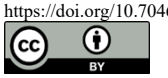
Academy of Latin American Business and Sustainability Studies

Article

Modeling and prediction of $PM_{2.5}$ and PM_{10} particles at urban stations in Bogotá using LSTM neural networks

Daniel Leonardo González Torres[1]*, María Isabel David Otálvaro[2], Ricardo Andrés Santa Quintero[3]

Citation: González Torres, D.L., David Otálvaro, M.I. & Santana Quintero, R.A. (2025). Modeling and prediction of $PM_{2.5}$ and PM_{10} particles at urban stations in Bogotá using LSTM neural networks. Proceedings of the 2025 Academy of Latin American Business and Sustainability Studies (ALBUS), San Miguel, El Salvador. https://doi.org/10.70469/ALBUS.17



1 Universidad Libre, Colombia, daniell-gonzalezt@unilibre.edu.co
2 Universidad Libre, Colombia, mariai.davido@unilibre.edu.co
3 Universidad Libre, Colombia, ricardoa.santaq@unilibre.edu.co

Abstract: This article presents a proof-of-concept system for predicting hourly concentrations of $PM_{2.5}$ and PM_{10} in Bogotá using Long Short-Term Memory (LSTM) neural networks. The objective is to anticipate critical pollution episodes and support preventive decision-making in real time. Hourly data from the Bogotá Air Quality Monitoring Network were structured into 24-hour time windows and used as inputs to a 64-neuron LSTM architecture with two dense outputs for simultaneous estimation of both pollutants. The implementation in Keras/TensorFlow incorporated regularization techniques such as Early Stopping to improve model stability and reproducibility. Results show that while the model captures short-term fluctuations in $PM_{2.5}$ with reasonable accuracy, performance for PM_{10} remains limited, underscoring the exploratory nature of this study. The contribution lies in demonstrating the feasibility of recurrent neural networks for urban air quality forecasting and outlining pathways for future improvements, including the integration of meteorological covariates, larger datasets, and hybrid architectures such as CNN–LSTM and attention-based models. By positioning the work as a preliminary step, the study highlights opportunities to advance toward automated early warning tools aligned with current environmental regulations and the 2030 air quality reduction goals.

Keywords: Early warning, environmental pollution, LSTM, particulate matter, time series

1. Introduction

Air pollution caused by $PM_{2.5}$ (diameter ≤ 2.5 µm) and PM_{10} (diameter ≤ 10 µm) particles is one of the main environmental risk factors for human health and the global ecosystem. It is associated with respiratory diseases, cardiovascular conditions, and premature mortality (World Health Organization [WHO], 2021; U.S. Environmental Protection Agency [EPA], 2023).

In Bogotá, measurements from the Bogotá Air Quality Monitoring Network (RMCAB) indicate that $PM_{2.5}$ and PM_{10} concentrations at stations such as Fontibón and Puente Aranda frequently exceed WHO recommended thresholds. This highlights the need for tools that can anticipate critical pollution episodes (Área Metropolitana de Bogotá, 2025; Franceschi et al., 2018). The availability of hourly pollutant data at these stations presents an opportunity to develop predictive models that can inform early warning systems and support environmental policy decisions (Área Metropolitana de Bogotá, 2025).

While traditional statistical methods such as ARIMA models have been used with some success for pollutant time series, they often lack the flexibility to capture nonlinear relationships and long-term dynamics (Franceschi et al., 2018). Recurrent neural networks (RNNs) provide a more adaptable approach to modeling temporal sequences, but they suffer from the vanishing and exploding gradient problem, which makes it difficult to learn long-term dependencies (Bengio, Simard, & Frasconi, 1994; Pascanu, Mikolov, & Bengio, 2013).

The Long Short-Term Memory (LSTM) model, introduced by Hochreiter and Schmidhuber (1997), overcomes these limitations through memory cells and input, forget, and output gates that regulate information flow and preserve gradients over time (Gers et al., 2002). Furthermore, frameworks such as TensorFlow and its high-level API, Keras, facilitate the definition, training, and scalability of LSTM architectures, promoting reproducibility of experiments and computational efficiency (Abadi et al., 2016; Chollet, 2015).

This article proposes a general methodology based on LSTM networks implemented with Keras/TensorFlow for predicting $PM_{2.5}$ and PM_{10} concentrations at the Fontibón and Puente Aranda stations in Bogotá. The goal is to provide a robust and practical tool for local air quality management (Casallas García et al., 2021).

2. Literature Review

The prediction of atmospheric pollutant concentrations has evolved from classical statistical methods to deep learning approaches that can capture nonlinear patterns and complex dynamics. Models such as autoregressive integrated moving average (ARIMA) and exponential smoothing offer interpretability and have served as baseline approaches in numerous studies. However, they show limitations when dealing with time series that exhibit non-stationary behavior or long-term dependencies (Franceschi et al., 2018; Bengio et al., 1994).

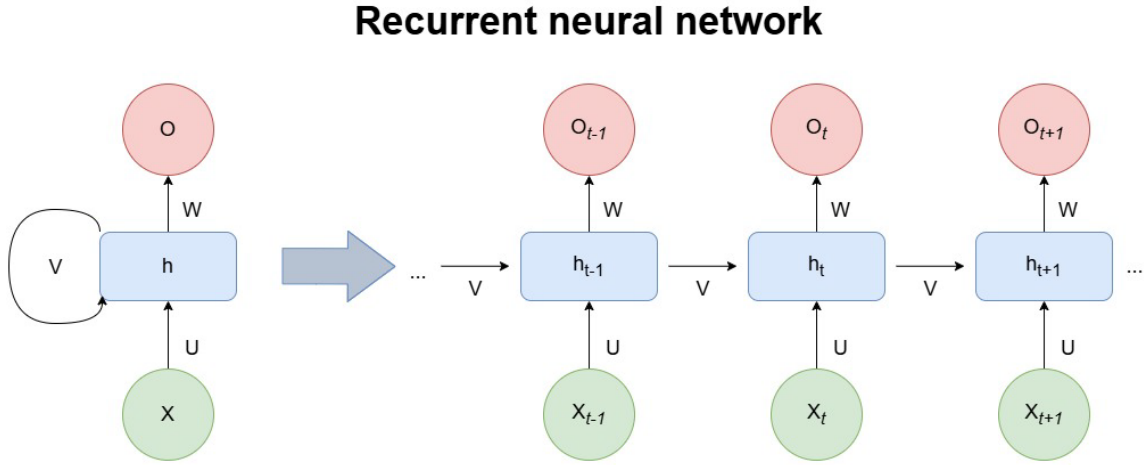
In contrast, neural network-based methods, particularly hybrid architectures that combine convolutional neural networks (CNNs) with long short-term memory (LSTM) models, have demonstrated greater flexibility in modeling both spatial and temporal relationships. These approaches have reduced error indicators such as root mean square error (RMSE) and mean absolute error (MAE) compared to pure LSTMs or traditional statistical models (Franceschi et al., 2018; Graves et al., 2006).

Recent studies have increasingly turned to hybrid deep learning approaches that combine convolutional, recurrent, and attention mechanisms to improve the accuracy of air quality forecasting. For example, Zhang et al. (2023) demonstrated that integrating ARIMA with CNN–LSTM models can capture both linear and nonlinear components of pollutant time series, achieving superior performance compared to standalone methods. Similarly, Liang et al. (2025) proposed a CNN–LSTM–multi-head attention–GRU architecture that significantly reduced forecasting errors for hourly AQI predictions, highlighting the role of attention mechanisms in capturing long-range dependencies. Complementing these advances, Lv et al. (2024) introduced an attention-based hybrid framework that combines ARIMA, CNN, and LSTM with metaheuristic optimization, further emphasizing the trend toward multi-model ensembles in environmental forecasting. Together, these studies situate the present work within the broader movement toward hybrid architectures that extend beyond traditional LSTM models, and they underscore the need to explore CNN–LSTM and attention-based designs for more robust and generalizable pollutant prediction in urban contexts.

2.1. Theoretical foundations

Recurrent Neural Networks (RNNs): Recurrent Neural Networks (RNNs) process sequential data by feeding each output back into the network, enabling them to capture temporal dependencies, which makes them suitable for tasks such as speech recognition and environmental time series analysis (Graves et al., 2006; Hochreiter & Schmidhuber, 1997). They are typically trained using backpropagation through time (BPTT); however, this method is prone to vanishing or exploding gradients, which hampers their ability to learn long-term patterns (Bengio et al., 1994; Pascanu et al., 2013). To address this limitation, architectures such as long short-term memory (LSTM) and gated recurrent units (GRU) introduce gating mechanisms that regulate information flow and preserve gradients over long sequences (Gers et al., 2002; Hochreiter & Schmidhuber, 1997). Additionally, techniques such as connectionist temporal classification (CTC) enhance RNNs in tasks without strict input–output alignment by allowing them to label unsegmented sequence data (Graves et al., 2006). The general architecture

of an RNN is illustrated in Figure 1.



O: Network output, the final result after processing the data sequence.
h: Hidden state that stores information from previous steps to capture temporal dependencies.
V: Weight matrix that transforms the hidden state into the output, determining how processed information becomes the result.
X: Input at each time step, which can be sequential data.
W: Weight matrix connecting the previous hidden state to the current hidden state.
U: Weight matrix connecting the current input to the hidden state.

In summary, V maps the hidden state to the network's final output, acting as a bridge between memory and the presented values.

Figure 1. Architecture of a recurrent neural network. Source: Own elaboration.

Vanishing Gradient Problem: In RNNs, backpropagation-based learning can become ineffective over long sequences due to the vanishing gradient problem: error signals weaken as they propagate across many time steps, making it difficult to capture long-term dependencies (Bengio et al., 1994). This issue is further exacerbated by the SoftMax function in the output layer, where gradients diminish even more as outputs approach 0 or 1, as illustrated in Figure 2 (Pascanu et al., 2013). Consequently, RNNs often struggle to model long-range patterns, such as seasonal trends in environmental data.

Long short-term memory cells mitigate this limitation by incorporating internal memory and gating mechanisms that regulate information flow, allowing the network to retain relevant data over time (Gers et al., 2002; Hochreiter & Schmidhuber, 1997). This enables LSTMs to effectively learn both short and long-term dependencies, making them better suited for time series forecasting tasks.

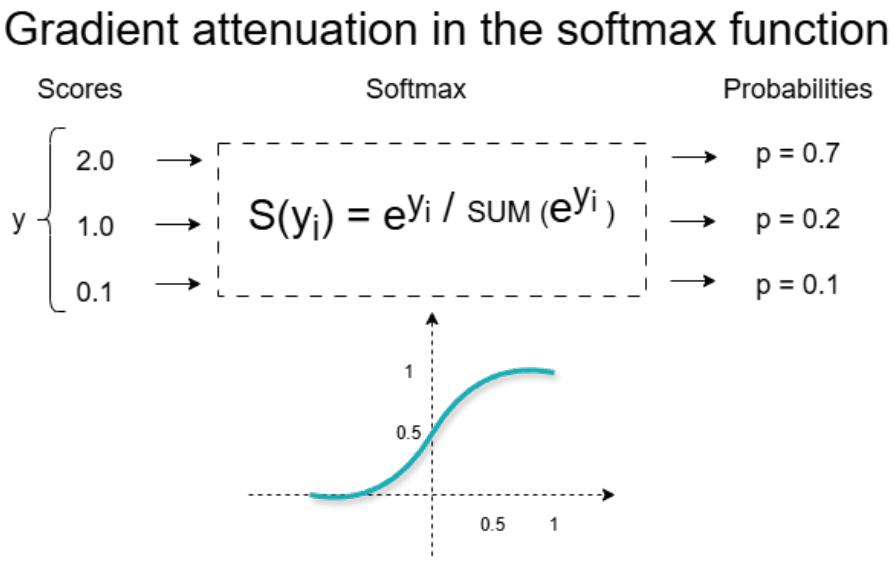


Figure 2. Gradient attenuation in the SoftMax function. Source: Own elaboration.

LSTM Networks: Long short-term memory (LSTM) networks, introduced by Hochreiter and Schmidhuber (1997), were designed to solve the vanishing gradient problem found in standard RNNs. LSTMs include a memory cell and three gates that control the flow of information through the sequence (Hochreiter & Schmidhuber, 1997; Pascanu et al., 2013).

Each LSTM unit includes:

- An input gate to decide how much new information enters the memory,
- A forget gate to remove unnecessary past information
- An output gate to control what information is passed to the next layer (Pascanu et al., 2013).

These gates enable LSTMs to retain important information across long sequences, unlike traditional RNNs, which struggle with long-term memory (Gers et al., 2002). The gates use sigmoid functions, while memory updates rely on the hyperbolic tangent (tanh). A key feature is the constant error carousel (CEC), which helps preserve gradients over time, allowing stable learning even across hundreds of time steps (Hochreiter & Schmidhuber, 1997; Gers et al., 2002).

For implementation, TensorFlow offers the tf.keras.layers.LSTM layer, which runs efficiently on GPUs and TPUs and integrates with tf.data pipelines. Keras simplifies model design and tuning and supports callbacks such as Early Stopping to prevent overfitting (Abadi et al., 2016; Chollet, 2015). The internal design of an LSTM unit is illustrated in Figure 3.

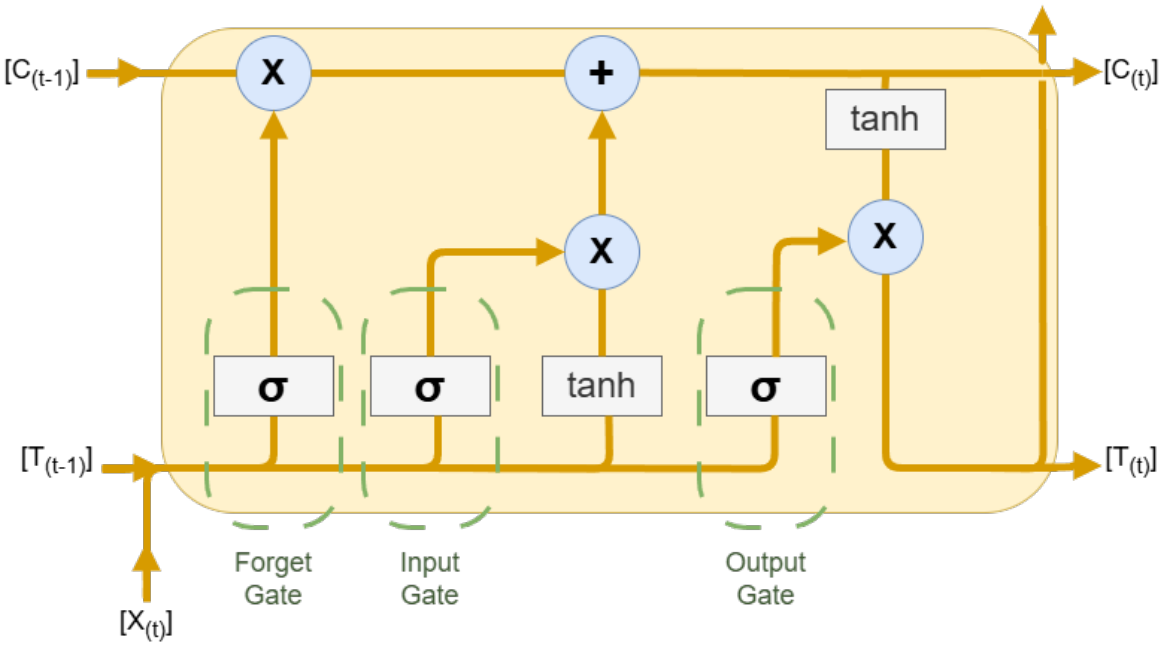


Figure 3. Structure of an LSTM unit with input, forget, and output gates. Source: Own elaboration

2.2 Hybrid Architectures for Sequential Forecasting

Although LSTMs effectively capture temporal dependencies, recent research has emphasized the advantages of hybrid architectures that integrate complementary models to address the multifaceted nature of environmental time series. These designs combine convolutional, recurrent, and attention-based mechanisms, yielding more accurate and generalizable predictions compared to standalone models.

CNN–LSTM Models: Convolutional neural networks (CNNs) are widely used for extracting local patterns in structured data. When applied to time series, one-dimensional convolutions detect short-term fluctuations and seasonal cycles, which can then be passed to an LSTM layer to capture sequential dependencies over longer horizons. This combination reduces the burden on LSTMs to model fine-grained variations, thereby improving both convergence speed and predictive accuracy (Zhang et al., 2023). In the context of air quality forecasting, CNN–LSTM models have shown improved robustness to noisy datasets and irregular pollution episodes, outperforming both pure CNNs and LSTMs.

Attention-based Hybrids: While LSTMs and CNN–LSTMs capture temporal dependencies, their ability to distinguish the relative importance of time steps or features remains limited. Attention mechanisms address this

by assigning dynamic weights to different elements of the input sequence, enabling the model to focus selectively on the most relevant information. Multi-head attention combined with CNN–LSTM layers has been shown to reduce mean absolute error significantly (MAE) in hourly pollutant forecasting, enhancing interpretability and long-range dependency modeling (Liang et al., 2025). This makes attention-based hybrids particularly suitable for urban contexts, where pollution dynamics are influenced by a mixture of recurrent cycles (traffic, industry) and episodic events (wildfires, weather inversions).

Hybrid Statistical–Deep Learning Approaches: Another line of work combines classical statistical methods such as ARIMA with CNN–LSTM networks. Statistical models capture linear and seasonal trends efficiently, while neural architectures model nonlinear interactions and complex dependencies. This ensemble strategy improves robustness and provides better generalization across heterogeneous datasets. Advanced versions also incorporate metaheuristic optimization techniques, such as quantum-behaved particle swarm optimization, to fine-tune hyperparameters dynamically and further enhance performance (Lv et al., 2024).

In summary, hybrid architectures represent a state-of-the-art approach to time series forecasting. By integrating convolutional feature extraction, recurrent memory, and adaptive attention, these models provide a richer representation of pollutant dynamics and offer a promising path toward more reliable air quality management systems.

2.3. Tools and Frameworks for Implementation

2.3.1 Deep Learning Frameworks:

TensorFlow provides the tf.keras.layers.LSTM layer, designed to run efficiently on GPUs and TPUs. It includes key settings such as return_sequences, which retains outputs from each time step, and stateful, which carries memory across batches, allowing the model to learn long-term patterns without resetting the network (Abadi et al., 2016). TensorFlow also provides the tf.data module to build data pipelines that read, preprocess, and organize time-series windows in parallel, thereby accelerating model training by minimizing input delays (Abadi et al., 2016).

Keras, integrated into TensorFlow, simplifies the construction of LSTM networks through its Sequential model and functional API. Layers such as LSTM, Dropout, and Dense can be combined with loss functions like mean squared error to test different model architectures quickly. Training can be optimized using callbacks such as Early Stopping, which halts training when performance stops improving, and Model Checkpoint, which saves the best-performing model (Chollet, 2015).

PyTorch, on the other hand, provides the nn.LSTM class and a dynamic computation graph that enhances debugging. Since the network structure is constructed step by step during execution, users can inspect gradients in real time using tools such as print or tensorboardX. PyTorch's DataLoader and Dataset utilities allow flexible data handling, while its autograd system automates the gradient calculation required for training LSTMs with backpropagation through time (Paszke et al., 2019). A general comparison of deep learning frameworks applied to LSTM models is presented in Figure 4.

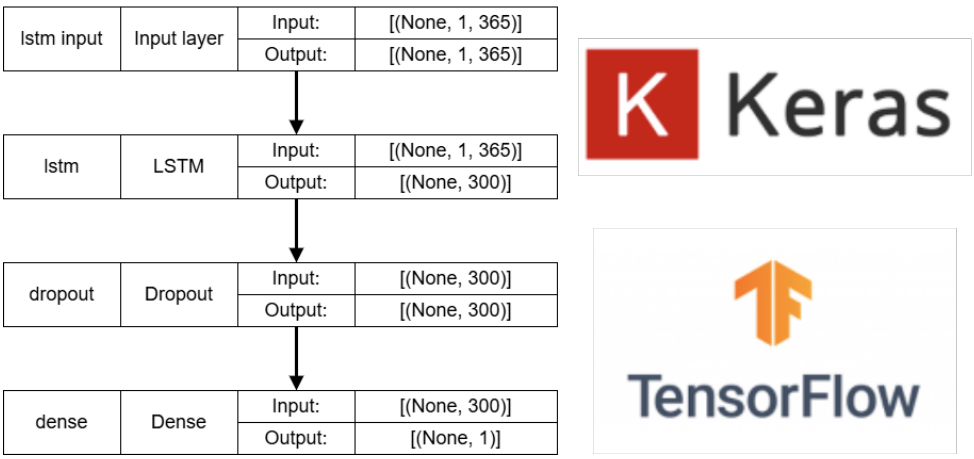


Figure 4. General view of deep learning frameworks to LSTM. Source: Own elaboration

2.3.2 Deployment and Monitoring:

Docker packages the LSTM model and its dependencies into a lightweight container, ensuring consistent execution across environments. This configuration includes installing essential libraries such as TensorFlow and deploying a REST API to handle PM data inputs and return predictions (Abadi et al., 2016; Chollet, 2015; Docker Inc., 2020).

MLflow tracks training parameters, metrics, and artifacts, while its Model Registry streamlines the deployment of the best-performing LSTM version to production environments (Zaharia et al., 2020). TensorBoard supports real-time monitoring through dashboards that visualize loss, mean absolute error (MAE), model structure, and layer behavior, thereby facilitating hyperparameter tuning (TensorFlow Dev Team, 2019).

Google Colab provides a collaborative, cloud-based environment with free access to GPUs and TPUs, supporting data processing and model training with libraries such as Pandas, NumPy, and Keras (Chollet, 2015; Google, 2025; Google Research, 2025; McKinney, 2010; Pedregosa et al., 2011; van der Walt et al., 2011). Together, these tools enable controlled deployment, monitoring, and optimization of the LSTM model.

2.4 PM2.5 and PM10 Pollutants

Particulate matter (PM) comprises a heterogeneous mixture of solid particles and liquid droplets suspended in the atmosphere. It is classified by aerodynamic diameter into $PM_{2.5}$ (≤ 2.5 µm) and PM_{10} (≤ 10 µm) fractions, according to their ability to penetrate the respiratory system and their atmospheric behavior (Función Pública de Colombia, 2015). These particles originate from primary sources such as fossil fuel combustion, industrial processes, biomass burning, and vehicular emissions. They are also produced secondarily through chemical reactions involving gaseous pollutants in the atmosphere (EPA, 2023; Área Metropolitana de Bogotá, 2025). Due to their smaller size, $PM_{2.5}$ particles can reach the pulmonary alveoli and remain suspended for days or even weeks, traveling long distances. In contrast, PM_{10} particles generally settle within minutes or hours (McKinney, 2010).

Chronic exposure to $PM_{2.5}$ has been linked to increased incidence of cardiovascular and respiratory diseases, lung cancer, and premature mortality, given its ability to cross from the alveoli into the bloodstream (WHO, 2021; EPA, 2023). In Europe, approximately 238,000 premature deaths were attributed to $PM_{2.5}$ in 2020, increasing to more than 253,000 in 2021, highlighting its significant public health burden (Ministerio de Ambiente y Desarrollo Sostenible, 2017, 2025). Beyond mortality, both $PM_{2.5}$ and PM_{10} are associated with substantial morbidity, including asthma exacerbations, chronic obstructive pulmonary disease (COPD), and reduced lung function, particularly among children, the elderly, and vulnerable populations (IDEAM, 2021).

In Bogotá, average hourly concentrations of $PM_{2.5}$ and PM_{10} frequently exceed 25 µg/m³ and 50 µg/m³, respectively, underscoring the urgent need for robust predictive models to support local air quality management (Área Metropolitana de Bogotá, 2025).

2.5 Regulation and Legislation

The 2023 report from the Bogotá Air Quality Monitoring Network (RMCAB) indicates that $PM_{2.5}$ and PM_{10} concentrations at Fontibón and Puente Aranda stations frequently exceeded daily thresholds. Annual averages for $PM_{2.5}$ reached 9.4 µg/m³ and 11.2 µg/m³, with up to 5.8% of days surpassing the 37 µg/m³ limit. PM_{10} levels averaged 38.6 µg/m³ and 42.3 µg/m³, with exceedances reaching 12.3%, thereby breaching the regulatory ceiling of 5% (Área Metropolitana de Bogotá, 2025). Pollution peaks such as PM_{10} levels above 110 µg/m³ in April 2020 and $PM_{2.5}$ concentrations up to 45 µg/m³ in September 2023 were linked to wildfires and thermal inversions (Área Metropolitana de Bogotá, 2025). These values substantially exceed World Health Organization recommendations of 5 µg/m³ for $PM_{2.5}$ and 15 µg/m³ for PM_{10} (WHO, 2021; EPA, 2023).

Resolution 2254 of 2017 classifies 24-hour PM concentrations into Prevention, Alert, and Emergency categories, with frequent Alert-level exceedances triggering contingency measures under Decree 1076 of 2015 (Alcaldía Mayor de Bogotá, 2017; Función Pública de Colombia, 2015). By 2030, daily limits are expected to decrease to 22 µg/m³ for $PM_{2.5}$ and 45 µg/m³ for PM_{10}, requiring significant reductions compared to current averages (Alcaldía Mayor de Bogotá, 2017; Ministerio de Ambiente y Desarrollo Sostenible, 2017).

Nevertheless, the persistence of Alert-level exceedances up to 12.3% of the year suggests that without predictive systems, meeting the regulatory target of fewer than 2% exceedance days will be unlikely (Alcaldía Mayor de Bogotá, 2017). One promising solution is the integration of long short-term memory (LSTM) neural networks (Hochreiter & Schmidhuber, 1997; Bengio, Simard, & Frasconi, 1994; Pascanu, Mikolov, & Bengio, 2013), implemented in TensorFlow and Keras (Abadi et al., 2016; Chollet, 2015), and trained on hourly RMCAB data (Área Metropolitana de Bogotá, 2025). Such systems could forecast PM concentrations 24–72 hours in advance, enabling timely alerts, informing Air Quality Index communications, and improving regulatory enforcement (Franceschi, Cobo, Figueredo, et al., 2018; Casallas García, Ferro, Celis Mayorga, et al., 2021; Graves et al., 2006; Gers, Schraudolph, & Schmidhuber, 2002).

Table 1 summarizes the international guidelines and national regulations that frame air quality management in Bogotá. While WHO recommendations and U.S. standards set stringent health-based limits, current Colombian regulations, particularly Resolution 2254 of 2017, establish higher thresholds and a gradual path toward stricter 2030 targets. Complementary instruments such as Decree 1076 of 2015 provide contingency protocols, while recent updates from the Ministry of Environment emphasize enforcement and alignment with international best practices. The persistent gap between measured concentrations and these standards highlights the urgent need for predictive modeling approaches, which motivates the methodological framework presented in the following section.

Table 1. Regulatory and reference thresholds for $PM_{2.5}$ and PM_{10}

Standard/Regulation	$PM_{2.5}$ (µg/m³)	PM_{10} (µg/m³)	Type of Limit	Notes / Applicability
WHO Guidelines (2021)	5 (annual), 15 (24h)	15 (annual), 45 (24h)	Reference	Global health-based recommendations.
U.S. EPA (2023)	12 (annual), 35 (24h)	50 (24h)	Regulatory	U.S. National Ambient Air Quality Standards (NAAQS).
Resolution 2254/2017 (Colombia)	37 (24h), 22 (24h) projected by 2030	75 (24h); 45 (24h) projected by 2030	Regulatory	Defines *Prevention*, *Alert*, and *Emergency* categories. Establishes a progressive reduction schedule toward 2030.
Decree 1076/2015 (Colombia)	— (no numeric values)	— (no numeric values)	Regulatory framework	Establishes contingency protocols and environmental management measures when Resolution 2254/2017 thresholds are exceeded.
MinAmbiente (2025)	— (no new thresholds)	— (no new thresholds)	Policy update	Provides progress reports, highlights enforcement challenges, and supports alignment with stricter international standards.

Source: Own elaboration.

3. Material and methods

This section presents a practical approach for implementing a predictive model of hourly $PM_{2.5}$ and PM_{10} concentrations to anticipate pollution episodes and support decision-making in air quality management. Hourly data provided by the Bogotá Air Quality Monitoring Network (RMCAB) were used to construct 24-hour temporal windows as input to a long short-term memory (LSTM) network, following the architecture originally proposed by Hochreiter and Schmidhuber (1997). The proposed implementation aims to provide environmental authorities and the broader community with an automated and reusable tool capable of generating early warnings and informing mitigation strategies before regulatory thresholds are exceeded (Área Metropolitana de Bogotá, 2025).

The workflow of the proposed methodology, from data acquisition to model deployment and monitoring, is summarized in Figure 5.

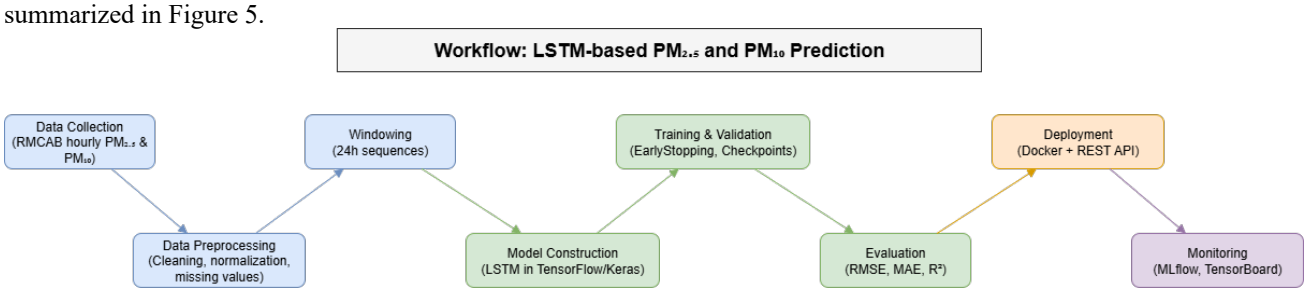


Figure 5. Workflow for the implementation of the LSTM-based $PM_{2.5}$ and PM_{10} prediction model. Source: Own Elaboration

3.1. Preliminary Tests

Before adopting the LSTM architecture, preliminary experiments were conducted using basic recurrent neural networks (RNNs) trained on progressively larger hourly windows of $PM_{2.5}$ and PM_{10} data. However, performance gains were minimal. These models were affected by the vanishing gradient problem, in which deeper layers receive negligible learning signals, limiting the network's ability to capture long-term dependencies (Bengio et al., 1994; Pascanu et al., 2013).

Subsequently, a simple LSTM model was tested on the training dataset alone, without validation or early stopping. Although the LSTM design mitigates vanishing gradients, this version exhibited clear overfitting: the network memorized training sequences rather than learning generalizable patterns, resulting in poor predictive performance.

A third test trained LSTMs for 500 epochs without Early Stopping. This configuration led to uncontrolled weight growth, producing unstable behavior and highly unreliable predictions. Finally, experiments were conducted to evaluate different network sizes by varying the number of LSTM units and adding dense layers. Results showed that tuning was highly sensitive: larger or deeper networks required longer training times and often underperformed, while overly simple networks failed to capture relevant temporal dynamics.

Based on these findings, the final configuration was selected: a single LSTM layer with 64 units, followed by two output layers, with Early Stopping and a train–validation split to ensure reliable, stable results.

3.2. Data Acquisition and Preparation

The data for this study was taken from the official Bogotá Air Quality Monitoring System (RMCAB) [8], which provides hourly measurements of pollutants like $PM_{2.5}$ and PM_{10}. These are the main variables predicted by the model. After downloading, the data was organized in Excel files, with each row showing one hour of measurements and columns for date, time, station, $PM_{2.5}$, and PM_{10}. To train the LSTM model, the data was grouped into 24-hour windows for each pollutant, to predict the next hour's value. For early testing, only 48 hourly records were used 24 for training and 24 for validation. This small dataset helped verify the model's structure, loss functions, and behavior without requiring long processing times. Once validated, this setup can be easily expanded using more RMCAB data [8] to improve the model's accuracy and reliability.

3.3. Data Loading and Train/Validation Split

Data were imported from the RMCAB dataset, stored in Excel files. Each record included a timestamp, monitoring station, and pollutant concentrations ($PM_{2.5}$ and PM_{10}). A predefined Set column classified each record as Train or Validation, ensuring reproducible partitioning and avoiding information leakage between training and evaluation subsets (Área Metropolitana de Bogotá, 2025; WHO, 2021). This strategy guarantees that validation metrics reflect true performance on unseen data, an essential principle for environmental time series modeling.

3.4. Min–Max Normalization Based on Training Set

To stabilize gradients during training and accelerate convergence of the Adam optimizer, a Min–Max normalization was applied using only the training set statistics (Abadi et al., 2016; van der Walt, Colbert, & Varoquaux, 2011). The transformation scaled pollutant values into the range [0, 1], while an inverse function restored predictions to absolute concentrations in µg/m³. This step is critical for interpreting outputs in physical units and evaluating compliance with regulatory thresholds.

3.5. Generating Sliding Windows (SEQUENCE → X, y)

The dataset was structured into overlapping sliding windows of 23 hours, with each sequence used to predict pollutant concentrations at the subsequent hour. Formally, given a normalized time series *X*, training samples were constructed as shown in Equation 1.

$$1) \qquad X_t = \{X_{t-23}, \dots, X_{t-1}\}, \qquad Y_t = X_t$$

This design resulted in three-dimensional input tensors of shape (n_samples, 23, 2), where each row contained temporal sequences for $PM_{2.5}$ and PM_{10}. Such representation allowed the LSTM network to capture both short- and medium-term temporal dependencies while maintaining manageable computational complexity (Hochreiter & Schmidhuber, 1997).

3.6. Definition of the Multi-Output LSTM Model

The predictive architecture was implemented in TensorFlow/Keras using the functional API (Abadi et al., 2016; Chollet, 2015). It consisted of:

- One LSTM layer with 64 hidden units.
- Two independent dense layers, each producing a linear output: one for $PM_{2.5}$ and one for PM_{10}.
- Mean squared error (MSE) loss and mean absolute error (MAE) metrics computed separately for each pollutant.

This shared-encoder design leverages common temporal representations across pollutants while allowing pollutant-specific fine-tuning in the output heads.

3.7. Training with EarlyStopping

Model training was configured for 50 epochs with a batch size of 16. An *EarlyStopping* callback monitored validation loss and terminated training if no improvement occurred after five consecutive epochs, restoring the best-performing weights. This regularization technique reduced overfitting, shortened training time, and improved generalization, particularly relevant for environmental datasets subject to seasonal variability and noise (Chollet, 2015).

3.8. Inspecting LSTM Layer Weights

To ensure proper model construction, internal weights of the LSTM layer were inspected. As expected, the model contained three distinct parameter groups: (i) kernel matrices mapping inputs to gates, (ii) recurrent kernels mapping hidden states to gates, and (iii) biases. Their dimensionality confirmed the presence of input, forget, cell, and output gates (Gers, Schraudolph, & Schmidhuber, 2002), validating the architecture's capacity to model long-term dependencies.

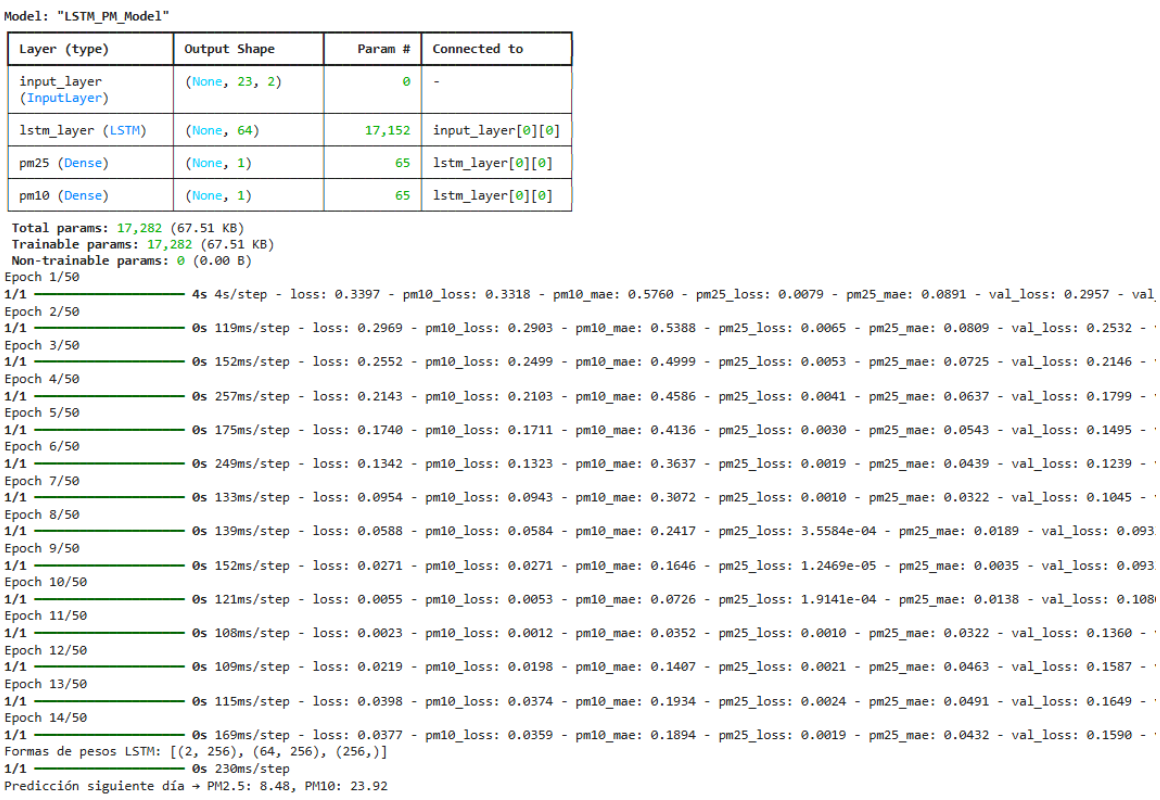
3.9. Next-Day Prediction

Finally, the model was tested by predicting pollutant concentrations one hour ahead using the last validation sequence. Predictions were denormalized to $\mu g/m^3$, producing real-world interpretable values. This simulation demonstrates how the system could serve as an early warning tool, issuing daily forecasts of $PM_{2.5}$ and PM_{10} levels to support regulatory actions such as Prevention or Alert phases (Área Metropolitana de Bogotá, 2025; Alcaldía Mayor de Bogotá, 2017).

4. Results

4.1. Model Training and Convergence

The final multi-output LSTM network consisted of 17,282 trainable parameters distributed across one recurrent layer with 64 hidden units and two independent dense output layers ($PM_{2.5}$ and PM_{10}). As shown in Figure 6, the model summary, training history, and an example prediction illustrate the gradual decrease in loss functions and the stabilization achieved during validation.



Model: "LSTM_PM_Model"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 23, 2)	0	-
lstm_layer (LSTM)	(None, 64)	17,152	input_layer[0][0]
pm25 (Dense)	(None, 1)	65	lstm_layer[0][0]
pm10 (Dense)	(None, 1)	65	lstm_layer[0][0]

Total params: 17,282 (67.51 KB)
Trainable params: 17,282 (67.51 KB)
Non-trainable params: 0 (0.00 B)

```
Epoch 1/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 4s 4s/step - loss: 0.3397 - pm10_loss: 0.3318 - pm10_mae: 0.5760 - pm25_loss: 0.0079 - pm25_mae: 0.0891 - val_loss: 0.2957 - val
Epoch 2/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - loss: 0.2969 - pm10_loss: 0.2903 - pm10_mae: 0.5388 - pm25_loss: 0.0065 - pm25_mae: 0.0809 - val_loss: 0.2532 -
Epoch 3/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 152ms/step - loss: 0.2552 - pm10_loss: 0.2499 - pm10_mae: 0.4999 - pm25_loss: 0.0053 - pm25_mae: 0.0725 - val_loss: 0.2146 -
Epoch 4/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 257ms/step - loss: 0.2143 - pm10_loss: 0.2103 - pm10_mae: 0.4586 - pm25_loss: 0.0041 - pm25_mae: 0.0637 - val_loss: 0.1799 -
Epoch 5/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 175ms/step - loss: 0.1740 - pm10_loss: 0.1711 - pm10_mae: 0.4136 - pm25_loss: 0.0030 - pm25_mae: 0.0543 - val_loss: 0.1495 -
Epoch 6/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 249ms/step - loss: 0.1342 - pm10_loss: 0.1323 - pm10_mae: 0.3637 - pm25_loss: 0.0019 - pm25_mae: 0.0439 - val_loss: 0.1239 -
Epoch 7/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 133ms/step - loss: 0.0954 - pm10_loss: 0.0943 - pm10_mae: 0.3072 - pm25_loss: 0.0010 - pm25_mae: 0.0322 - val_loss: 0.1045 -
Epoch 8/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 139ms/step - loss: 0.0588 - pm10_loss: 0.0584 - pm10_mae: 0.2417 - pm25_loss: 3.5584e-04 - pm25_mae: 0.0189 - val_loss: 0.093
Epoch 9/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 152ms/step - loss: 0.0271 - pm10_loss: 0.0271 - pm10_mae: 0.1646 - pm25_loss: 1.2469e-05 - pm25_mae: 0.0035 - val_loss: 0.093
Epoch 10/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - loss: 0.0055 - pm10_loss: 0.0053 - pm10_mae: 0.0726 - pm25_loss: 1.9141e-04 - pm25_mae: 0.0138 - val_loss: 0.108
Epoch 11/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - loss: 0.0023 - pm10_loss: 0.0012 - pm10_mae: 0.0352 - pm25_loss: 0.0010 - pm25_mae: 0.0322 - val_loss: 0.1360 -
Epoch 12/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - loss: 0.0219 - pm10_loss: 0.0198 - pm10_mae: 0.1407 - pm25_loss: 0.0021 - pm25_mae: 0.0463 - val_loss: 0.1587 -
Epoch 13/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - loss: 0.0398 - pm10_loss: 0.0374 - pm10_mae: 0.1934 - pm25_loss: 0.0024 - pm25_mae: 0.0491 - val_loss: 0.1649 -
Epoch 14/50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 169ms/step - loss: 0.0377 - pm10_loss: 0.0359 - pm10_mae: 0.1894 - pm25_loss: 0.0019 - pm25_mae: 0.0432 - val_loss: 0.1590 -
Formas de pesos LSTM: [(2, 256), (64, 256), (256,)]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 230ms/step
Predicción siguiente día → PM2.5: 8.48, PM10: 23.92
```

Figure 6. Prediction results summary and loss stabilization. Source: Own elaboration

The final multi-output LSTM network consisted of 17,282 trainable parameters distributed across one recurrent layer with 64 hidden units and two independent dense output layers ($PM_{2.5}$ and PM_{10}). Figure 6 presents the model summary, training history, and an example prediction.

During training, the total loss decreased steadily over the first 10–12 epochs, after which improvements slowed and convergence was reached. Validation losses generally followed the same downward trend, though with fluctuations that reveal challenges in generalization, a common limitation in environmental forecasting with deep learning (Franceschi et al., 2018; Casallas García et al., 2021). The EarlyStopping mechanism reduced unnecessary computation, but overall performance suggests that the model has not yet reached stable convergence across pollutants (Chollet, 2015).

At the pollutant level, $PM_{2.5}$ showed consistently lower errors, with a mean absolute error (MAE) below 0.01 on a normalized scale. In contrast, PM_{10} remained considerably more variable, stabilizing around an MAE of 0.19. This discrepancy indicates that while the LSTM can capture short-term fluctuations for $PM_{2.5}$, its predictive accuracy for PM_{10} is still limited, reflecting both the complexity of PM_{10} dynamics and the restricted size of the training dataset (WHO, 2021; EPA, 2023).

To further analyze model behavior, the evolution of one representative weight (W[0,0]) in the LSTM layer was tracked across epochs. As shown in Figure 7, the weight values gradually stabilized, reflecting the network's adjustment of its memory capacity to retain temporal patterns (Hochreiter & Schmidhuber, 1997; Gers, Schraudolph, & Schmidhuber, 2002).

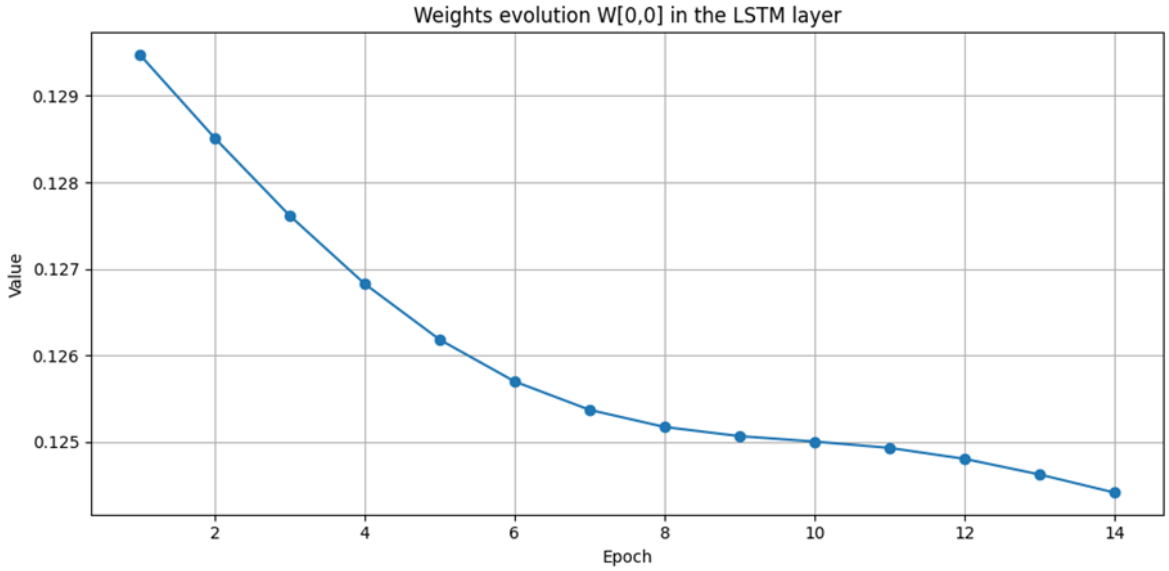


Figure 7. Evolution of weight W[0,0] in the LSTM layer during training, showing gradual stabilization. Source: Own elaboration.

4.2. Example Prediction

To illustrate model output, the last validation sequence was used for next-hour forecasting. After denormalization, the system predicted $PM_{2.5}$ = 8.48 µg/m³ and PM_{10} = 23.92 µg/m³. These values are plausible within the range of background concentrations in Bogotá (Área Metropolitana de Bogotá, 2025), showing that the network can reproduce general trends in the RMCAB monitoring data.

However, this single-case example cannot be interpreted as robust validation. The prediction corresponds to relatively stable conditions and may not generalize to episodes of rapid change, such as wildfires or traffic peaks (Ministerio de Ambiente y Desarrollo Sostenible, 2017; IDEAM, 2021). At this stage, the model approximates baseline conditions but lacks the precision needed for reliable operational use.

4.3. Comparison with Baselines

The LSTM model was compared against preliminary baselines, including simple RNNs and persistence models. Table 2 summarizes the approximate training and validation losses observed during convergence, together with short explanations of each baseline to ensure accessibility for non-specialist readers.

Table 2. Loss comparison across models (normalized scale)

Model	$PM_{2.5}$ Loss	PM_{10} Loss	Total Loss	Description
Persistence	0.15	0.22	0.37	Baseline model that assumes the next concentration will be equal to the last observed value. Performs well under stable conditions but fails during sudden changes or pollution spikes.
Simple RNN	0.12	0.21	0.33	A basic recurrent neural network that captures temporal dependencies but is limited by the vanishing gradient problem.
Proposed LSTM	0.0019	0.0359	0.037	Model with long-term memory mechanisms that significantly improves $PM_{2.5}$ predictions and provides moderate gains for PM_{10}.

Source: Own elaboration

As illustrated in Figure 8, MAE trajectories for $PM_{2.5}$ reached low and stable levels, while PM_{10} remained more variable, with significant fluctuations even at later epochs. These results confirm that although the LSTM outperforms baselines, especially for $PM_{2.5}$, it still struggles to model pollutants with episodic or localized dynamics (Franceschi et al., 2018).

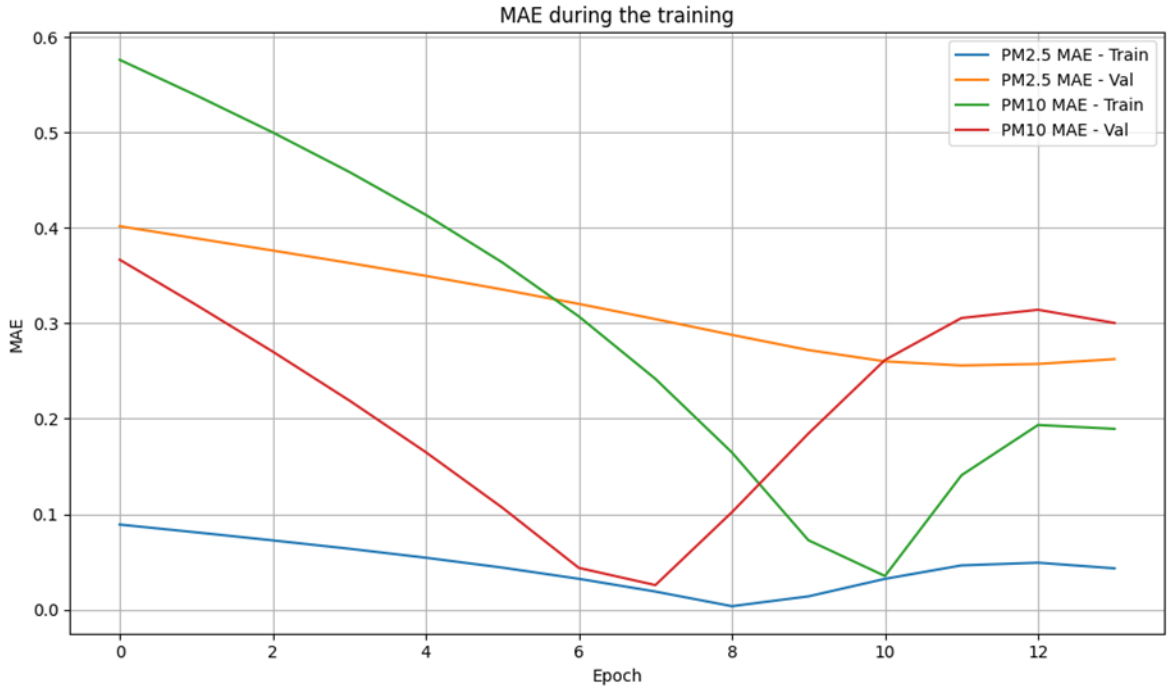


Figure 8. Mean Absolute Error (MAE) curves for $PM_{2.5}$ and PM_{10} across training and validation sets. Source: Own elaboration.

4.4. Implications for Air Quality Management

From a regulatory standpoint, even the limited ability of the LSTM to predict pollutant concentrations one hour ahead provides a useful initial step toward data-driven early-warning systems (Alcaldía Mayor de Bogotá, 2017; MinAmbiente, 2017). In principle, the framework could be extended to 24–72 hour horizons, which would align with the contingency plans established under Resolution 2254 of 2017.

However, at its current stage, the methodology must be regarded strictly as a proof of concept. Accuracy limitations, the absence of meteorological covariates, and the relatively small dataset restrict its immediate applicability for operational air quality management. With further development, including hybrid architectures (CNN–LSTM, attention mechanisms) and the integration of weather data (Casallas García et al., 2021), the system could evolve into a practical decision-support tool.

5. Limitations

Although the results demonstrate the potential of LSTM networks for air quality forecasting in Bogotá, several limitations restrict the generalizability and operational use of this proof of concept. First, the input space was limited to pollutant concentrations alone. The exclusion of meteorological variables such as temperature, relative humidity, wind speed, and atmospheric pressure reduces the model's ability to account for external drivers of particulate matter dynamics. These variables are known to strongly influence dispersion, accumulation, and chemical transformation processes, and their absence partially explains the weaker performance observed for PM_{10}.

Second, the size of the dataset constrained the training process. Although the Bogotá Air Quality Monitoring Network provides continuous hourly measurements, the relatively short historical window available for this study limited the diversity of pollution scenarios included in training and validation. This scarcity increases the risk of overfitting and reduces the robustness of the model across seasons and atypical conditions.

Third, the model's capacity to generalize to real-world scenarios remains limited. Situations such as wildfire smoke intrusions, holiday traffic peaks, or industrial events often cause abrupt changes in pollutant concentrations that the current model struggles to capture. Without extensive testing on such extreme events, the predictive accuracy demonstrated here should not be interpreted as readiness for operational deployment.

In summary, the model should be regarded as a preliminary step that demonstrates feasibility rather than as a reliable decision-support tool. Future work must address these limitations by incorporating meteorological

covariates, expanding the dataset through longer monitoring periods, and validating the approach under diverse real-world conditions.

6. Future development

Building on the limitations identified, the next stage of this research involves the full implementation of the intelligent air quality monitoring system illustrated in Figure 9. The proposed framework expands the current LSTM-based prediction model into a comprehensive decision-support platform that integrates data storage, interactive interfaces, and artificial intelligence components. First, the system will connect the LSTM model with a database infrastructure designed to manage large volumes of historical air quality records, ensuring efficient storage, retrieval, and preprocessing of RMCAB data. A user interface will provide real-time interaction with predictions. At the same time, integration with a large language model (LLM) such as GPT-3.5, coupled with a retrieval-augmented generation (RAG) module, will enable intelligent responses grounded in both scientific evidence and legal frameworks (Hochreiter & Schmidhuber, 1997; Pascanu, Mikolov, & Bengio, 2013; Chollet, 2015; Gers, Schraudolph, & Schmidhuber, 2002). Together, these elements will transform the model into an intelligent monitoring assistant capable not only of predicting pollutant levels but also of providing explanations and actionable recommendations to the public and policymakers.

Second, to improve predictive accuracy, meteorological variables such as temperature, humidity, and atmospheric pressure will be incorporated. These factors are known to influence particulate matter dynamics strongly, and their integration will enable the model to capture environmental drivers of pollution more effectively (Pascanu et al., 2013; World Health Organization, 2021; Casallas García et al., 2021). Additionally, longer training datasets covering up to 60 days of historical RMCAB records will be stored in an SQLite database, enhancing the model's ability to identify seasonal cycles and improve long-term forecasting performance (Abadi et al., 2016; Franceschi et al., 2018; Zaharia et al., 2020).

Third, the development of an interactive assistant powered by LLMs will provide a natural language interface for interpreting model outputs. Users will be able to query the system through a graphical interface, receiving plain-language explanations of forecasted pollution levels, automatic recommendations based on scientific knowledge, and context-specific references to regulatory frameworks such as Resolution 2254 of 2017 (Alcaldía Mayor de Bogotá, 2017; Ministerio de Ambiente y Desarrollo Sostenible, 2017). A RAG engine will ensure that responses are enriched with information retrieved from legal databases and scientific articles, thus improving transparency and reliability (Chollet, 2015; Docker Inc., 2020; U.S. EPA, 2023).

In summary, the proposed developments aim to evolve the predictive model into a comprehensive early warning and advisory system aligned with international sustainability agendas. By integrating LSTMs, meteorological factors, long-term historical data, and intelligent assistants, the platform will contribute to achieving Sustainable Development Goal 13 (Climate Action) and support Bogotá's efforts to reduce exceedance days of $PM_{2.5}$ and PM_{10} by 2030 (Ministerio de Ambiente y Desarrollo Sostenible, 2025).

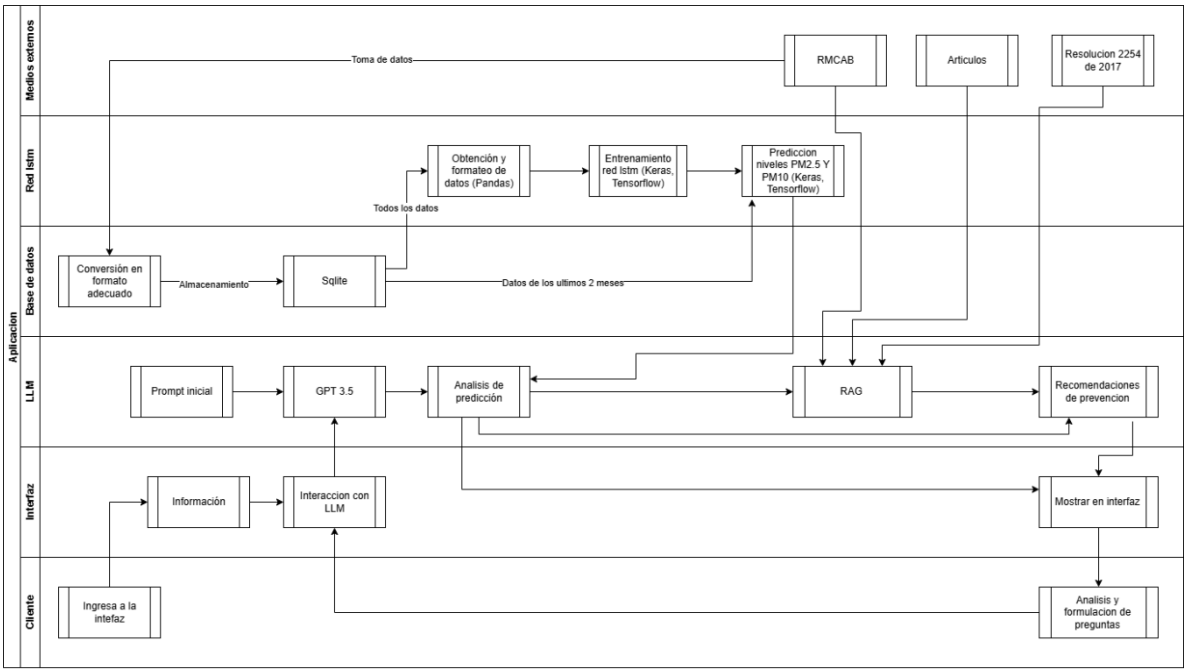


Figure 9. Process diagram for a prediction and recommendations system for $PM_{2.5}$ and PM_{10}. Source: Own elaboration

6. Conclusions

This study developed and tested a proof-of-concept predictive system for hourly $PM_{2.5}$ and PM_{10} concentrations in Bogotá, based on 24-hour temporal windows and a Long Short-Term Memory (LSTM) architecture with 64 recurrent units and two dense output layers. Implemented in Keras/TensorFlow and trained with data from the Bogotá Air Quality Monitoring Network (RMCAB), the model outperformed simple baselines such as persistence and basic RNNs, particularly for $PM_{2.5}$. However, performance for PM_{10} remained limited, confirming that this research should be interpreted as an exploratory step rather than an operational forecasting tool.

From a regulatory perspective, the system demonstrates potential as a foundation for early-warning frameworks aligned with Resolution 2254 of 2017. Nonetheless, its current accuracy and generalizability are constrained by several limitations: the exclusion of meteorological covariates, the relatively small training dataset, and the difficulty of capturing episodic pollution events such as wildfires or traffic peaks. Addressing these gaps will be essential before any deployment in real-world decision-making contexts.

The study highlights the importance of reproducibility and scalability in environmental modeling. The integration of collaborative platforms such as Google Colab, containerization through Docker, and monitoring frameworks like MLflow and TensorBoard provides a transferable workflow for future projects.

Future work should expand the input space to include meteorological variables (temperature, humidity, atmospheric pressure), test hybrid architectures (CNN–LSTM, attention mechanisms), and evaluate forecasts at longer horizons (24–72 hours). Recent advances show that hybrid designs integrating CNNs, LSTMs, and attention can substantially reduce forecasting errors and improve robustness compared to standalone models (Zhang et al., 2023; Liang et al., 2025; Lv et al., 2024). Incorporating these strategies will be essential for improving predictive accuracy and moving from experimental prototypes toward practical systems for air quality management.

In summary, while the results remain preliminary, this research contributes an initial step toward data-driven air quality forecasting in Bogotá. It offers both methodological lessons and a pathway for developing more reliable, interpretable, and impactful predictive systems that can ultimately support public health strategies, regulatory compliance, and broader sustainability goals, including Sustainable Development Goal 13 (Climate Action).

6.1. Managerial Implications

For policymakers and environmental authorities, this research illustrates how machine learning can eventually support proactive air quality management in Bogotá. Even at a proof-of-concept stage, the framework points to the possibility of using LSTM-based models to anticipate pollution episodes and align responses with Resolution 2254 of 2017. If refined and validated, such tools could guide contingency measures, improve communication through Air Quality Index platforms, and foster greater public awareness. The study, therefore, highlights a pathway for integrating predictive analytics into regulatory practice while recognizing that further development is necessary before operational adoption.

6.2. Theorical Implications

From an academic standpoint, this work contributes to the literature on deep learning for environmental forecasting by adapting a multi-output LSTM to simultaneously predict $PM_{2.5}$ and PM_{10}. This approach underscores the value of shared representations in time-series modeling and demonstrates how recurrent architectures can be tailored to complex urban datasets. Beyond the model itself, the use of reproducible tools such as TensorFlow, Docker, and MLflow emphasizes the importance of transparent, scalable workflows in applied machine learning research. These insights open avenues for future studies exploring hybrid neural architectures and the integration of meteorological and regulatory data, advancing both machine learning methodology and environmental science.

References:

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... Kudlur, M. (2016). TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)* (pp.265–283). USENIX Association. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

Alcaldía Mayor de Bogotá. (2017). Resolución 2254 de 2017 Ministerio del Medio Ambiente. SISJUR. Retrieved from https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=82634

Área Metropolitana de Bogotá. (2025). Sistema de Monitoreo de la Calidad del Aire de Bogotá (RMCAB). Retrieved from http://rmcab.ambientebogota.gov.co/dynamicTabulars/index

Casallas García, A., Ferro, C., & Celis Mayorga, N. (2021). Long short-term memory artificial neural network approach to forecast meteorology and $PM_{2.5}$ local variables in Bogotá, Colombia. Modeling Earth Systems and Environment, 8(3), 2951–2964. https://doi.org/10.1007/s40808-021-01274-6

Chollet, F. (2015). Keras [Computer software]. GitHub. https://github.com/fchollet/keras

Docker Inc. (2020). What is Docker? Retrieved from https://docs.docker.com/get-started/docker-overview/

Franceschi, F., Cobo, M., & Figueredo, M. (2018). Discovering relationships and forecasting PM_{10} and $PM_{2.5}$ concentrations in Bogotá, Colombia, using artificial neural networks, principal component analysis, and k-means clustering. Atmospheric Pollution Research, 9(5), 912–922. https://doi.org/10.1016/j.apr.2018.02.006

Función Pública de Colombia. (2015). Decreto 1076 de 2015: Sector Ambiente y Desarrollo Sostenible. Gestor Normativo. Retrieved from https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=78153

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research, 3*, 115–143. https://www.jmlr.org/papers/volume3/gers02a/gers02a.pdf

Google. (2025). Welcome to Colab. Colaboratory. Retrieved from https://colab.research.google.com/

Google Research. (2025). Colaboratory FAQ. Retrieved from https://research.google.com/colaboratory/faq.html

Graves, A., Fernández, S., Gómez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning (ICML '06) (pp. 369–376). ACM. https://doi.org/10.1145/1143844.1143891

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM). (2021). *Proporción de datos del Índice de Calidad del Aire (ICA), Bogotá* [Informe técnico]. http://archivo.ideam.gov.co/documents/11769/641368/2.01+HM+%C3%ADndice+calidad+aire.pdf/5130ffb3-a1bf-4d23-a663-b4c51327cc05

McKinney, W. (2010). Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference (pp. 51–56). SciPy. https://doi.org/10.25080/Majora-92bf1922-00a

Ministerio de Ambiente y Desarrollo Sostenible. (2017). Resolución 2254 de 2017. Retrieved from https://www.minambiente.gov.co/wp-content/uploads/2021/10/Resolucion-2254-de-2017.pdf

Ministerio de Ambiente y Desarrollo Sostenible. (2025). Contaminación atmosférica. Retrieved from https://www.minambiente.gov.co/asuntos-ambientales-sectorial-y-urbana/contaminacion-atmosferica/

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32). Curran Associates, Inc. https://papers.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In S. Dasgupta & D. McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning* (Vol. 28, pp. 1310–1318). PMLR. https://proceedings.mlr.press/v28/pascanu13.html

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. https://jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

U.S. Environmental Protection Agency. (2023). Particulate matter (PM) basics. U.S. Environmental Protection Agency. Retrieved September 16, 2025, from https://www.epa.gov/pm-pollution/particulate-matter-pm-basics

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. Computing in Science & Engineering, 13(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

World Health Organization. (2021). WHO global air quality guidelines: Particulate matter ($PM_{2.5}$ and PM_{10}), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. Geneva: WHO. Retrieved from https://apps.who.int/iris/handle/10665/345329

Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, M., Konwinski, A., & Xin, R. (2020). Accelerating the machine learning lifecycle with MLflow. IEEE Data Engineering Bulletin, 43(3), 34–45. Retrieved from https://people.eecs.berkeley.edu/~matei/papers/2018/ieee_mlflow.pdf

Duan, J., Gong, Y., Luo, J., & Zhao, Z. (2023). Air-quality prediction based on the ARIMA–CNN–LSTM combination model optimized by dung beetle optimizer. Scientific Reports, 13(1), 12127. https://doi.org/10.1038/s41598-023-36620-4

Sreenivasulu, T., & Mokesh Rayalu, G. (2025). Accurate hourly AQI prediction using temporal CNN–LSTM–MHA+GRU: A case study of seasonal variations and pollution extremes in Visakhapatnam, India. Results in Engineering, 27, 106303. https://doi.org/10.1016/j.rineng.2025.106303

Nguyen, A. T., Pham, D. H., Oo, B. L., Ahn, Y., & Lim, B. T. H. (2024). Predicting air quality index using attention hybrid deep learning and quantum-inspired particle swarm optimization. Journal of Big Data, 11, Article 71. https://doi.org/10.1186/s40537-024-00926-5