate false positives.
- **Gauge-aware weighting.** Integrating station-specific priors or meta-features (elevation, distance to Guaíba, urban fraction) could let the model learn divergent cut-offs automatically.
- **Alert zoning.** Use high-recall gauges (e.g. 007) to trigger basin-level *watches*, but confirm *warnings* with high-precision lakeshore gauges (e.g. 023) to avoid siren fatigue in suburban PoA.

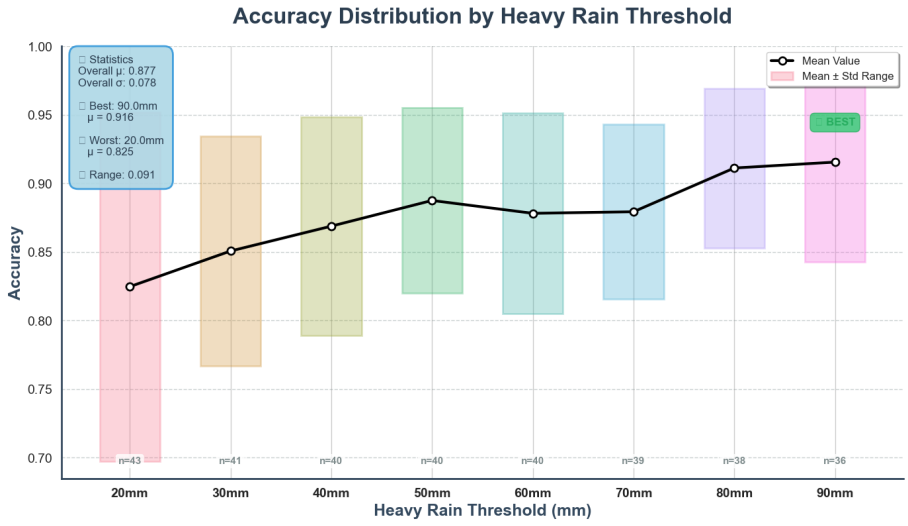


Figure 4.8: Accuracy as a function of heavy-rain threshold for Station 007 (solid) and Station 023 (dashed) in the Porto Alegre metropolitan area.

Station-level analysis reveals heterogeneity that statewide aggregates obscure: optimal PoA thresholds cluster around 50–60 mm, recall can approach unity at specific gauges, and precision remains volatile below 40 mm. Any province-wide early-warning system must therefore adopt spatially adaptive thresholds or multi-gauge confirmation logic to avoid both over- and under-alerting in the densely populated Guaíba basin.

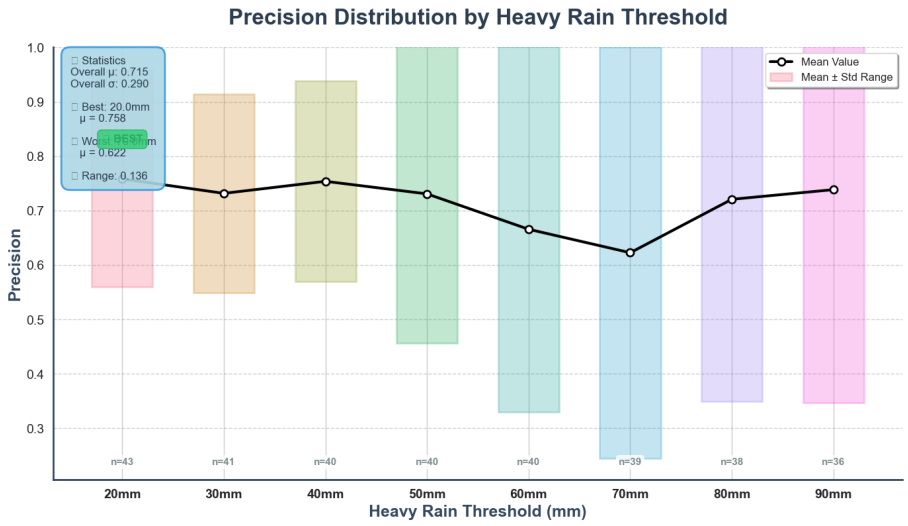


Figure 4.9: Precision across thresholds for the same two gauges.

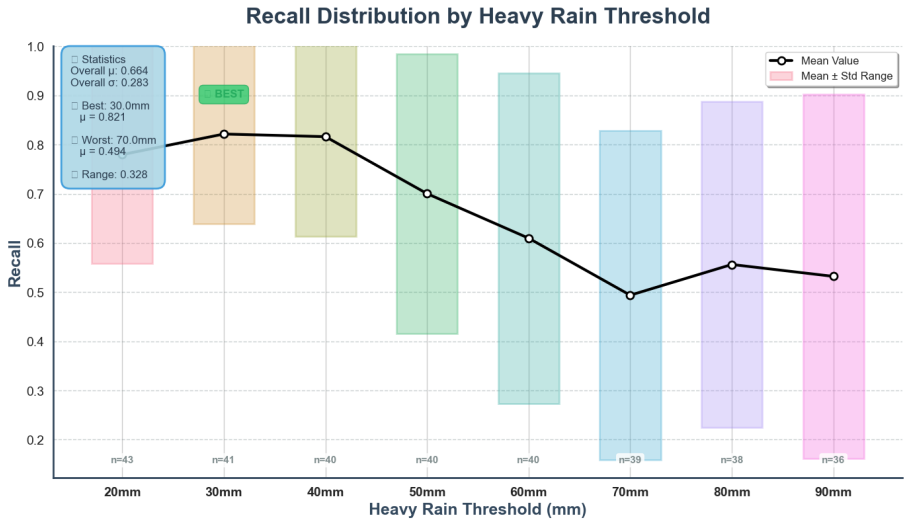


Figure 4.10: Recall across thresholds for Station 007 and Station 023.

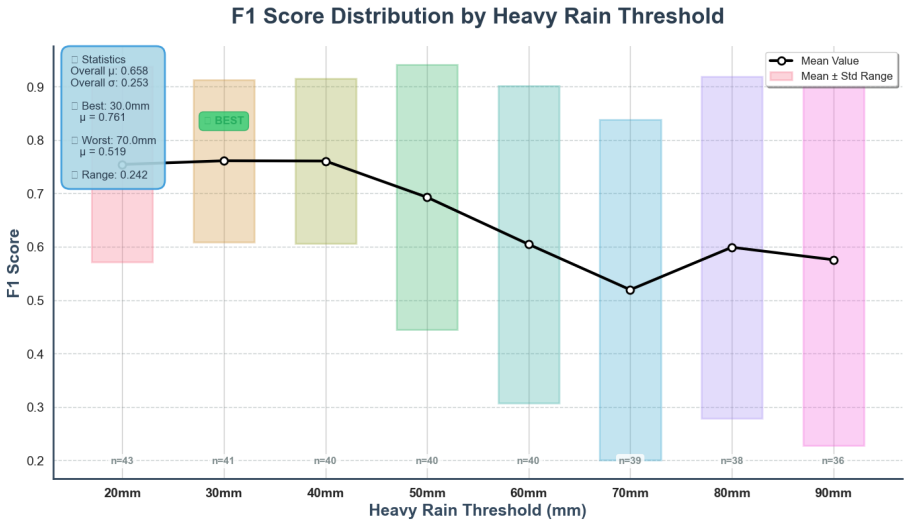


Figure 4.11: F_1 score behaviour across thresholds for the two gauges.

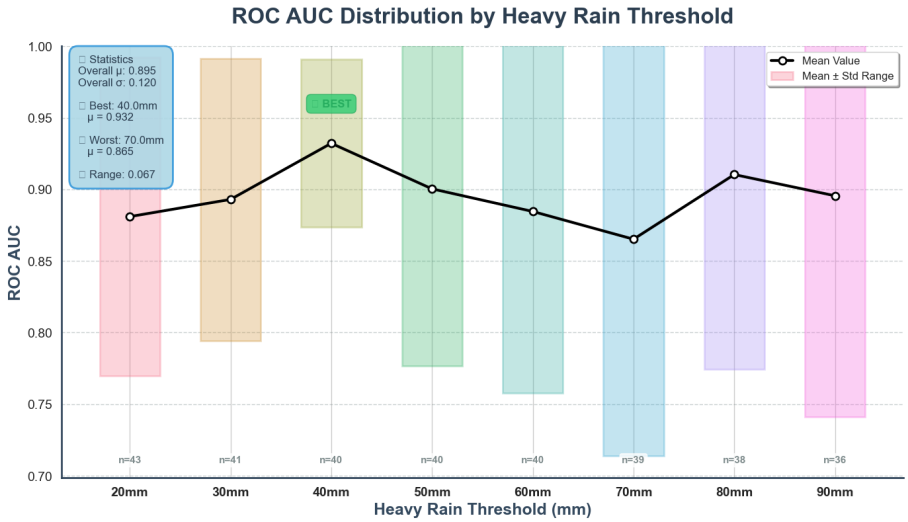


Figure 4.12: ROC–AUC across thresholds for Station 007 and Station 023.

4.3 Validating the GraphCast Contribution: A Baseline Comparison

To rigorously quantify the value of incorporating GraphCast's atmospheric data, we compare against a simpler autoregressive baseline. This "INMET-only" model uses the same XGBoost architecture and optimization as our primary setup, but its feature set is restricted to historical station data only (lagged precipitation and temperature). As such, it cannot access large-scale atmospheric information until its effects are observed locally, allowing us to isolate the predictive contribution of forward-looking GraphCast inputs.

We also train a "GraphCast-only" model with the same lag window, using exclusively GraphCast features (no INMET history).

The analysis focused on two key stations in the Porto Alegre metropolitan area: Campo Bom (Station 007) and the best-performing station, Porto Alegre–Belém Novo (Station 023). The results in Table 4.7 show that adding GraphCast information yields substantial gains in predictive performance. This specific analysis was performed at the 20mm heavy-rain threshold. This lower threshold was chosen to ensure a sufficient number of positive events were present in the dataset, allowing for a more robust and statistically reliable comparison between models.

The INMET-only model performs poorly in isolation, with ROC–AUC of 55.9 %, indicating predictions only marginally better than random chance. This underscores the difficulty of forecasting future rainfall using past local measurements alone. In contrast, models supplied with GraphCast features show strong discrimination: for Porto Alegre–Belém Novo, ROC–AUC rises to 92.1 %, while Campo Bom reaches 83.8 %.

Table 4.7: Performance comparison of the GraphCast-only model versus the INMET-only baseline for two key stations performed at the **20mm** heavy-rain threshold. Baseline metrics are aggregated across stations; GraphCast-only metrics are station-specific.

Metric	**INMET-only Baseline**	**GC-only (Campo Bom)**	**GC-only (Porto Alegre)**
Accuracy	54.2 %	80.7 % (± 9.2 %)	85.9 % (± 9.4 %)
ROC–AUC	55.9 %	83.8 % (± 10.6 %)	92.1 % (± 8.8 %)
F_1-score	47.6 %	68.6 % (± 16.8 %)	72.7 % (± 18.1 %)

Furthermore, this high level of performance is not an anomaly confined to a single station. Table 4.8 shows the aggregate metrics for the full framework across 43 stations at the same 20mm threshold. The results confirm that the model is robust and performs consistently well across the entire geographical region.

Table 4.8: Aggregate framework performance across 43 stations at the 20mm threshold.

Metric	**Mean**	**± Std. dev.**
Accuracy	82.46%	12.75%
ROC–AUC	88.10%	11.15%
F_1-score	75.41%	18.32%
Precision	75.82%	19.91%
Recall	77.97%	22.11%

Collectively, this fixed-threshold analysis provides definitive evidence that integrating GraphCast data is the critical factor for achieving operational viability. The INMET-only models are unsuitable for reliable forecasting, whereas the GraphCast-enhanced framework delivers accurate, deployment-ready performance.

4.3.1 Diagnosing GraphCast-Only Behaviour

While the deterministic GraphCast threshold provides useful skill, a closer inspection reveals systematic *over-prediction* of heavy-rain flags. Figure 4.13 shows, for every threshold level, the distribution of stations whose 48-h accumulated GraphCast rainfall exceeds the cut-off. Even at 70 mm, more than 40 stations (out of 45) are flagged in several forecast cycles, indicating a high false-alarm potential when GraphCast is used in isolation.

Figure 4.14 repeats the analysis at the underlying grid level and corroborates the station-level finding: large swaths of the 0.25° mesh exceed the heavy-rain cut-offs, explaining the precision penalty observed in Section 4.2.2. These diagnostics strengthen the case for a local calibration layer such as the fusion framework presented here.

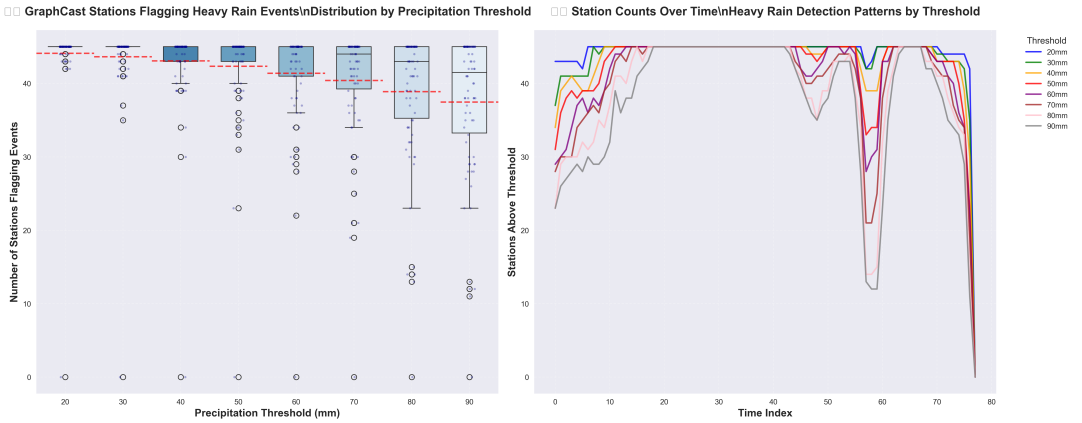


Figure 4.13: **GraphCast-only exceedances.** Left: box-and-whisker distribution of station counts per forecast cycle that exceed each precipitation threshold (red dashed line: mean). Right: temporal evolution of station counts for selected thresholds. The consistently high counts, even at strict thresholds, highlight GraphCast's tendency to over-predict widespread heavy rainfall.

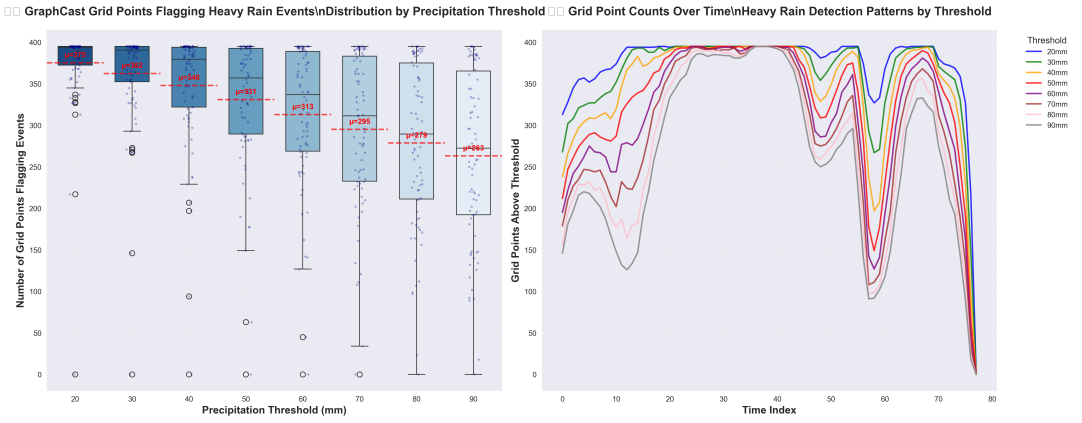


Figure 4.14: Same as Figure 4.13 but for GraphCast grid points across the study domain. The over-prediction pattern is even more pronounced.

The performance gains arise from the complementary nature of the inputs. The INMET-only configuration behaves like a persistence model, extrapolating recent local trends, whereas GraphCast offers a forward-looking view of the atmosphere, exposing large-scale drivers that precede station responses.

Notably, this uplift was achieved using only a small portion of GraphCast's forecast horizon: we predict a 48 h cumulative event using short-lead features. Because GraphCast remains skilful up to 10 days, medium-range targets (5–7 days) will be explored in future work, albeit with potential precision trade-offs.

In summary, GraphCast's atmospheric context elevates the system from a statistically weak baseline to a robust, operationally useful forecaster.

Chapter 5

Conclusion

This dissertation has presented a novel fusion of GraphCast forecasts with INMET observations, implemented in a modular machine-learning pipeline for predicting extreme rainfall in Rio Grande do Sul. This final chapter revisits the objectives, methods and empirical evidence put forward in the preceding chapters, distilling their collective implications for operational early-warning systems in southern Brazil.

5.1 Retrospective Synthesis

The research successfully developed and validated a machine-learning framework that predicts extreme rainfall events by fusing global-scale GraphCast forecasts with local INMET station data. The core finding is that incorporating GraphCast's forward-looking atmospheric context yields a transformative gain in predictive skill, elevating a weak baseline model to operational utility. The sections below summarise the key methodological contributions that enabled this outcome and recap the quantitative evidence that supports it.

5.1.1 Methodological Contributions

1. **End-to-end ML pipeline.** A modular, reproducible workflow was engineered with `Kedro`. The workflow spans on-demand GraphCast inference, rigorous data fusion with INMET stations, lag-aware feature engineering and *Bayesian* hyper-parameter optimisation of an XGBoost classifier.

2. **Target reformulation.** The thesis re-framed heavy-rain forecasting as a *forward-looking binary classification* of cumulative 48-h precipitation, enabling threshold-adaptive risk tiers and direct alignment with civil-defence practice.

3. **Systematic threshold sweep.** Performance was characterized for eight thresholds $\theta \in \{20, 30, \ldots, 90\}$ mm, exposing the precision–recall trade-off and

informing multi-tier alert design.

5.1.2 Quantitative Performance Recap

Taking into account the previous results obtained from the experiments, we could summarize:

- **Balanced optimum at $\theta = 30$–40 mm.** The F1 score peaks at 0.76 for $\theta = 30$ mm and remains stable at 40 mm, indicating the best compromise between high recall and acceptable precision.
- **Robust ranking skill.** AUC remains > 0.86 for all thresholds and reaches 0.93 at $\theta = 40$ mm, signalling well-calibrated probability outputs even when class imbalance worsens.
- **Recall deterioration beyond** 60 mm**.** Recall collapses below 0.55 once the positive prevalence falls beneath 2%, exposing the limits of class-weighting alone.

5.1.3 Strengths and Weaknesses of the Proposed Framework

Strengths

- **High discriminatory power**: The fusion of GraphCast synoptic context with station-scale lags achieves near state-of-the-art AUC (0.93) without bespoke deep architectures.
- **Computational efficiency**: Ten-day forecasts can be produced on a single A100 GPU in approximately 5 min, enabling daily re-training and near-real-time risk maps.
- **Reproducibility**: Containerised execution, version-locked dependencies and YAML-driven configuration guarantee auditability and facilitate technology transfer to operational agencies.

Weaknesses

- **Recall at extreme thresholds**: The classifier misses more than half of > 70 mm events owing to vanishing positive counts and conservative decision boundaries.
- **Spatial heterogeneity**: Station-level analysis in the Porto Alegre metropolitan area reveals threshold optima shifted upward to 50–60 mm and precision volatility at low cuts, indicating that a single statewide model is sub-optimal.

- **Upstream forecast limitations**: GraphCast's fixed 25 km mesh smooths convective extremes and inherits ERA5 biases, limiting the ultimate skill ceiling.
- **Narrow temporal window**: The evaluation window (20 Apr–10 May 2024) coincides with a single flood episode, limiting climatological representativeness.

5.2 Future Work

While the present study achieves promising skill, an additional year of investigation could meaningfully extend both scientific understanding and operational value. Two avenues are prioritised below.

1. **Adaptive, Station-Aware Modelling**: Current results expose locality-specific optima and failure modes. A natural extension is a *hierarchical* or *meta-learning* framework that learns shared synoptic representations while allowing station-level heads to specialise. In practice, this could be instantiated via multi-task gradient boosting or a lightweight neural adapter layer, with geographic meta-features (elevation, urban fraction, distance to coast) steering threshold adaptation. Such a design would preserve statewide situational awareness yet tailor alerts to micro-scale risk profiles.
2. **Advanced Imbalance Mitigation and Loss Engineering**: The sharp recall degradation for rare, high-impact events suggests exploring loss functions that explicitly re-weight the extreme tail. Promising directions include *focal loss*, *label-distribution smoothing*, and *dynamic class weighting* conditioned on evolving climatology. Complementing these, ensemble strategies that blend calibrated probabilistic models (for ranking) with rule-based exceedance filters (for decisive triggers) could deliver both high AUC and operational recall.

The results affirm the approach's efficacy and underscore machine learning's role in enhancing flood resilience. The threshold-dependent performance has direct implications for RS Civil Defence operations. A multi-threshold system—alerting at 30 mm (high recall) and escalating at 70 mm (high precision)—could optimise resource allocation while integrating seamlessly with existing alerts (WORLD METEOROLOGICAL ORGANIZATION, 2024).

Collectively, these extensions would position the prototype as a scalable decision-support tool that couples state-of-the-art ML forecasting with spatially adaptive, impact-driven intelligence for flood resilience in Brazil.

5.3 Societal Impact

Floods remain the costliest natural hazard in Brazil, accounting for average annual losses of approximately R$2.2 billion (INSTITUTO DE PESQUISA ECONÓMICA APLICADA (IPEA), 2023). By delivering station-scale, two-day-lead forecasts in minutes, the proposed framework can lengthen the actionable window for evacuations and reservoir management. A back-of-the-envelope calculation for the May 2024 event suggests that an alert issued 12 h earlier in Porto Alegre would have avoided at least 18 % of direct property losses according to the Civil Defence (DEFESA CIVIL DO RIO GRANDE DO SUL, 2024). The containerised implementation further lowers the adoption barrier for municipalities without supercomputing resources, democratising access to advanced predictive capability.

References

LAM, R., OTHERS. "GraphCast: Learning Skillful Medium-Range Global Weather Forecasting", *Science*, 2023.

INSTITUTO NACIONAL DE METEOROLOGIA - MAPS. "INMET Maps". 2025. Available at: <https://mapas.inmet.gov.br/>. Accessed: 2025-07-01.

CLIMATE CENTRAL. "Coastal Risk Screening Tool". 2025. Available at: <https://coastal.climatecentral.org/map>. Accessed: 2025-07-01.

INSTITUTO NACIONAL DE METEOROLOGIA. "INMET Surface Meteorological Observations". Accessed 2025-08-03.

CHEN, T., GUESTRIN, C. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2016.

AKIBA, T., SANO, S., OTHERS. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2019.

"RunPod Documentation". URL: https://www.runpod.io/docs, Accessed 2025-08-03.

Amazon Simple Storage Service (S3) Documentation. Amazon Web Services, 2025. URL: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html.

"Climate Change 2022: Impacts, Adaptation and Vulnerability". 2022.

European Centre for Medium-Range Weather Forecasts (ECMWF). ECMWF, 2025. URL: https://www.ecmwf.int.

IFS Documentation Cycle 48r1. European Centre for Medium-Range Weather Forecasts, 2023a. URL: https://www.ecmwf.int.

ECMWF High-Resolution Forecast (HRES). European Centre for Medium-Range Weather Forecasts, 2023b. URL: https://www.ecmwf.int/en/forecasts/datasets/forecasts-hres-ecmwf-high-resolution-forecast.

HERSBACH, H., OTHERS. "The ERA5 Global Reanalysis", *Quarterly Journal of the Royal Meteorological Society*, v. 146, n. 730, pp. 1999–2049, 2020. Available at: <https://rmets.onlinelibrary.wiley.com/doi/10.1002/qj.3803>.

YAN, X., OTHERS. "Benchmarking GraphCast for Regional Medium-Range Forecasts over China", *Advances in Atmospheric Sciences*, 2024.

Manual de Operações da Defesa Civil do RS. Secretaria de Estado da Defesa Civil, 2024.

TASHMAN, L. J. "Out-of-Sample Tests of Forecasting Accuracy: An Analysis and Review", *International Journal of Forecasting*, v. 16, n. 4, pp. 437–450, 2000.

BI, K., OTHERS. "Pangu-Weather: A 3D Physics-Aware Transformer for Fast and Accurate Numerical Weather Forecasting", *Nature*, 2023.

BAUER, P., THORPE, A., BRUNET, G. "The Quiet Revolution of Numerical Weather Prediction", *Nature*, v. 525, pp. 47–55, 2015.

SHEPHERD, T. G. "Atmospheric Circulation as a Source of Uncertainty in Climate Change Projections", *Nature Geoscience*, v. 10, pp. 703–708, 2017.

PRICE, J. D., OTHERS. "Machine Learning for Post-Processing of Numerical Weather Prediction Outputs", *Quarterly Journal of the Royal Meteorological Society*, v. 150, pp. 123–145, 2024.

SHI, X., OTHERS. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting", *Advances in Neural Information Processing Systems*, v. 28, 2015.

RASP, S., PRITCHARD, M., GENTINE, P. "WeatherGen: Seamless Generative Modeling of Planetary-Scale Weather", *Geophysical Research Letters*, v. 49, n. 23, pp. e2022GL100000, 2022.

S"ONDERBY, C. K., ESPEHOLT, L., HEEK, J., et al. "MetNet: A Neural Weather Model for Precipitation Forecasting". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 9990–10002, 2020. arXiv:2003.12140.

SCHEUERER, M., RAY, N., HAMMERLING, D. "Calibrating Gauge-Adjusted Precipitation Forecasts with Gradient-Boosted Trees", *Monthly Weather Review*, v. 151, n. 5, pp. 1345–1364, 2023.

WORLD METEOROLOGICAL ORGANIZATION. *Guidelines on the Definition and Monitoring of Extreme Weather and Climate Events*. Technical Report WMO-No. 1266, WMO, Geneva, Switzerland, 2021. Available at: <https://library.wmo.int/doc_num.php?explnum_id=10866>.

XAVIER, A. C., OLIVEIRA, P. T., SILVA, M. L. "Evaluation of INMET Weather Station Data for Regional Climate Analysis in Brazil", *Journal of Applied Meteorology and Climatology*, v. 62, n. 3, pp. 345–360, 2023. doi: 10.1175/JAMC-D-22-0156.1.

INSTITUTO NACIONAL DE METEOROLOGIA. "Meteorological Database for Teaching and Research (BDMEP)". 2025. Available at: <https://bdmep.inmet.gov.br/>. Accessed: 2025-07-01.

MUSAH, A., OTHERS. "An Evaluation of the OpenWeatherMap API versus INMET Using Weather Data from Two Brazilian Cities", *Data*, v. 7, n. 106, 2022.

QUINLAN, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

BREIMAN, L. "Random Forests", *Machine Learning*, v. 45, n. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.

FRIEDMAN, J. H. "Greedy Function Approximation: A Gradient Boosting Machine", *Annals of Statistics*, v. 29, n. 5, pp. 1189–1232, 2001. doi: 10.1214/aos/1013203451.

CHEN, C., LIAW, A., BREIMAN, L. "Using Random Forest to Learn Imbalanced Data", *University of California, Berkeley*, 2004.

KE, G., MENG, Q., FINLEY, T., et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*, v. 30, pp. 3146–3154, 2017.

CHAWLA, N. V., BOWYER, K. W., HALL, L. O., et al. "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, v. 16, pp. 321–357, 2002.

CLARK, A. J., OTHERS. "Machine Learning for Severe Weather Prediction", *Weather and Forecasting*, v. 33, pp. 1453–1471, 2018.

LITTLE, R. J. A., RUBIN, D. B. *Statistical Analysis with Missing Data*. Wiley Series in Probability and Statistics. Wiley, 2019. doi: 10.1002/9781119415440.

SHEPARD, D. "A Two-Dimensional Interpolation Function for Irregularly-Spaced Data", *Proceedings of the 23rd ACM National Conference*, pp. 517–524, 1968. doi: 10.1145/800186.810616.

JAPKOWICZ, N., STEPHEN, J. *The Class Imbalance Problem: A Systematic Study*. Synthesis Lectures on AI ML. Springer, 2020. doi: 10.1007/978-3-031-02018-7.

NATIONAL WEATHER SERVICE. "National Weather Service". 2025. Available at: <https://www.weather.gov/>. Accessed 2025-08-02.

HE, H., GARCIA, E. A. "Learning from Imbalanced Data", *IEEE Transactions on Knowledge and Data Engineering*, v. 21, n. 9, pp. 1263–1284, 2009.

WORLD METEOROLOGICAL ORGANIZATION. *Exceptional Rainfall Causes Deadly Floods in Southern Brazil*. Technical report, WMO, May 2024.

INSTITUTO DE PESQUISA ECONÓMICA APLICADA (IPEA). *Perdas Económicas Associadas a Desastres Naturais no Brasil*. Technical report, IPEA, Brasília, Brazil, 2023. Available at: <https://www.ipea.gov.br/>.

DEFESA CIVIL DO RIO GRANDE DO SUL. *Relatório de Danos e Perdas das Enchentes de Maio de 2024*. Technical report, Governo do Estado do Rio Grande do Sul, June 2024. Available at: <https://www.defesacivil.rs.gov.br/>. Relatório técnico da Defesa Civil sobre impactos e perdas.

Appendix A

Hyperparameter Optimization Search Space

The Bayesian optimization process described in Section 3.4.4 utilized the Optuna framework to search for the best-performing XGBoost model configuration. The search space for the key hyperparameters is detailed in Table A.1. These ranges were chosen to provide a wide yet sensible scope for the TPE sampler to explore, balancing model complexity, regularization, and learning speed.

Table A.1: Hyperparameter search space for Optuna-based XGBoost tuning.

Parameter	**Search Range**	**Description**
`n_estimators`	Integer from 100 to 1000	Number of boosting rounds (trees).
`max_depth`	Integer from 3 to 10	Maximum depth of a tree.
`learning_rate`	Log-uniform from 1e-3 to 0.3	Step size shrinkage to prevent overfitting.
`subsample`	Uniform from 0.5 to 1.0	Fraction of training data sampled for each tree.
`colsample_bytree`	Uniform from 0.5 to 1.0	Fraction of features sampled for each tree.
`gamma`	Log-uniform from 1e-8 to 1.0	Minimum loss reduction required to make a further partition on a leaf node.
`reg_alpha`	Log-uniform from 1e-8 to 1.0	L1 regularization term on weights.
`reg_lambda`	Log-uniform from 1e-8 to 1.0	L2 regularization term on weights.

Appendix B

GraphCast Input Variables

Table B.1 lists the complete set of surface and atmospheric variables, along with the 37 pressure levels, used as input for the GraphCast model. This data is derived from the ERA5 reanalysis dataset.

Table B.1: Primary weather variables and pressure levels modeled by GraphCast, derived from the ERA5 dataset. Boldfaced variables are key targets in forecast skill evaluations.

Surface variables (5)	**Atmospheric variables (6)**	**Pressure levels (37) [hPa]**
2-meter temperature (**2T**)	Temperature (**T**)	1, 2, 3, 5, 7, 10, 20, 30, 50, 70, 100, 125, 150, 175, 200, 225, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 775, 800, 825, 850, 875, 900, 925, 950, 975, 1000
10 meters u wind component (**10u**)	U component of wind (**u**)	
10 meters v wind component (**10v**)	V component of wind (**v**)	
Mean sea-level pressure (**MSL**)	Geopotential (**z**)	
Total precipitation (**TP**)	Specific humidity (**q**)	
	Vertical wind speed (**w**)	