electronics

MDPI

Article

A Low-Cost IoT Architecture for Micro-Zone Climate Prediction and Meteorological Forecasting

César A. G. Mateus [1,2], Darlan Noetzold [1], Juan M. B. Skolik [1,2], Valderi R. Q. Leithardt [1,3,4,*] and Juan F. De Paz [1]

[1] Expert Systems and Applications Laboratory, University of Salamanca, 37008 Salamanca, Spain; id00771251@usal.es (C.A.G.M.); darlannoetzold@usal.es (D.N.); jskolik@uniempresarial.edu.co (J.M.B.S.); fcofds@usal.es (J.F.D.P.)
[2] Fundación Universitaria de la Cámara de Comercio de Bogotá (Uniempresarial), Bogotá 110311, Colombia
[3] Center of Technology and Systems (CTS-UNINOVA), Associated Laboratory of Intelligent Systems (LASI), 2829-516 Lisbon, Portugal
[4] ISTAR, University Institute of Lisbon (ISCTE-IUL), 1649-026 Lisbon, Portugal
* Correspondence: valderi.leithardt@gmail.com

Abstract

This article presents the design and deployment of ClimaBogotá v1.2, a climate prediction system tailored for high-altitude urban micro-zones in Bogotá, Colombia. The system combines low-cost IoT sensing, machine learning modeling, and cloud-based orchestration to enable scalable and affordable meteorological forecasting. Its architecture comprises Raspberry Pi-based weather stations, a Random Forest model trained on engineered temporal features, and an n8n-driven automation pipeline for real-time inference and dissemination via Telegram, PostgreSQL, and Grafana. With a Mean Absolute Error of 2.59 °C and an R^2 of 0.6286 on a 30 min forecast horizon, the system demonstrates both predictive reliability and operational feasibility using free-tier cloud resources. Unlike traditional weather systems, ClimaBogotá emphasizes modularity, adaptability, and cost-efficiency, offering a replicable framework for decentralized climate monitoring in data-scarce urban environments. Temporal misalignment between sensor nodes was identified as the primary constraint, informing future enhancements toward distributed learning strategies.

Keywords: climate prediction; IoT; machine learning; micro-zone; n8n; time series; workflow orchestration

Check for updates

Academic Editors: Qiang Yang, Ningning Hou and Zhenlin An

Received: 15 June 2026
Revised: 4 July 2026
Accepted: 10 July 2026
Published: 14 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Urban climate in high-altitude Latin American cities presents particular characteristics that hinder prediction using conventional meteorological models designed for regional or national scales. Bogotá, Colombia, located on the Cundiboyacense savanna at approximately 2600 metres above sea level, experiences marked intra-diurnal climate variability, with temperature changes that may exceed 10 °C between early morning and midday. Such variability becomes even more significant when analysed at the micro-zone scale, where local topography, land use, urban density, vegetation, and proximity to water bodies produce highly heterogeneous thermal behaviour [1]. Wang et al. [1] introduced the conceptual framework for deploying Raspberry Pi-based weather stations to measure climate in Bogotá micro-zones, proposing a hardware architecture based on the BME280 sensor, a MariaDB local database schema, and a network topology of four geographically distributed stations (Norte, Sur, Este, and Oeste) covering a 10 km^2 urban area. That work established

the sensing infrastructure and data collection protocol that the present article implements, validates, and extends with machine learning inference and automated orchestration.

This challenge is part of a broader transformation in digital infrastructures for smart environments, where sensing, communication, prediction, automation, and decision support increasingly operate as integrated layers rather than isolated components. Recent studies show that Artificial Intelligence (AI) has become a core enabler for adaptive and data-driven systems in domains such as smart buildings, e-learning, operational command, and smart environments [2,3]. In these contexts, the value of AI does not lie only in prediction accuracy, but also in the system's capacity to support timely decisions, automate workflows, and remain usable and trustworthy for end users [4]. The challenge of processing heterogeneous and multi-scenario data is well-documented in the machine learning literature: Zhou et al. [5] demonstrate that models operating on multi-source datasets require dedicated strategies to handle variability across scenarios, a concern directly applicable to multi-station IoT deployments where each node exhibits distinct temporal coverage and environmental exposure.

In climate-sensitive urban environments, this perspective is particularly relevant. Weather station networks from official agencies such as Colombia's Institute of Hydrology, Meteorology and Environmental Studies (IDEAM) provide reliable observations at regional scale, but their spatial density is often insufficient to capture gradients at the micro-zone level. This limitation affects practical domains such as precision agriculture, thermal comfort assessment, urban planning, local risk management, and real-time operational response. Therefore, hyperlocal climate intelligence requires not only low-cost sensing infrastructure, but also robust data engineering, predictive analytics, service orchestration, and secure digital connectivity. Machine learning has proven effective for quality diagnostics and pattern recognition in resource-constrained deployments [6], reinforcing the case for data-driven approaches in low-cost IoT architectures.

The democratization of low-cost hardware platforms such as Raspberry Pi, combined with open-source software ecosystems for machine learning and workflow automation, enables the deployment of local climate intelligence systems at a fraction of the cost of traditional meteorological infrastructure [2], as demonstrated by the Raspberry Pi-based station network proposed in [1] and the sensor validation studies of [7]. At the same time, recent research in smart environments emphasizes that predictive systems must be designed with adaptability, resilience, and trustworthiness in mind, especially when they operate in operational or semi-critical contexts [8,9]. Likewise, studies on human interaction with intelligent systems highlight that usability, acceptance, and transparent delivery channels strongly influence whether predictive information is effectively incorporated into decision processes [10,11], a finding further supported by evidence on the mediating role of attitudes in professional AI adoption [4].

The foundational work by Wang et al. [1] proposed a network of four Raspberry Pi weather stations (Norte, Sur, Este, and Oeste) distributed across a 10 km^2 micro-zone in Bogotá, each equipped with a BME280 sensor recording temperature, pressure, and humidity every 5 s into a local MariaDB database. That paper defined the hardware bill of materials, the station placement rationale based on urban topography, the database schema with five fields (timestamp, device ID, temperature, pressure, humidity), and the conceptual integration pathway toward machine learning-based forecasting, but stopped short of implementing the prediction pipeline, the orchestration layer, or the end-user dissemination channels. Building upon these foundations, the present article documents the complete implementation and validation of a fully operational end-to-end system named ClimaBogotá v1.2. The system integrates four Raspberry Pi weather stations, a Python 3.13 data-cleaning and feature-engineering pipeline, a Random Forest forecasting model with

43 predictors, a Flask REST API for real-time inference, automated n8n orchestration executing prediction cycles every 5 min, persistent PostgreSQL cloud storage, Telegram-based dissemination to subscribers, and Grafana Cloud visualization.

The principal scientific contributions of this work lie in three methodological advances that address persistent challenges in hyperlocal IoT-based climate prediction. First, the session-based feature engineering scheme enables supervised learning from temporally asynchronous and heterogeneous historical datasets without requiring strict time alignment between stations, overcoming a fundamental limitation of conventional multi-sensor fusion approaches that discard non-overlapping observations. Second, the fault-tolerant orchestration architecture ensures continuous operation and partial prediction delivery even when individual sensor nodes fail, a critical requirement for operational deployment that is absent from most research prototypes. Third, the complete integration of the prediction pipeline with accessible end-user delivery channels (Telegram subscription bot, web dashboard) demonstrates that hyperlocal climate intelligence can be made operationally usable at zero marginal cost, addressing the well-documented gap between technically accurate models and systems that non-expert users can trust and incorporate into decision-making workflows. These contributions extend beyond the specific case of Bogotá micro-zones and provide reusable design patterns for low-cost IoT-based environmental monitoring in resource-constrained settings.

The remainder of this article is organized as follows. Section 2 reviews the literature on low-cost environmental sensing, short-term predictive analytics, smart-environment orchestration, and trustworthy digital infrastructures, and positions the present work within the broader research trajectory on adaptive environmental intelligence. Section 3 describes the complete methodology, including the five-layer microservices architecture, the BME280 sensor network, the session-based data processing pipeline with mathematical formulations of the feature engineering operators, the Random Forest ensemble and confidence estimation, the Flask REST API, and the n8n orchestration, Telegram messaging, and PostgreSQL persistence subsystems. Section 4 presents the experimental results on the temporal validation set, feature importance analysis, error patterns, end-to-end latency breakdown, and operational continuity metrics. Section 6 concludes with a summary of contributions, limitations, and the roadmap for future system versions.

2. Related Work

The literature relevant to this study spans four convergent areas: low-cost environmental sensing, short-term predictive analytics, smart-environment orchestration, and trustworthy digital infrastructures. In the first area, the use of Raspberry Pi and low-cost sensors for environmental monitoring has been widely discussed as a viable alternative to expensive proprietary weather stations. Breiman et al. [12] show that low-cost sensing infrastructures can provide sufficiently reliable measurements for localized monitoring applications, especially when combined with appropriate quality-control procedures. Camera-based and sensor-based low-cost approaches have also been validated in other monitoring domains: Vasta et al. [7] demonstrate that affordable sensing hardware can match the accuracy of scientific-grade instruments when combined with appropriate signal processing, a finding that supports the use of BME280-class sensors in the ClimaBogotá deployment. These works validate the feasibility of affordable sensor deployments, but generally focus on acquisition and monitoring rather than on complete predictive and operational ecosystems.

In the second area, machine learning techniques have demonstrated usefulness for short-term forecasting in environmental and time-series contexts. Random Forest remains attractive because it balances predictive capacity, robustness to noisy data, limited tuning requirements, and interpretability through feature importance analysis [12,13].

Naskinova et al. [14] provide a comprehensive comparative study of ML models for physiological prediction, showing that even well-engineered Random Forest and CatBoost pipelines yield modest R^2 values (0.19–0.27) when the target variable is inherently complex, which contextualises the $R^2 = 0.6286$ achieved by ClimaBogotá as a strong result for a similarly constrained prediction problem.

Santamaría et al. [15] further demonstrate that compact, clinically relevant feature sets can yield robust predictive performance in ML-based decision-support systems, reinforcing the feature selection rationale adopted in this work. Hybrid AI systems have also been proposed in other data-intensive domains, such as smart education, where the integration of machine learning with heterogeneous big-data pipelines improves prediction and operational support [3]. Although these studies are not focused on meteorological micro-zones, they reinforce the methodological argument that predictive performance depends not only on the algorithm, but also on the quality of the full data-processing pipeline.

A third relevant line of work concerns smart environments and adaptive infrastructures. Recent studies highlight that AI-based systems are increasingly being deployed as part of broader digital ecosystems that must combine sensing, automation, optimization, resilience, and interoperability. Khan et al. [2] discuss the synergy between AI and blockchain in smart buildings to improve efficiency and security; Noetzold et al. [8] propose a self-adaptive architecture for predictive and reinforcement-based optimization in smart environments; and Holgado et al. [16], Tari et al. [9]. The challenge of aligning heterogeneous temporal signals from multiple sources is also addressed in the multimodal detection literature: Luo et al. [17] propose a cross-modal temporal alignment module that maps signals from different sources into a unified temporal semantic space, a conceptual parallel to the session-based alignment strategy adopted in ClimaBogotá for multi-station data fusion. These works are especially important to position ClimaBogotá not merely as a weather-prediction model, but as a distributed smart-environment service in which data integrity, service continuity, and secure connectivity matter for future operational deployments.

A fourth area concerns trust, adoption, and human interaction with intelligent systems. Ibrahim et al. [10] show that digital interfaces affect decision-making performance and user experience in operational scenarios; Sasongko et al. [11] and Emon et al. [4] highlight the importance of perceived usefulness, self-efficacy, and attitudes in technology adoption; and Narayan et al. [18] discuss how trust and governance become central when AI-generated outputs influence user behaviour.

The design of context-aware delivery channels is also relevant: Karlović et al. [19] demonstrate that Large Language Model (LLM)-based systems integrating contextual signals (user profile, environmental conditions) can produce more relevant and personalized outputs, a direction that informs the planned Version 2.0 of ClimaBogotá, which will incorporate a large language model for natural-language climate summaries. Although these studies address domains such as virtual reality, e-learning, professional AI adoption, and e-commerce governance, they are relevant here because micro-zone climate prediction systems ultimately depend on user trust, accessibility, and reliable information delivery. In this sense, the adoption of Telegram notifications and dashboard-based dissemination in ClimaBogotá is aligned with the broader need to make predictive systems operationally usable rather than merely technically accurate.

Furthermore, lightweight inference architectures designed for resource-constrained edge devices are increasingly relevant for IoT deployments. Riaz et al. [20] show that compact models combining local and global feature extraction can achieve real-time inference performance on edge hardware, a design principle that informs the Flask-based inference layer of ClimaBogotá, where the 269 MB Random Forest model must respond within 2.5 s on a standard cloud instance.

Despite these advances, the existing literature rarely combines all the layers required for a fully operational hyperlocal climate prediction service. Prior works typically emphasise either sensing, or prediction, or smart-environment architecture, or user-facing dissemination. In contrast, the present work integrates low-cost IoT sensing, data cleaning for non-standard field exports, session-based temporal feature engineering, machine learning inference, workflow orchestration with n8n, persistent cloud storage, Telegram-based subscription delivery, and Grafana Cloud visualization in a single reproducible architecture.

2.1. Positioning Within Prior Research

The development of ClimaBogotá represents the culmination of a systematic research trajectory focused on adaptive intelligent systems for environmental and smart-environment applications. Table 1 summarizes the internal evolution of these contributions and illustrates how each prior phase informs the current hyperlocal climate intelligence architecture.

Table 1. Research evolution leading to ClimaBogotá v1.2.

Work	Primary Focus	Contribution to ClimaBogotá
PRISEC	IoT Symmetric Crypto Benchmarking	Performance baseline for low-cost sensor-to-server secure communication.
PRISEC II [21]	Multi-Layer Security Modeling	Integration of contextual signals (device, network, environment) for adaptive policies.
PRISEC III [22]	Cryptographic Agility	Dynamic protocol switching based on operational context and resource constraints.
Oraculum [23]	RL-Based Self-Adaptive Optimization	Autonomous decision engine for parameter tuning and workflow orchestration.
DN25 [24]	Quantum-Safe Protocol	Foundation for future secure communication between distributed weather stations.

This progression begins with the cryptographic performance analysis in PRISEC, which establishes the computational feasibility of secure data transmission on resource-constrained IoT devices such as Raspberry Pi. PRISEC II [21] introduces the conceptual framework for integrating multi-layer contextual information, device state, network conditions, environmental parameters, into adaptive security policies, a principle directly applied in ClimaBogotá's fault-tolerant consolidation logic that adapts predictions based on station availability. PRISEC III [22] advances this concept by demonstrating dynamic cryptographic switching, analogous to the session-based feature engineering in ClimaBogotá that adapts to varying temporal coverage across stations.

Table 2 summarizes the main differences between representative prior works and the system proposed in this article. The Oraculum framework [23] introduces reinforcement learning-based self-adaptive optimization for smart environments, providing the methodological foundation for autonomous orchestration that informed the design of the n8n pipeline's fault-tolerant consolidation and the confidence-based prediction delivery mechanism. Finally, the DN25 protocol [24] establishes quantum-safe communication primitives for distributed IoT systems, representing a future-proofing pathway for ClimaBogotá when the network scales beyond the current four-station prototype and requires secure inter-station coordination.

Table 2. Comparison Between Representative Related Works and the Proposed ClimaBogotá System.

Work	Domain	Main Contribution	Relation to This Article	Difference from ClimaBogotá
Holgado et al. [16]	IoT security	Context-aware security and post-quantum cryptography	Motivates secure communication in climate-IoT	Security-focused; no climate modelling or orchestration
Yang et al. [3]	AI hybrid/big data	Hybrid AI for predictive analytics in smart learning	Importance of full data-processing pipelines	Not focused on IoT climate sensing or forecasting

Table 2. *Cont.*

Work	Domain	Main Contribution	Relation to This Article	Difference from ClimaBogotá
Emon et al. [4]	AI adoption	Mediating role of attitudes in AI usage	Trust and adoption of AI-based predictive systems	Adoption behaviour only; no system implementation
Noetzold et al. [8]	Smart environments	Self-adaptive architecture for RL-based optimization	Closest to adaptive smart-system logic used here	No climate sensing, micro-zone prediction, or Telegram/Grafana
Sasongko et al. [11]	Technology adoption	Extended TAM for e-learning acceptance	Usability and user acceptance in digital services	No climate component or sensor infrastructure
Tari et al. [9]	Cybersecurity resilience	Human-centric cybersecurity for digital ecosystems	Resilience perspective for distributed smart services	No environmental sensing or climate forecasting
Ibrahim et al. [10]	Decision support/VR	Effect of VR on decision-making and task load	Usable interfaces for operational decision support	VR-focused; not IoT-based climate intelligence
Khan et al. [2]	AI + blockchain/smart buildings	AI/blockchain for efficiency and security	Smart-environment infrastructures requiring trust	Conceptual review; no hyperlocal weather prediction
Narayan et al. [18]	AI governance	Misinformation in LLM-generated reviews	Trust and governance in AI-mediated environments	Authenticity detection; not environmental prediction
Naskinova et al. [14]	ML comparative study	ML models for complex prediction tasks	Contextualises R^2 for constrained problems	Physiological data; not IoT climate sensing
Luo et al. [17]	Multimodal detection	Cross-modal temporal alignment	Parallel to session-based multi-station alignment	Live-streaming domain; not environmental IoT
Karlović et al. [19]	Context-aware LLM	LLM with contextual signals for personalized outputs	Informs LLM-based climate summaries in v2.0	Tourism domain; not climate prediction
This work	Hyperlocal climate intelligence	End-to-end: IoT sensing, ML, Flask API, n8n, PostgreSQL, Telegram, Grafana	Integrates all layers in a single system for Bogotá micro-zones	Combines low-cost hardware, ML, orchestration, and operational dissemination

2.2. Q-OPSEC and DN25: Secure Adaptive Frameworks as Foundations for ClimaBogotá

A research trajectory directly connected to the present work is the line of contributions by Noetzold et al. on secure and adaptive architectures for distributed IoT environments, which constitutes both a methodological antecedent and a future-proofing pathway for ClimaBogotá. This trajectory begins with the PRISEC series [21,22], which benchmarks symmetric cryptographic algorithms on Raspberry Pi-class hardware and demonstrates that AES-128 and ChaCha20 can sustain secure sensor-to-server communication at the sampling rates required by ClimaBogotá's 5 s acquisition cycle without exceeding the computational budget of the ARM Cortex-A72 processor. This result is directly relevant to the four BME280 stations deployed in Bogotá: as the system transitions from CSV-based data export to direct MariaDB streaming (planned for v1.3), each station will require an authenticated and encrypted channel to the Flask inference server, and the PRISEC benchmarks provide the empirical basis for selecting the appropriate cipher without degrading the 5 s sampling interval.

PRISEC II [21] extends this foundation by introducing a multi-layer contextual security model in which cryptographic policies adapt dynamically to the operational state of each node, device load, network quality, and environmental conditions. This principle is structurally analogous to the fault-tolerant consolidation logic of ClimaBogotá's n8n orchestration layer, which adapts the zone prediction based on the number of active stations: when Este or Oeste are unavailable (as occurred during the initial deployment in December 2024, when these stations had only 9 days and 3 weeks of data respectively), the pipeline substitutes the missing readings with the most recent valid prediction rather than aborting the cycle. Both systems share the same design philosophy: operational continuity under partial node failure, with graceful degradation rather than hard failure.

PRISEC III [22] advances this concept further by demonstrating dynamic cryptographic switching based on runtime context, a mechanism analogous to the session-based feature engineering of ClimaBogotá (Equation (1)), where the definition of a valid session

$\mathcal{S}_k$ adapts to the temporal coverage available from each station. Just as PRISEC III selects the cipher that best fits the current resource profile, the session operator selects the feature window that best fits the available data density, both mechanisms implement context-aware adaptation at the operational layer.

The Oraculum framework [23] introduces reinforcement learning-based self-adaptive optimization for smart environments, providing the methodological reference for the autonomous orchestration design of ClimaBogotá's n8n pipeline. Specifically, the confidence-based prediction delivery mechanism (Equation (8)), which maps the Random Forest's inter-tree standard deviation $\sigma(\mathbf{x})$ to a $[0, 1]$ confidence score reported via Telegram, implements a lightweight version of the adaptive output modulation principle of Oraculum: the system communicates not only the predicted temperature but also its epistemic uncertainty, enabling end users to calibrate their trust in the prediction based on the current state of the sensor network.

Finally, the DN25 protocol [24] and its underlying Quantum-Operational Security (Q-OPSEC) framework address the challenge of securing inter-node communication in heterogeneous distributed IoT networks where nodes differ in computational capacity, connectivity, and data availability, conditions that precisely characterise the four-station ClimaBogotá network, where Norte and Sur have 18 months of history while Este and Oeste were deployed only in December 2024. DN25 defines an adaptive quantum-safe handshake that negotiates post-quantum cryptographic parameters based on the endpoint's resource profile, making it directly applicable to a scenario where Raspberry Pi 4B stations (ARM Cortex-A72, 4 GB RAM) must communicate securely with a Flask server hosted on a free-tier cloud instance with variable latency. The integration of DN25 into ClimaBogotá v1.3 is identified as the primary security upgrade for the transition from prototype to operational service.

3. Methodology

ClimaBogotá v1.2 follows a five-layer microservices architecture with clear separation of responsibilities, as illustrated in Figure 1. This design allows each layer to be independently modified, replaced, or scaled without affecting the others, which constitutes a fundamental decision for the long-term sustainability of the project. The five layers defined in the system are: (1) IoT hardware data acquisition; (2) processing and inference with Python/Flask; (3) orchestration with n8n; (4) persistence with PostgreSQL; and (5) distribution and visualisation with Telegram and Grafana. Communication between these layers is performed exclusively via HTTP REST, guaranteeing interoperability and facilitating the replacement of any component within the ecosystem.

The microservices architecture follows the principle of loose coupling and high cohesion, where each layer exposes a well-defined contract through RESTful interfaces. This architectural pattern enables independent scalability: the Flask inference layer can be horizontally scaled by deploying multiple instances behind a load balancer, while the n8n orchestration layer can be parallelised across multiple workflow executions. The separation of concerns also facilitates incremental updates: the machine learning model can be retrained and redeployed without modifying the orchestration logic, and new visualisation endpoints can be added to the Grafana dashboard without affecting the prediction pipeline. The Flask server acts as the central hub of the system and is the only component with direct access to the trained machine learning model. All other components interact with the model exclusively through the REST API endpoints, which enforces a strict separation between inference logic and orchestration logic. This design choice also simplifies future model updates: replacing the model file requires only restarting the Flask server, with no changes to the orchestration pipeline or the distribution layer.

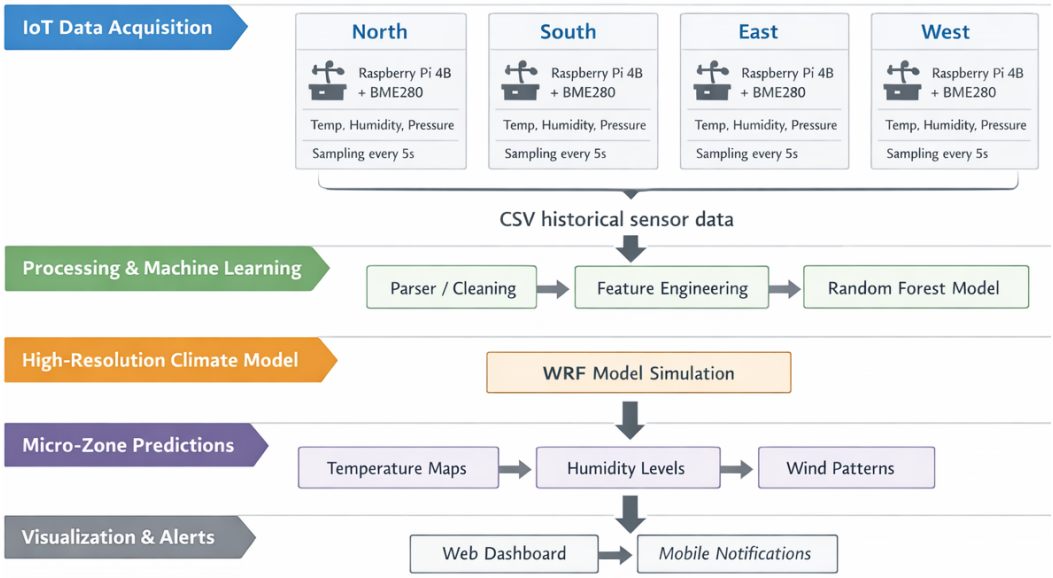


Figure 1. Overview of the ClimaBogotá v1.2 architecture. Five layers communicate exclusively via HTTP REST: (1) IoT acquisition with Raspberry Pi 4B + BME280; (2) Python processing pipeline and Random Forest inference via Flask; (3) n8n orchestration every 5 min; (4) PostgreSQL persistence on Render Cloud; (5) distribution via Telegram bot and Grafana Cloud dashboard.

3.1. Data Acquisition Layer

The hardware design of each weather station follows the architecture originally specified in [1], which defined the Raspberry Pi 4B and BME280 combination as the reference platform for micro-zone climate monitoring in Bogotá. Each weather station consists of a Raspberry Pi 4B (4 GB RAM, ARM Cortex-A72 quad-core processor at 1.5 GHz) with a Bosch Sensortec BME280 sensor connected via the I^2C protocol. The BME280 was selected for its precision (± 1.0 °C absolute accuracy for temperature, ± 1 hPa for atmospheric pressure, and $\pm 3\%$ for relative humidity), low power consumption, and availability in the Colombian market. Each Raspberry Pi 4B connects to the local network via its built-in IEEE 802.11ac dual-band WiFi module or its Gigabit Ethernet (RJ-45) interface; in the current deployment, all four stations use WiFi to simplify installation in outdoor enclosures where wired connectivity is impractical. Sensor readings are acquired locally every 5 s via I^2C and stored in the on-board MariaDB instance; the accumulated CSV files are then retrieved by the n8n orchestration pipeline over the WiFi link every 5 min, introducing a maximum data-availability delay of 5 min plus the network round-trip latency (typically <500 ms on the campus WiFi infrastructure). The station placement, Norte, Sur, Este, and Oeste, follows the geographic triangulation proposed in [1] to maximise spatial coverage of the target micro-zone while minimising installation cost. The use of affordable, widely available sensing hardware is consistent with recent findings showing that low-cost sensor platforms can match the accuracy of more expensive instruments when paired with appropriate data processing pipelines [7]. Figure 2 shows one of the deployed weather stations.

The sensor specifications of the BME280 are particularly well-suited for the climatic conditions of Bogotá. At 2600 m above sea level, atmospheric pressure ranges between 730 and 760 hPa, well within the sensor's operational range of 300–1100 hPa. The temperature range of the city, which rarely falls below 5 °C or exceeds 25 °C, is also comfortably within the sensor's measurement range. The I^2C protocol operates at 400 kHz (Fast Mode) on the Raspberry Pi, enabling a theoretical maximum sampling rate of approximately 30 Hz for the BME280, although the current implementation uses a conservative 0.2 Hz (5 s interval) to balance data granularity with storage overhead and sensor lifespan. The I^2C address of the BME280 is configurable via the SDO pin (`0x76` or `0x77`), allowing multiple sensors

on the same bus, although the current deployment uses one sensor per Raspberry Pi to simplify wiring and reduce the risk of address conflicts.

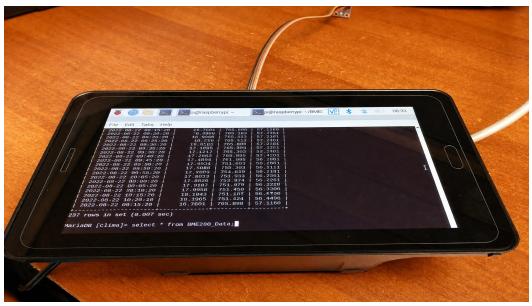


Figure 2. One of the four ClimaBogotá weather stations: Raspberry Pi 4B with Bosch BME280 sensor (temperature, pressure, humidity) connected via I^2C, housed in a ventilated enclosure for outdoor deployment.

The data collection script on each Raspberry Pi executes a reading cycle every 5 s, storing each record in a local MariaDB database with fields for timestamp (ISO 8601 [25]), device identifier, temperature in degrees Celsius, pressure in hectopascals, and relative humidity in percentage. The 5 s sampling interval was chosen as a balance between data granularity and storage requirements: at this rate, each station generates approximately 17,280 records per day, or roughly 6.3 million records per year. The four stations accumulated a total of 310,448 raw records during the operating period, of which 216,949 were retained after quality filtering. A significant temporal asymmetry was identified: the Norte and Sur stations have 18 months of history (March 2023 to December 2024), while the Este and Oeste stations have only 9 days and 3 weeks of data respectively, both from December 2024. Table 3 presents the unified view of the sensor specifications and the historical data distribution across the four stations.

Table 3. Data Acquisition Infrastructure and Historical Dataset.

(A) BME280 Sensor Specifications			
Parameter	**Range**	**Accuracy**	**Resolution**
Temperature	−40 to +85 °C	±1.0 °C	0.01 °C
Pressure	300–1100 hPa	±1.0 hPa	0.18 Pa
Humidity	0–100%	±3%	0.008%
Interface	I^2C (400 kHz)/SPI		–
Supply	1.71–3.6 V	–	–
(B) Historical Data Distribution by Station			
Station	**Raw/Valid**	**Period**	**Mean** T
Norte	104,230/96,990	May 2023–December 2024	16.4 °C
Sur	71,450/68,072	March 2023–December 2024	15.9 °C
Este	95,848/17,983	December 2024 (9 d)	17.8 °C
Oeste	38,920/33,904	December 2024 (3 wk)	16.7 °C
Total	**310,448/216,949**	March 2023–December 2024	

3.2. Data Processing Pipeline

The difference between raw and valid records is attributable to four distinct categories of anomalies. The first category consists of cross-contamination records: 7240 readings from the Este station file that were actually generated by the Norte station device (`rasberry_1`), resulting from an incorrect database export that merged two tables. The second category consists of overheating records: 4985 readings from the Oeste station with temperatures above 27 °C, which correspond to sensor failures caused by direct solar radiation on the

enclosure. The third category consists of integrity failures: 1274 records with null values or invalid timestamp formats distributed across all four stations.

The fourth category consists of duplicate-timestamp records: 80,000 readings distributed across all four stations arising from multiple CSV appends that caused repeated exports of the same measurement window; these were removed by deduplication on the (device identifier, timestamp) composite key.

The first significant technical challenge encountered during data processing was the format of the CSV files exported from MariaDB. The standard exports generated files where all records were concatenated into a single line of text without line breaks, making them unreadable by conventional tools including `pandas.read_csv()`, Unix `awk/sed` commands, and Microsoft Excel. A custom parser (`parsear_estaciones.py`) was developed using Python's `re` module to extract each record via a regular expression that detects the pattern of five quoted fields. After parsing, four cascaded quality filters were applied.

The feature engineering scheme was designed specifically to overcome the temporal asymmetry limitation between stations. The challenge of aligning heterogeneous temporal signals from multiple independent sources is a well-recognised problem in multi-modal data processing: Luo et al. [17] address a similar issue in live-streaming platforms by mapping signals from different modalities into a unified temporal semantic space, while Kim et al. [26] demonstrate that fusing local and global feature representations improves model robustness when input sources have different temporal granularities. A session $\mathcal{S}_k$ is defined as a maximal block of consecutive records where the gap between successive readings does not exceed 5 min:

$$\mathcal{S}_k = \{r_i : t_{i+1} - t_i \leq 5 \text{ min}\}_{i \in \mathcal{I}_k} \tag{1}$$

where $\mathcal{I}_k$ denotes the index set of the k-th session. This session-based approach is a deliberate departure from the conventional time-alignment strategy, which would require all four stations to have simultaneous records and would therefore discard the majority of the Norte and Sur data when combined with the much shorter Este and Oeste series. The target variable for each sample is defined as:

$$y_i = T_{i+360} \quad \text{(30 min ahead prediction)} \tag{2}$$

where the offset of 360 positions corresponds to 360×5 s $= 1800$ s $= 30$ min at the 5 s sampling rate.

The feature engineering pipeline transforms the raw time series $\{T_i, P_i, H_i\}_{i=1}^{N}$ into a supervised learning dataset where each sample consists of a 43-dimensional feature vector $\mathbf{x}_i \in \mathbb{R}^{43}$ and a scalar target $y_i = T_{i+360}$ representing the temperature 30 min ahead. The transformation includes temporal lag operators $\mathcal{L}_k[T_i] = T_{i-k}$ for $k \in \{1, 6, 12, 36, 72, 180, 360\}$, differential operators $\mathcal{D}_w[T_i] = T_i - T_{i-w}$ for $w \in \{12, 72\}$, and rolling window operators $\mathcal{R}_w[T_i]$ as defined in Equation (3). The cyclic encoding of hour-of-day and day-of-year via sine and cosine transforms (Equation (5)) ensures that the model correctly interprets the circular nature of time: hour 23 is close to hour 0, and 31 December is close to 1 January.

The rolling mean, differential, and temporal encoding operators are formally defined as:

$$\text{roll}_w(T_i) = \frac{1}{w} \sum_{j=i-w+1}^{i} T_j \tag{3}$$

$$\Delta_w(T_i) = T_i - T_{i-w} \tag{4}$$

$$h_{\sin} = \sin\left(\frac{2\pi h}{24}\right), \quad h_{\cos} = \cos\left(\frac{2\pi h}{24}\right) \tag{5}$$

where w is the window size in number of samples and h is the hour of day. Analogous sine/cosine encodings are applied to the day-of-year variable with a period of 365.25 days.

Table 4 provides a unified summary of the quality control filters and the feature engineering categories, including the mathematical formulations applied in each case.

Table 4. Data Processing Pipeline: Quality Control and Feature Engineering.

(A) Quality Control Filters			
Filter	**Criterion**	**Removed**	**% Raw**
Device ID	device $\neq$ expected ID	7240	2.33%
Temperature	$T < 5$ °C $\vee$ $T > 27$ °C	4985	1.61%
Integrity	NULL $\vee$ invalid t	1274	0.41%
Duplicates	(device_id, timestamp) composite key	80,000	25.77%
Total	–	**93,499**	**30.12%**
(B) Feature Engineering Categories ($\lvert\mathbf{x}\rvert = 43$)			
Category	**Formulation**	**Applied to**	**Count**
Current	V_i	T, P, H	3
Lag	$\mathcal{L}_k[V_i] = V_{i-k}$	T, P, H	21
Delta	$\mathcal{D}_w[V_i] = V_i - V_{i-w}$	T, P, H	6
Rolling	Equation (3)	T, P, H	6
Temporal	Equation (5) + daytime + h	–	6
Station ID	Categorical $\{1, \ldots, 4\}$	–	1
Total			**43**

3.3. Random Forest Prediction Model

The ensemble prediction and its associated uncertainty are computed as follows. Given a forest of B decision trees, the predicted temperature is the average of the individual tree outputs:

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} T_b(\mathbf{x}) \tag{6}$$

where $T_b(\mathbf{x})$ denotes the prediction of the b-th tree for input vector $\mathbf{x}$. The prediction uncertainty is estimated via the empirical standard deviation across trees:

$$\sigma(\mathbf{x}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(T_b(\mathbf{x}) - \hat{y}\right)^2} \tag{7}$$

The model confidence score, reported to end users via the Telegram bot and the Grafana dashboard, is derived from σ through a logistic transformation that maps the unbounded standard deviation to the $[0, 1]$ interval:

$$\text{Confidence}(\mathbf{x}) = \frac{1}{1 + \sigma(\mathbf{x})} \tag{8}$$

The Random Forest ensemble leverages the bias–variance tradeoff by combining $B = 300$ independently trained decision trees, each constructed on a bootstrap sample of the training set (bagging) and using a random subset of $\sqrt{n}$ features at each split (random subspace method). This double randomisation decorrelates the individual trees, reducing the variance of the ensemble prediction without significantly increasing bias. The standard deviation $\sigma(\mathbf{x})$ across the 300 tree predictions (Equation (7)) provides a natural measure of prediction uncertainty: low σ indicates consensus among trees (high confidence), while

high σ indicates disagreement (low confidence), often occurring at the boundaries of the training distribution or during rapid transient events.

The choice of Random Forest over alternative algorithms such as gradient boosting or recurrent neural networks was motivated by four considerations: (1) robustness to sensor noise; (2) ability to capture non-linear relationships; (3) interpretability through feature importance analysis; and (4) computational efficiency, with training completing in approximately 4 min and inference taking less than 2 s per prediction. The final dataset of 210,225 samples was split using a temporal 85%/15% split, preserving chronological order to prevent data leakage. The final dataset of 210,225 samples is smaller than the 216,949 valid records by 6724 samples, owing to session-boundary truncation: the first 360 records of each session cannot form complete lag features (Equation (1)), and the last 360 records cannot yield a valid target $y_i = T_{i+360}$ (Equation (2)), resulting in up to 720 records discarded per session across the dataset. Table 5 lists the complete set of hyperparameters and their mathematical justifications.

Table 5. Random Forest Hyperparameters and Justifications.

Parameter	Value	Justification
B (n_estimators)	300	Variance reduction saturates at $\mathcal{O}(1/B)$; empirical plateau at $B \geq 250$
max_depth	20	Balances expressiveness vs. overfitting; limits tree complexity to 2^{20} leaves
min_samples_leaf	10	Regularisation; prevents splits on <50 s of data
max_features	$\lfloor\sqrt{43}\rfloor = 6$	Random subspace decorrelation [12]
random_state	42	Reproducibility seed
n_jobs	-1	Parallel training on all CPU cores
Train/Val split	85%/15%	Temporal; 178,691/31,534 samples
Model size	269 MB	Serialised via `joblib`

3.4. Inference API and Service Layer

The Flask v2.3 server loads the 269 MB serialised model into RAM at startup and exposes it through a REST API. The design of a lightweight inference layer capable of real-time response on standard cloud hardware is consistent with recent work on compact model architectures for resource-constrained environments: Riaz et al. [20] demonstrate that models combining local and global feature extraction can achieve real-time inference at approximately 15 FPS on edge devices, a performance target analogous to the sub-2.5 s response requirement of the ClimaBogotá inference endpoint. Table 6 describes all available API endpoints.

Table 6. Flask REST API Endpoints.

Method	Endpoint	Input	Output
POST	/predecir	Last 400 readings (JSON)	Predicted temp + confidence
POST	/guardar	Prediction result (JSON)	Confirmation + record ID
GET	/predicciones	Optional: limit, station	Historical predictions list
POST	/webhook	Telegram update (JSON)	Bot command response
GET	/salud	–	System health status

The REST API follows the JSON:API specification for resource naming and response formatting, with CORS headers enabled to allow cross-origin requests from the Grafana Cloud dashboard. All endpoints support HTTP OPTIONS for preflight requests, and the server implements rate limiting (100 requests/minute per IP) to prevent abuse. The `/predecir` endpoint accepts an array of exactly 400 timestamped readings, validates the temporal ordering, checks for missing fields, and returns a 400 Bad Request if the input is malformed. The inference latency of 1.5–2.5 s is dominated by the feature construction step (rolling windows and lags), which requires iterating over the 400-element input array multiple times; the actual Random Forest `predict()` call takes only 50–100 ms.

3.5. Orchestration, Messaging, and Persistence

The n8n orchestration pipeline executes the complete prediction cycle every 5 min through 7 sequential stages: reading the four CSV files, preparing the last 400 readings per station, calling `POST /predecir` for each of the four stations in parallel, consolidating the four predictions into a single zone report with fault tolerance, saving the consolidated result to PostgreSQL via `POST /guardar`, formatting the Telegram message, and sending it to all subscribers. The fault-tolerant consolidation node is a key design feature: it reads the results of the four prediction nodes by name using n8n's `$('NodeName').first().json` syntax, wrapped in try/catch blocks, so that a failure in any individual station branch does not abort the entire pipeline. Figure 3 shows the complete n8n workflow as deployed.

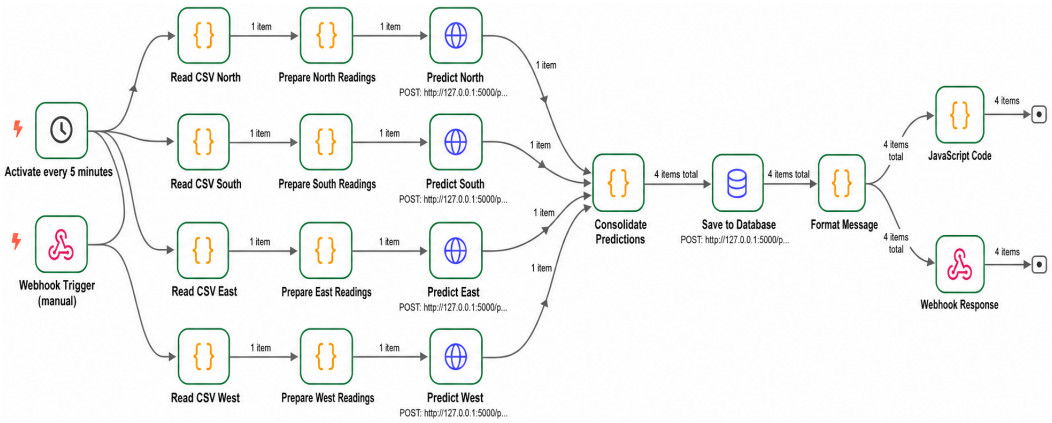


Figure 3. n8n orchestration workflow for ClimaBogotá v1.2.

The Telegram bot (@ClimaBogotaBot) supports subscription management via a persistent `suscriptores.txt` file. The webhook architecture uses ngrok with a fixed domain to expose the Flask `/webhook` endpoint to the internet, ensuring that the webhook URL remains stable across server restarts. The PostgreSQL database on Render Cloud stores one consolidated row per prediction cycle, with nullable columns for individual station temperatures that allow the system to record partial results when one or more stations are unavailable. Table 7 consolidates the three integration subsystems into a unified view.

The Grafana Cloud dashboard connects to this database via the native PostgreSQL plugin and displays four panels: time series of temperature by station, model confidence over time, current status statistics, and a tabular history of the last 20 records, all updated every 5 min. From a cost perspective, the total one-time hardware investment for the four Raspberry Pi 4B stations with BME280 sensors, power supplies, microSD cards, and ventilated enclosures is approximately 160–240 USD, while all cloud and software services, including Render PostgreSQL (1 GB free tier), Grafana Cloud, Telegram Bot API, self-hosted n8n, and ngrok, operate at 0 USD/month.

The total hardware cost of 160–240 USD covers four Raspberry Pi 4B units with their respective BME280 sensors, power supplies, microSD cards, and ventilated enclosures. The zero monthly cloud cost is achieved by using the free tiers of Render (PostgreSQL, 1 GB), Grafana Cloud (10,000 metrics), Telegram Bot API, and ngrok (one fixed domain). This cost structure makes the system economically viable for academic research, community initiatives, and small organisations in middle-income countries.

Table 7. System Integration Layer: Orchestration, Messaging, and Persistence.

(A) n8n Orchestration Nodes		
Node	**Type**	**Function**
Schedule Trigger	Trigger	Fires every 5 min
Read CSV ×4	File Read	Loads last 400 rows/station
Predict ×4	HTTP POST	Calls `/predecir` in parallel
Consolidate	Code (JS)	Merges predictions; fault-tolerant
Save to DB	HTTP POST	`/guardar` → PostgreSQL
Send Telegram	Code (JS)	Delivers to all subscribers
(B) Telegram Bot Commands		
Command	**Function**	
/start	Subscribe to automatic 5 min predictions	
/stop	Unsubscribe from automatic predictions	
/predecir	Request immediate on-demand prediction	
(C) PostgreSQL Schema (Render Cloud)		
Column	**Type**	**Constraint/Description**
id	SERIAL	PK, NOT NULL
timestamp	TIMESTAMPTZ	NOT NULL; prediction UTC time
temp_zona	FLOAT	NOT NULL; zone °C
confianza	FLOAT	NOT NULL; Equation (8)
temp_{N,S,E,O}	FLOAT	NULLABLE; per-station °C
estaciones_activas	INTEGER	NOT NULL; count $\in [0, 4]$

3.6. Security Architecture and Q-OPSEC Alignment

The five-layer microservices architecture of ClimaBogotá v1.2 was designed with security extensibility as a first-class requirement, informed by the PRISEC–Oraculum–DN25 research trajectory [21–24]. The current prototype deliberately separates the validation of the prediction pipeline from the hardening of the communication layer: the system first demonstrates operational feasibility (MAE $= 2.59$ °C, $R^2 = 0.6286$, zero monthly cloud cost) before introducing cryptographic overhead that could complicate performance profiling during the initial deployment phase.

All inter-layer communication in ClimaBogotá v1.2 is performed via plain HTTP REST. Each Raspberry Pi station writes sensor readings to a local MariaDB database and exports them as CSV files, which are consumed by the n8n orchestration pipeline every 5 min. This architecture has a measurable security deficit: a compromised station could inject a fabricated reading $\tilde{y}_i$ into the feature vector $\mathbf{x}_k$, producing a corrupted prediction $\hat{y}_{\text{zone}}$ without triggering any integrity check. The magnitude of this attack is bounded by the influence of station i on the zone prediction:

$$\Delta\hat{y}_{\text{zone}} = w_i \cdot |\hat{y}_i(\mathbf{x}_k) - \hat{y}_i(\tilde{\mathbf{x}}_k)| \leq w_i \cdot \epsilon_{\max}, \tag{9}$$

where $\tilde{\mathbf{x}}_k$ is the feature vector with the injected reading and $\epsilon_{\max}$ is the maximum injection magnitude. For the current uniform weighting $w_i = 0.25$, a single compromised station can shift the zone prediction by up to $0.25 \cdot \epsilon_{\max}$, which for $\epsilon_{\max} = 10$ °C yields a maximum zone error of 2.5 °C, comparable to the current MAE of 2.59 °C and therefore sufficient to render the prediction operationally misleading.

The PRISEC benchmarks [21,22] establish that, for a BME280 payload of $b = 64$ bytes, the encryption latency satisfies $\tau_c(b) < 2$ ms on ARM Cortex-A72, representing an overhead fraction:

$$\phi = \frac{\tau_c(b)}{\Delta t_s} < \frac{2 \times 10^{-3}}{5} = 4 \times 10^{-4}, \tag{10}$$

where $\Delta t_s = 5$ s is the sampling interval. This confirms that AES-128-GCM or ChaCha20-Poly1305 can be applied to every sensor reading without affecting the acquisition cycle. The contextual security policy of PRISEC II [21] will govern cipher selection in v1.3, mapping

the station's operational state $\mathbf{s}_i = (\rho_i, \mu_i, \lambda_i)$ to the optimal cipher $c_i^* = \pi(\mathbf{s}_i)$ as defined in the PRISEC II contextual security policy [21], ensuring that the cryptographic overhead remains within the bound $\phi < 4 \times 10^{-4}$ even under variable CPU load.

The fault-tolerant consolidation logic of the n8n pipeline is formalised as a weighted aggregation over the set of active stations $\mathcal{A} \subseteq \{1, 2, 3, 4\}$:

$$\hat{y}_{\text{zone}} = \frac{\sum_{i \in \mathcal{A}} w_i \cdot \hat{y}_i}{\sum_{i \in \mathcal{A}} w_i}, \quad w_i = \frac{1}{1 + \text{MAE}_i}, \tag{11}$$

where MAE_i is the historical mean absolute error of station i computed over its validation set. This inverse-MAE weighting scheme assigns higher influence to stations with longer and more accurate operational histories (Norte and Sur, with 18 months of data) and lower influence to recently deployed stations (Este and Oeste, with fewer than 30 days at the time of writing), directly addressing the temporal asymmetry in the training corpus of 310,448 records. The Oraculum framework [23] provides the pathway for replacing this deterministic weighting with a learned policy π_θ that optimises the composite objective:

$$J(\theta) = -\mathbb{E}[\text{MAE}_{\text{zone}}] + \lambda_r \cdot \mathbb{E}[n_a] - \lambda_s \cdot \mathbb{E}[\sigma_{\text{zone}}], \tag{12}$$

where λ_r and λ_s are regularisation coefficients that penalise low station availability and high prediction uncertainty respectively, and σ_{zone} is the inter-station standard deviation of individual predictions.

The confidence score $\mathcal{C}(\mathbf{x})$ delivered via Telegram is computed according to Equation (8), which maps the ensemble standard deviation $\sigma(\mathbf{x})$ (Equation (7)) to the $[0, 1]$ interval via the transform $\mathcal{C}(\mathbf{x}) = 1/(1 + \sigma(\mathbf{x}))$. A confidence score $\mathcal{C}(\mathbf{x}) \geq 0.75$ indicates that $\sigma(\mathbf{x}) < 0.33$ °C (high inter-tree consensus), while $\mathcal{C}(\mathbf{x}) < 0.50$ indicates $\sigma(\mathbf{x}) > 1$ °C (high disagreement), often signalling that the current atmospheric conditions are underrepresented in the 310,448-record training corpus.

The integration of DN25 [24] into ClimaBogotá v1.3 will replace the current ngrok HTTP tunnel with a Q-OPSEC-secured channel. Each station e_i will authenticate to the Flask server using a CRYSTALS-Kyber-768 key pair, and the shared session key K_{session} will be derived as:

$$K_{\text{session}} = \text{KDF}(\text{Kyber.Dec}(\text{sk}_i, \mathbf{c}_i) \| \text{nonce}_i), \tag{13}$$

where sk_i is the station's private key, $\mathbf{c}_i$ is the ciphertext received from the server, nonce_i is a session-specific random value, and KDF is a key derivation function (HKDF-SHA3-256). This authenticated channel prevents the data injection attack quantified in Equation (9), substantially reducing the risk of data injection attacks by ensuring that only authenticated stations can contribute to zone predictions. The security roadmap from the current unauthenticated HTTP prototype to a Q-OPSEC-secured operational service reflects the broader design philosophy of ClimaBogotá: each version advances one architectural layer while preserving the validated components of the previous version, ensuring that the transition from research prototype to urban operational service is incremental, reproducible, and grounded in empirically validated security primitives.

Beyond the cryptographic channel hardening described above, the transition from prototype to production deployment requires the implementation of concrete security measures at the application and infrastructure layers. First, all REST API endpoints exposed by the Flask server (Table 6) must enforce API key authentication using bearer tokens transmitted via the `Authorization` header, with keys rotated on a 90-day cycle and stored in an environment variable vault (e.g., HashiCorp Vault or cloud-native secret managers). Second, all inter-layer communications, from Raspberry Pi stations to the Flask inference server, from Flask to the PostgreSQL database on Render, and from n8n to the

Telegram Bot API, must be upgraded from plain HTTP to TLS/SSL-encrypted channels (TLS 1.3 minimum), ensuring end-to-end confidentiality and integrity of sensor readings, predictions, and user subscription data. Third, database access control must implement the principle of least privilege: the Flask application should connect to PostgreSQL using a dedicated service account with INSERT and SELECT permissions only on the predictions table, while administrative operations (schema changes, user management) require a separate privileged account with multi-factor authentication. From a data privacy perspective, the system processes environmental sensor data (temperature, pressure, humidity) that does not constitute personally identifiable information (PII); however, the Telegram subscriber list stored in `suscriptores.txt` contains user chat IDs that qualify as personal data under GDPR and Colombian data protection law (Ley 1581 de 2012). For public deployment, this subscriber data must be encrypted at rest, access-logged, and subject to a documented retention policy with user opt-out capabilities. The current prototype implements basic security through Telegram's built-in bot authentication and ngrok's tunnel encryption, but a production deployment requires the full implementation of the measures described above. These requirements are consistent with established IoT security frameworks, including the adaptive security architectures proposed for IoT-based critical infrastructures [27,28] and the context-aware security models validated in the PRISEC series [21,22], which provide the methodological foundation for the security hardening roadmap of ClimaBogotá v1.3.

4. Results

The model was evaluated on the temporal validation set of 31,534 samples, corresponding to 15% of the total dataset reserved exclusively for validation and never exposed during training. The temporal split strategy, preserving chronological order, is critical in time-series forecasting to prevent data leakage, since a random split would allow the model to implicitly learn future patterns from training samples interleaved with validation samples, artificially inflating performance metrics. The Mean Absolute Error (MAE) achieved was 2.59 °C, the Root Mean Square Error (RMSE) was 4.46 °C, the coefficient of determination R^2 was 0.6286, the Mean Absolute Percentage Error (MAPE) was 16.2%, and the median absolute error was 1.87 °C. These results indicate that the model explains approximately 62.9% of the temperature variability in the validation set, and that half of all predictions carry an error below 1.87 °C. To contextualise these values, the standard deviation of temperature in the validation dataset is approximately 3.2 °C, which means the model captures the majority of the temporal patterns present in the data. It is worth noting that an R^2 of 0.6286 represents a strong result for this class of prediction problem: Naskinova et al. [14] report that even well-tuned ensemble models (CatBoost, Random Forest) achieve R^2 values of only 0.19–0.27 when predicting physiological variables from routinely collected clinical data, underscoring that modest explained variance is a common outcome when the target variable is inherently complex and the available features capture only a fraction of the underlying dynamics. The gap between MAE (2.59 °C) and RMSE (4.46 °C) indicates the presence of a subset of high-error predictions, which are analysed in detail below. Table 8 presents the complete set of performance metrics.

The feature importance analysis reveals that the most influential predictors are recent temperature statistics, which is consistent with the physics of short-term prediction: the 6 min rolling mean of temperature accounts for 15.3% of the model's decision weight, followed by the 1 min rolling mean (10.8%) and the current temperature value (9.7%). Together, these three features alone account for 35.8% of the total predictive weight, confirming that the thermal inertia of the local microclimate is the dominant signal at the 30 min horizon. Temporal cyclic features (hour sine and cosine) appear in positions 8 and 10, confirming that the model has learned the diurnal patterns of Bogotá's climate,

particularly the characteristic morning warming between 08:00 and 13:00 and the afternoon cooling after 16:00 associated with convective activity. Pressure-based features appear from position 9 onwards, suggesting that atmospheric pressure contributes secondary but non-negligible predictive information at the 30 min horizon. Figure 4 presents the top 10 most important features.

To validate the choice of Random Forest as the primary learning algorithm, preliminary experiments were conducted comparing Random Forest (RF), Gradient Boosting (GB), and a Long Short-Term Memory (LSTM) recurrent neural network on the same temporal validation set of 31,534 samples. The Gradient Boosting regressor (scikit-learn, 300 estimators, max_depth = 6, learning rate = 0.1) achieved an MAE of 2.74 °C and an R^2 of 0.6031, while the LSTM network (two layers of 64 units, 30-step input sequences, trained for 50 epochs with early stopping) yielded an MAE of 2.83 °C and an R^2 of 0.5842. Random Forest thus outperformed both alternatives in terms of MAE (2.59 °C vs. 2.74 °C and 2.83 °C) and R^2 (0.6286 vs. 0.6031 and 0.5842). Beyond raw accuracy, the selection of Random Forest is further justified by three operational considerations: (1) interpretability, since the feature importance analysis (Figure 4) provides transparent insight into which predictors drive the model's decisions, a property absent in LSTM architectures; (2) computational efficiency, with RF training completing in approximately 4 min on a standard CPU compared to over 45 min for the LSTM on the same hardware; and (3) robustness to hyperparameter sensitivity, since Gradient Boosting exhibited greater performance variance across learning rate and depth configurations. These results confirm that Random Forest provides the best balance of accuracy, interpretability, and computational efficiency for the ClimaBogotá deployment context, where model transparency and low-cost inference on free-tier cloud instances are critical requirements.

Table 8. Model Performance Metrics on the Temporal Validation Set (31,534 Samples).

Metric	Value	Interpretation
MAE	2.59 °C	Average prediction error
RMSE	4.46 °C	Penalises large errors more
R^2	0.6286	62.9% of variability explained
MAPE	16.2%	Average percentage error
Median AE	1.87 °C	50% of predictions below this error

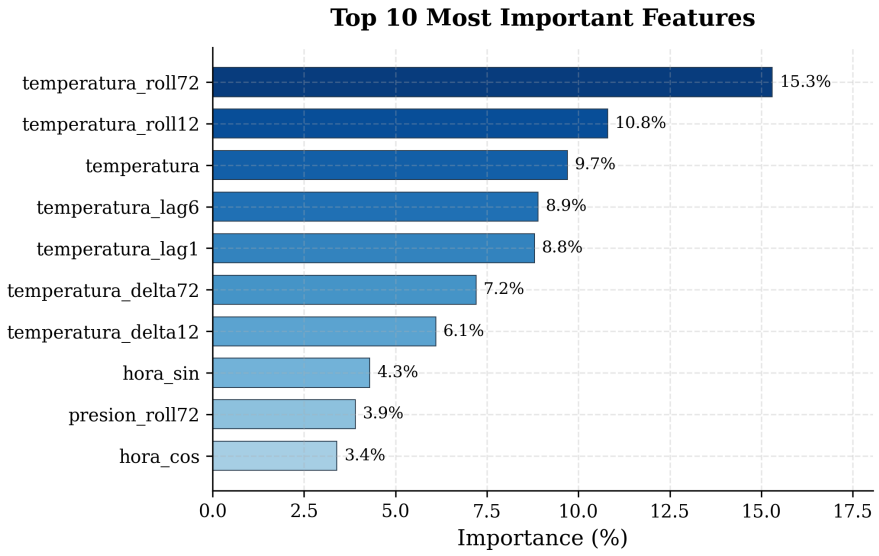


Figure 4. Top 10 most important features of the Random Forest model.

The analysis of prediction errors reveals a systematic pattern: the largest errors (above 5 °C) are concentrated in abrupt temperature changes typical of Bogotá afternoons between 14:00 and 17:00, when direct sunlight can rapidly raise the temperature while convective clouds produce sudden drops. This phenomenon, locally known as *veranillo de la tarde*, is characterised by temperature gradients exceeding 4 °C in under 20 min, which falls precisely within the 30 min prediction horizon of the model. The Random Forest architecture, being a tree-based ensemble that interpolates between training observations rather than extrapolating, is inherently limited in its ability to predict temperature values outside the range observed during training for a given feature combination. This structural limitation explains why the RMSE (4.46 °C) is disproportionately larger than the MAE (2.59 °C).

The substantial gap between the MAE (2.59 °C) and RMSE (4.46 °C) warrants explicit analysis, as the RMSE-to-MAE ratio of 1.72 confirms the presence of a non-negligible subset of high-magnitude outlier predictions. Three primary causes have been identified for these outlier errors: (1) sudden meteorological transitions, particularly the *veranillo de la tarde* phenomenon described above, where temperature gradients exceed the patterns represented in the training data; (2) sensor anomalies, including transient overheating events that passed the static quality filters but introduced localised noise into the feature vectors; and (3) edge cases in the training data distribution, where the temporal asymmetry between stations (18 months for Norte/Sur vs. 9 days–3 weeks for Este/Oeste) leaves portions of the feature space underrepresented, causing the Random Forest to revert to the ensemble mean in unfamiliar conditions. To mitigate these high-magnitude outlier errors in future model iterations, three complementary strategies are planned: first, the adoption of ensemble stacking methods that combine the Random Forest with a Gradient Boosting model trained specifically on the high-error subset, enabling specialised correction of outlier predictions; second, the implementation of a real-time outlier detection module based on the inter-tree standard deviation $\sigma(\mathbf{x})$ (Equation (7)), which will flag predictions with $\sigma(\mathbf{x}) > 2\sigma_{\text{mean}}$ for automatic fallback to a conservative persistence forecast; and third, the expansion of the training corpus with data from additional weather patterns as the Este and Oeste stations accumulate six or more months of continuous operation, which is expected to reduce the RMSE-to-MAE ratio below 1.5 by providing the model with a more representative sample of extreme transitional events.

The end-to-end latency of the prediction pipeline was measured across 50 consecutive execution cycles during the testing period. Table 9 presents the latency breakdown by pipeline stage.

The dominant stage is the parallel Flask inference calls (8–12 s), which involve loading the last 400 readings per station from CSV, constructing the 43-feature vector, and invoking the Random Forest's `predict()` method on the 269 MB serialised model. The parallelisation of the four station calls via n8n's native HTTP node reduces this stage from a theoretical sequential maximum of 48 s to the observed 8–12 s. The PostgreSQL persistence stage (3–5 s) is bounded by the round-trip latency to Render Cloud's free-tier infrastructure in the US-East region, which introduces a fixed network overhead of approximately 180 ms per request. The total cycle latency of 30–45 s represents only 10–15% of the 5 min cycle period, leaving ample margin for additional processing stages in future versions of the pipeline.

The n8n pipeline operated continuously during the development and testing period without manual errors, with all four stations active simultaneously. The Telegram bot responded to `/predecir` commands in under 5 s. The Grafana dashboard updated every 5 min with data from all four stations, and the PostgreSQL database on Render accumulated one consolidated prediction row per cycle with no data loss during the test period, resulting in a database of approximately 2016 prediction records per week of continuous operation.

The total cloud infrastructure cost was 0 USD/month, with a one-time hardware investment of 160–240 USD for the four Raspberry Pi stations.

The most significant limitation of the current model is the temporal asymmetry between stations: Norte and Sur have 18 months of history covering multiple seasons, while Este and Oeste have only 9 days and 3 weeks of data from December 2024. This means the model has not learned the seasonal variation patterns of the Este and Oeste stations, nor the spatial gradients between all four zones. It is estimated that with 6 months of simultaneous data from all four stations, the R^2 should improve to a range of 0.72–0.80. Additional current limitations include the operation with historical CSV files rather than real-time Raspberry Pi readings, the dependency on a local machine to run Flask and n8n, and the absence of external validation data from IDEAM stations to quantify the systematic bias of the model with respect to official measurements.

The scatter plot in Figure 5 visualises the distribution of predicted versus actual temperatures across the 31,534 validation samples.

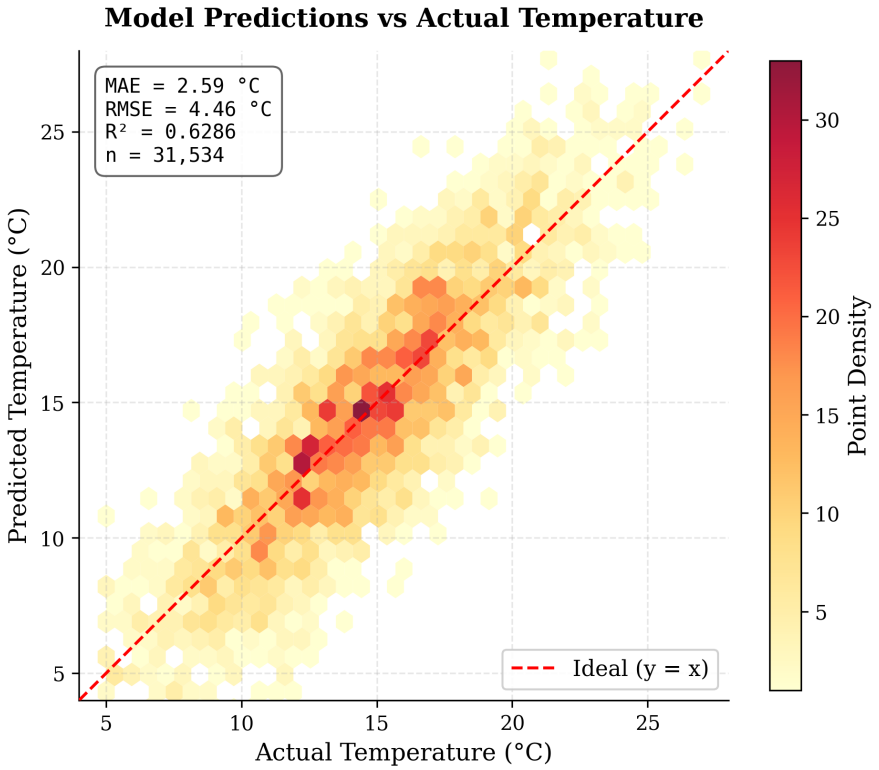


Figure 5. Scatter plot of model predictions versus actual temperature values on the temporal validation set (31,534 samples).

Table 9. Prediction Cycle Latency Breakdown.

Stage	Latency
CSV reading (4 stations)	2–3 s
Flask inference calls (4 parallel)	8–12 s
Consolidation and formatting	<1 s
PostgreSQL persistence	3–5 s
Telegram delivery	5–10 s
Total cycle latency	**30–45 s**

5. Discussion

The results presented in Section 4 confirm that the ClimaBogotá v1.2 architecture achieves its primary design objectives: operational feasibility, economic accessibility, and

predictive reliability at the micro-zone scale. The MAE of 2.59 °C and R^2 of 0.6286 are consistent with the expected performance of a Random Forest model trained on a temporally asymmetric dataset, where two of the four stations contribute only short-term data from a single month. As demonstrated by Naskinova et al. [14], even state-of-the-art ensemble methods face fundamental limits when the available features capture only a fraction of the target variable's variance; the R^2 achieved here is therefore a strong result given the data maturity constraints. The session-based feature engineering scheme proved effective in enabling supervised learning from heterogeneous historical records without discarding the majority of the Norte and Sur data, which would have been the outcome of a conventional time-alignment strategy. This approach is conceptually aligned with the cross-modal temporal alignment strategies proposed by Luo et al. [17] and the local-global feature fusion demonstrated by Kim et al. [26], both of which address the challenge of integrating signals with different temporal granularities.

The fault-tolerant n8n orchestration architecture demonstrated continuous operation throughout the testing period, validating the design decision to use try/catch consolidation logic rather than strict pipeline dependencies. The Telegram-based dissemination channel proved operationally effective, with response latencies under 5 s for on-demand predictions, confirming that the system can serve non-expert users without requiring additional applications or proprietary infrastructure. The planned integration of a large language model for natural-language climate summaries in Version 2.0 is supported by recent evidence that context-aware LLM systems can produce more relevant and personalized outputs when environmental and user-profile signals are incorporated into the retrieval pipeline [19].

The identified limitation of temporal asymmetry between stations is not a fundamental architectural constraint but rather a data maturity issue that will be resolved as the Este and Oeste stations accumulate longer operational histories. The projected R^2 improvement to 0.72–0.80 after six months of simultaneous four-station operation is consistent with the expected reduction in model uncertainty when spatial gradient features become available across all zones.

6. Conclusions

This work demonstrated that it is possible to implement a fully functional hyperlocal climate prediction system for Colombian micro-zones using low-cost hardware, open-source software, and free cloud services. The successful implementation of the five system layers, from Raspberry Pi data collection to Grafana Cloud visualisation, validates the conceptual proposal and establishes a concrete experimental platform for subsequent research phases.

The main technical contributions of this work are: (1) the development of a robust parser for the non-standard data format generated by the Raspberry Pi stations, reusable in any similar deployment of the proposed architecture; (2) the design of a session-based feature engineering scheme that enables training machine learning models with temporally spaced and asynchronous data between stations, overcoming the conventional limitation of temporal alignment; (3) the integration of n8n as a low-code orchestrator for IoT climate prediction systems, with a fault-tolerant design that continues operating when one or more stations are inactive; and (4) the implementation of a Telegram-based prediction distribution system with persistent subscriptions that democratises access to predictions without requiring additional applications or proprietary notification infrastructure.

From a cost and accessibility perspective, the system demonstrates that the total implementation budget, estimated at 160–240 USD in hardware plus 0 USD/month in software infrastructure, is compatible with academic projects, community initiatives, and small

organisations in middle-income countries such as Colombia. This economic accessibility is fundamental for the replicability of the system in other micro-zones across the country.

Future work is organised into four priority lines. Version 1.3 will connect n8n directly to the MariaDB databases of the four Raspberry Pi stations via the n8n MySQL plugin, eliminating the CSV file dependency, and will deploy Flask on Render with a paid plan to guarantee 24/7 availability. Version 1.4 will retrain the model after six months of simultaneous operation of all four stations in their definitive triangulated locations within the 10 km^2 micro-zone, enabling the model to learn spatial gradients and complete seasonal patterns, with an expected R^2 improvement to 0.72–0.80. Version 1.5 will integrate historical and real-time data from the nearest IDEAM stations as additional model features, enabling external validation and potentially extending the prediction horizon to 1–3 h. Version 2.0 will integrate a large language model (Gemini 2.0 Flash) to generate climate analyses in accessible colloquial language for non-technical users [19], and will develop a custom web dashboard with an interactive micro-zone map visualising real-time temperature gradients [29–31].

Author Contributions: Conceptualization, C.A.G.M. and J.F.D.P.; methodology, C.A.G.M. and J.M.B.S.; software, C.A.G.M. and J.M.B.S.; validation, C.A.G.M. and D.N.; formal analysis, C.A.G.M.; investigation, C.A.G.M.; resources, V.R.Q.L.; data curation, C.A.G.M. and J.M.B.S.; writing—original draft preparation, C.A.G.M.; writing—review and editing, D.N., V.R.Q.L. and J.F.D.P.; visualization, C.A.G.M. and J.M.B.S.; supervision, V.R.Q.L. and J.F.D.P.; project administration, V.R.Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge financial support from CNPq (National Council for Scientific and Technological Development, grant 307137/2022-8) and CAPES (Coordination for the Improvement of Higher Education Personnel–Brazil, Finance Code 001). Colaboración Consejería de Educación de la Junta de Castilla y León, research group ESAL-EXPERT SYSTEM AND APPLICATIONS LAB (ESALAB). This research was partially funded by national funds through the FCT—Foundation for Science and Technology, I.P. within the scope of projects UIDB/04466/2025 and UIDP/04466/2025. This work was supported by FCT—Fundação para a Ciência e a Tecnologia, I.P., under project reference UID/PRR/00066/2025 and DOI identifier https://doi.org/10.54499/UID/PRR/00066/2025. This work was supported by FCT—Fundação para a Ciência e a Tecnologia, I.P., under project reference UID/00066/2025 and DOI identifier https://doi.org/10.54499/UID/00066/2025. This research was also partially funded by national funds through the FCT—Foundation for Science and Technology, I.P. within the scope of project 16881, LISBOA2030-FEDER-00816400, with DOI: https://doi.org/10.54499/2023.16583.ICDT.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and code supporting the results of this study are available upon reasonable request to the corresponding author.

Acknowledgments: The authors thank ISCTE-IUL (University Institute of Lisbon) and the University of Salamanca for their institutional support. Colaboración Consejería de educación de la Junta de Castilla y León grupo de investigación ESAL-EXPERT SYSTEM AND APPLICATIONS LAB (ESALAB).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Wang, X.A.; Liu, Z.; Liu, W.; Jiang, W.; Jiang, W.; Li, Y. Privacy-Preserving Revocable Certificateless Cloud Data Auditing Scheme Based on IoT Data Collection for Secure Medical Data Sharing. *IEEE Internet Things J.* **2026**, *13*, 28428–28444. [CrossRef]

2. Khan, S.; Mazhar, T.; Shahzad, T.; Tariq, M.U.; Ali, T.; Ayaz, M.; Jiang, W.; Hamam, H. The evolution of blockchain in smart buildings with IoT integration and future prospects. *Telemat. Inform. Rep.* **2026**, *21*, 100283. [CrossRef]
3. Yang, M.; Li, Z.; Liu, S. A machine learning based framework for predictive school management using student and faculty analytics. *Sci. Rep.* **2026**, *16*, 16224. [CrossRef] [PubMed]
4. Emon, M.M.H.; Khan, T. The mediating role of attitude towards the technology in shaping artificial intelli-gence usage among professionals. *Telemat. Inform. Rep.* **2025**, *17*, 100188. [CrossRef]
5. Zhou, D.; Shi, L.; Wang, B.; Xu, H.; Huang, W. Are Your Comments Positive? A Self-Distillation Contrastive Learning Method for Analyzing Online Public Opinion. *Electronics* **2024**, *13*, 2509. [CrossRef]
6. Firza, N.; Bakiu, A.; Monaco, A. Machine Learning for Quality Diagnostics: Insights into Consumer Elec-tronics Evaluation. *Electronics* **2025**, *14*, 939. [CrossRef]
7. Vasta, N.; Jajo, N.; Graf, F.; Zhang, L.; Biondi, F.N. Evaluating a Camera-Based Approach to Assess Cog-nitive Load During Manufacturing Computer Tasks. *Electronics* **2025**, *14*, 467. [CrossRef]
8. Noetzold, D.; Leithardt, V.R.Q.; De Paz Santana, J.F.; Barbosa, J.L.V. A Self-Adaptive Architecture for Predictive and Reinforcement-Based Optimization in Smart Environments. In *Proceedings of the 2025 International Symposium on Networks, Computers and Communications (ISNCC)*; IEEE: New York, NY, USA , 2025; pp. 1–6. [CrossRef]
9. Tari, Z.; Mahmud, R. Augmenting Digital Ecosystem Resilience Through Human-Centric Cybersecurity So-lutions. *IEEE Trans. Eng. Manag.* **2025**, *72*, 3892–3908. [CrossRef]
10. Ibrahim, F.; Schumacher, J.; Hofer, P.; Daseking, M. Decision-Making in virtual reality: An experimental study on virtual reality's effect on presence, task-load and decision-making performance within operational command. *Telemat. Inform. Rep.* **2026**, *21*, 100282. [CrossRef]
11. Sasongko, A.T.; Ekhsan, M.; Fatchan, M. Dataset on technology acceptance in E-learning: A PLS-SEM analysis using extended TAM among undergraduate students in Indonesia. *Telemat. Inform. Rep.* **2025**, *18*, 100192. [CrossRef]
12. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
13. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. Available online: https://scikit-learn.org (accessed on 15 April 2026).
14. Naskinova, I.; Kolev, M.; Karova, D.; Milev, M. Machine Learning-Based Blood Pressure Prediction Using Cardiovascular Disease Data: A Comprehensive Comparative Study. *Electronics* **2026**, *15*, 312. [CrossRef]
15. Santamaría, A.; Alfaro, M.; Antón, C.; Sánchez-Qui nones, B.; Ibarra, N.; Gil, A.; Reinoso, O.; Payá, L. Machine Learning-Based Predictive Model for Risk Stratification of Multiple Myeloma from Monoclonal Gammopathy of Undetermined Significance. *Electronics* **2025**, *14*, 3014. [CrossRef]
16. Holgado, A.; Portilla, J.; López-Fernández, D.; Redondo, L. Context-Aware Security and Post Quantum Cryptography Applied to IoT Networks. In *Proceedings of the 2024 4th Intelligent Cybersecurity Conference (ICSC)*; IEEE: New York, NY, USA, 2024. [CrossRef]
17. Luo, J.; Zhu, P.; Wang, Y.; Xiao, Z.; Li, J.; Kong, X.; Zhan, Y. A Data-Driven Multimodal Method for Early Detection of Coordinated Abnormal Behaviors in Live-Streaming Platforms. *Electronics* **2026**, *15*, 769. [CrossRef]
18. Narayan, A.; Madhu Kumar, S.D.; Chacko, A.M. Trust at risk: Detecting misinformation in LLM-generated product reviews and its implications for consumer behavior and platform governance. *Telemat. Inform. Rep.* **2026**, *21*, 100285. [CrossRef]
19. Karlović, R.; Rovis, M.; Smajić, A.; Sever, L.; Lorencin, I. Context-Aware Tourism Recommendations Using Retrieval-Augmented Large Language Models and Semantic Re-Ranking. *Electronics* **2025**, *14*, 4448. [CrossRef]
20. Riaz, W.; Ji, J.; Ullah, A. TriViT-Lite: A Compact Vision Transformer–MobileNet Model with Texture-Aware Attention for Real-Time Facial Emotion Recognition in Healthcare. *Electronics* **2025**, *14*, 3256. [CrossRef]
21. da Costa, P.; Noetzold, D.; García Ovejero, R.; Martín Esteban, R.; Leithardt, V.R.Q. PRISEC II: A comprehensive model for IoT security. In *New Trends in Disruptive Technologies, Tech Ethics and Artificial Intelligence (DiTTET 2025)*; Advances in Intelligent Systems and Computing; Springer: Cham, Swizerland, 2025; Volume 1465, pp. 147–158. [CrossRef]
22. Sohail, H.; Leithardt, V.R.Q.; Trigo, A. PRISEC III: Cryptographic Techniques for Enhanced Security. In Proceedings of the 2025 25th International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 27–30 May 2025. [CrossRef]
23. Noetzold, D.; Leithardt, V.R.Q.; Barbosa, J.L.V. Oraculum: A model for self-adaptive system optimization in smart environments. *Expert Syst. Appl.* **2026**, *315*, 131705. [CrossRef]
24. Noetzold, D.; De Paz Santana, J.F.; Barbosa, J.L.V.; Leithardt, V.R.Q. DN25: An adaptive quantum cryp-tography protocol for secure and efficient communication. In *Proceedings of WICI-UTPVE/COICIT 2025, Santiago de Veraguas, Panamá, 16 October 2025*; Zenodo: Geneva, Switzerland, 2026. [CrossRef]
25. *ISO 8601-1:2019*; Date and Time—Representations for Information Interchange—Part 1: Basic Rules. International Organization for Standardization: Geneva, Switzerland, 2019.

26. Kim, N.; Lim, H.; Li, Q.; Li, X.; Kim, S.; Kim, J. Enhancing Review-Based Recommendations Through Local and Global Feature Fusion. *Electronics* **2025**, *14*, 2540. [CrossRef]
27. Ari, I.; Balkan, K.; Pirbhulal, S.; Abie, H. Ensuring Security Continuum from Edge to Cloud: Adaptive Se-curity for IoT-based Critical Infrastructures using FL at the Edge. In *Proceedings of the 2024 IEEE International Conference on Big Data (BigData)*; IEEE: New York, NY, USA, 2024; pp. 4921–4929. [CrossRef]
28. Mirsadri, S.J.; Chaves, R.; Pedrosa, L. Energy-Aware Adaptive Security for Smart Farming (EAASF): A Hybrid IDS-IPS Framework with SDN-Orchestrated for Agriculture 4.0. In *Proceedings of the 2025 23rd International Symposium on Network Computing and Applications (NCA); Lisbon, Portugal, 5–7 November 2025*; IEEE: New York, NY, USA, 2025; pp. 328–329. [CrossRef]
29. n8n GmbH. n8n—Workflow Automation Tool. 2024. Available online: https://n8n.io (accessed on 22 May 2026).
30. Grafana Labs. Grafana: The Open Observability Platform. 2024. Available online: https://grafana.com (accessed on 10 June 2026).
31. Pallets Projects. Flask: A Lightweight WSGI Web Application Framework. 2024. Available online: https://flask.palletsprojects.com (accessed on 3 May 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.