POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Geoinformatics-Based Environmental Data Fusion and Machine Learning for Surface-Level Air Pollution Estimation: A European-Scale Framework with a Milan Case Study

Tesi di Laurea Magistrale in
Geoinformatics Engineering - Ingegneria Geoinformatica

Author: **Claudia Isabela Saud-Miño**

Student ID: 10895430
Advisor: Prof. Vasil Yordanov
Academic Year: 2025-2026

Abstract

Air pollution monitoring remains constrained by the sparse spatial distribution of ground-based monitoring stations and by the challenge of translating satellite-derived atmospheric observations into accurate estimates of near-surface pollutant concentrations. This thesis presents a scalable geoinformatics and environmental machine learning framework for the integration of heterogeneous atmospheric datasets, combining European Environment Agency (EEA) ground measurements, ERA5 meteorological reanalysis products, Sentinel-5P TROPOMI atmospheric observations, and Copernicus Atmosphere Monitoring Service (CAMS) variables.

The proposed framework was developed at the European scale to support harmonized metadata reconstruction, pollutant-specific filtering, spatial co-location, temporal synchronization, and reproducible environmental data fusion. The predictive capabilities of the framework were subsequently evaluated through a localized case study in the Milan Metropolitan Area, focusing on daily surface-level SO_2 estimation. Multiple modeling paradigms were investigated, including classical machine learning algorithms, ensemble learning approaches, AutoML forecasting systems, and Time Series Foundation Models (TSFMs), such as Chronos and TimesFM.

The results demonstrate that forecasting based exclusively on historical SO_2 observations is insufficient for accurate prediction, confirming that surface-level SO_2 variability is strongly influenced by meteorological forcing, atmospheric transport processes, boundary-layer dynamics, and regional background atmospheric conditions. Classical machine learning performance improved progressively through enhanced feature engineering, chronological validation strategies, autoregressive lag structures, rolling-window statistics, and cyclical temporal encodings. The highest predictive performance was achieved by Chronos-2 using environmental covariates and Low-Rank Adaptation (LoRA) fine-tuning, highlighting the potential of pretrained temporal foundation models when adapted to atmospheric forecasting applications.

The final covariate-aware forecasting results should be interpreted as a retrospective upper-bound performance scenario because observed meteorological and atmospheric vari-

ables were available throughout the evaluation period. Operational deployment would require the integration of forecasted covariates obtained from numerical weather prediction and atmospheric composition models. Overall, the findings demonstrate that scalable air-quality estimation benefits substantially from the integration of ground observations, Earth Observation products, meteorological reanalysis, physically informed feature engineering, and advanced temporal forecasting architectures.

Keywords: environmental machine learning, air pollution, SO_2, Sentinel-5P, ERA5, Time Series Foundation Models, Chronos.

Abstract in lingua italiana

Il monitoraggio dell'inquinamento atmosferico è ancora limitato dalla distribuzione spazialmente disomogenea delle stazioni di monitoraggio a terra e dalla difficoltà di trasformare le osservazioni satellitari dell'atmosfera in stime affidabili delle concentrazioni di inquinanti al livello del suolo. Questa tesi propone un framework scalabile di geoinformatica e machine learning ambientale per l'integrazione di dataset atmosferici eterogenei, combinando misure a terra dell'European Environment Agency (EEA), prodotti di rianalisi meteorologica ERA5, osservazioni satellitari Sentinel-5P TROPOMI e variabili del Copernicus Atmosphere Monitoring Service (CAMS).

Il framework è stato sviluppato a scala europea per supportare la ricostruzione armonizzata dei metadati, il filtraggio degli inquinanti, la co-localizzazione spaziale, la sincronizzazione temporale e la fusione riproducibile di dati ambientali provenienti da fonti indipendenti. Le sue capacità predittive sono state successivamente valutate attraverso un caso di studio nell'Area Metropolitana di Milano, focalizzato sulla stima giornaliera delle concentrazioni superficiali di SO_2. Sono stati confrontati diversi paradigmi di modellazione, tra cui algoritmi classici di machine learning, approcci ensemble, configurazioni AutoML e Time Series Foundation Models (TSFM), quali Chronos e TimesFM.

I risultati dimostrano che la previsione basata esclusivamente sulla serie storica di SO_2 non è sufficiente per descrivere accuratamente la variabilità delle concentrazioni al suolo. Le prestazioni ottenute confermano infatti che la dinamica della SO_2 è fortemente influenzata dalle condizioni meteorologiche, dai processi di trasporto atmosferico, dalla variabilità dello strato limite planetario e dalle condizioni atmosferiche di fondo a scala regionale. Le prestazioni dei modelli classici di machine learning sono migliorate progressivamente grazie all'introduzione di tecniche avanzate di feature engineering, validazione cronologica, variabili lag, statistiche mobili e codifiche cicliche delle componenti temporali. Le migliori prestazioni predittive sono state ottenute dal modello Chronos-2 mediante l'integrazione di covariate ambientali e l'applicazione di strategie di fine-tuning basate su Low-Rank Adaptation (LoRA), evidenziando il potenziale dei modelli fondazionali temporali preaddestrati quando adattati a problemi di previsione della qualità dell'aria.

I risultati finali ottenuti mediante l'utilizzo delle covariate ambientali devono tuttavia essere interpretati come uno scenario retrospettivo di limite superiore, poiché durante la fase di valutazione erano disponibili osservazioni meteorologiche e atmosferiche reali. Un'applicazione operativa del framework richiederebbe invece l'impiego di covariate previste da modelli numerici di previsione meteorologica e da sistemi di previsione della composizione atmosferica. Nel complesso, questa ricerca dimostra come la stima scalabile della qualità dell'aria possa beneficiare significativamente dell'integrazione tra osservazioni a terra, prodotti di Earth Observation, dati di rianalisi meteorologica, tecniche di feature engineering fisicamente motivate e avanzate architetture di modellazione temporale.

Parole chiave: machine learning ambientale, inquinamento atmosferico, SO_2, Sentinel-5P, ERA5, Time Series Foundation Models, Chronos.

Contents

1 | Introduction

1.1. Air Pollution Context

Air pollution has become one of the most critical environmental and public health challenges worldwide, particularly within rapidly urbanizing and industrialized regions. The continuous growth of urban populations, industrial activity, transportation systems, and energy consumption has significantly increased atmospheric emissions of harmful pollutants. According to the World Health Organization (WHO) [38], air pollution is directly associated with millions of premature deaths annually and is considered one of the leading environmental risk factors affecting human health. Pollutants such as sulfur dioxide (SO_2), nitrogen dioxide (NO_2), particulate matter ($PM_{2.5}$ and PM_{10}), ozone (O_3), and carbon monoxide (CO) have been strongly linked to respiratory diseases, cardiovascular disorders, neurological impairments, and increased mortality rates.

Similarly, urban environments are particularly vulnerable to air pollution due to the concentration of anthropogenic emission sources within relatively small geographic areas. Traffic congestion, industrial facilities, residential heating systems, and energy production contribute substantially to the accumulation of atmospheric pollutants in cities. Furthermore, urban morphology and meteorological conditions can intensify pollutant accumulation through reduced atmospheric dispersion and the formation of urban heat islands. As a result, densely populated metropolitan regions often experience pollutant concentrations that exceed international air quality guidelines established by organizations such as the WHO and the European Environment Agency (EEA) [10].

Among atmospheric pollutants, SO_2 and NO_2 are especially important due to their environmental relevance, direct health impacts, and association with combustion-related activities. SO_2 is mainly emitted from fossil fuel combustion, industrial processes, shipping activities, and power generation facilities. Exposure to elevated SO_2 concentrations can irritate the respiratory system, aggravate asthma, and contribute to the formation of secondary particulate matter and acid rain. NO_2, on the other hand, is strongly associated with traffic emissions and urban combustion processes. Long-term exposure to

NO_2 has been linked to reduced lung function, chronic respiratory diseases, and increased susceptibility to infections. Additionally, NO_2 plays a major role in atmospheric photochemical reactions that contribute to ozone formation and secondary aerosol production [10, 38].

Despite the importance of air quality monitoring, the characterization of atmospheric pollution remains limited by the spatial sparsity of conventional monitoring networks. Ground-based monitoring stations provide highly accurate and temporally continuous pollutant measurements; however, their installation, operation, and maintenance are economically expensive. Consequently, monitoring infrastructure is usually concentrated in major urban centers, while suburban, rural, and transitional industrial regions remain insufficiently monitored. This spatial heterogeneity restricts the availability of fully characterizing pollutant variability across large geographic domains and limits the representativeness of exposure assessments.

To partially overcome these limitations, satellite remote sensing has emerged as an important complementary source of atmospheric information. In Europe, the Copernicus Sentinel-5 Precursor (Sentinel-5P) mission, carrying the TROPOspheric Monitoring Instrument (TROPOMI), provides daily observations of atmospheric trace gases at continental and global scales. Sentinel-5P represents an important step toward a new generation of atmospheric monitoring missions and serves as a precursor to future Copernicus atmospheric observation systems, including geostationary platforms capable of providing higher temporal resolution measurements.

Despite their broad spatial coverage, satellite observations do not directly measure near-surface pollutant concentrations. Instead, products such as TROPOMI retrieve atmospheric column densities, introducing additional challenges when estimating ground-level air quality and human exposure. Consequently, integrating satellite observations with meteorological reanalysis products and ground-based monitoring networks through geoinformatics and machine learning methodologies has become an increasingly important research direction for improving spatially continuous air-quality estimation, atmospheric monitoring, and environmental intelligence systems.[34].

1.2. Earth Observation and Environmental Machine Learning

Recent advances in Earth Observation (EO) technologies have significantly transformed the monitoring and analysis of atmospheric pollution at regional and global scales. Satel-

lite remote sensing platforms provide spatially continuous observations that complement traditional ground-based monitoring networks, enabling the characterization of atmospheric dynamics across areas where in-situ measurements are sparse or unavailable. Among the most important recent missions for atmospheric composition monitoring is the Sentinel-5P satellite, developed by the European Space Agency (ESA) under the Copernicus Programme. Sentinel-5P carries the TROPOMI, which provides daily observations of atmospheric trace gases, including SO_2, NO_2, CO, methane (CH_4), and O_3, at high spatial resolution and near-global coverage [34]. These observations have become fundamental for atmospheric studies due to their capability to monitor pollutant transport, industrial emissions, biomass burning events, and urban air quality patterns over large geographic domains.

However, satellite observations cannot be interpreted as direct measurements of near-surface pollutant concentrations. Instruments such as TROPOMI retrieve vertically integrated atmospheric column densities, which quantify the total amount of a trace gas within the atmospheric column rather than its concentration at a specific height above the surface. As a result, the satellite-derived signal represents an integrated atmospheric burden over the sensor footprint and vertical column, whereas air-quality assessment and exposure analysis require ground-level concentrations representative of the breathing zone. This vertical representativeness gap introduces the need for additional modeling procedures to relate columnar satellite information to surface-level pollutant variability.

The relationship between atmospheric column measurements and surface concentrations is influenced by several atmospheric processes, including boundary-layer height, vertical mixing, atmospheric stability, emission source distribution, and pollutant transport. As illustrated in Figure 1.1, two locations may exhibit similar satellite-derived column concentrations while presenting substantially different ground-level concentrations due to differences in meteorological and atmospheric conditions. Therefore, satellite observations alone are generally insufficient for accurately estimating human exposure and near-surface air quality.

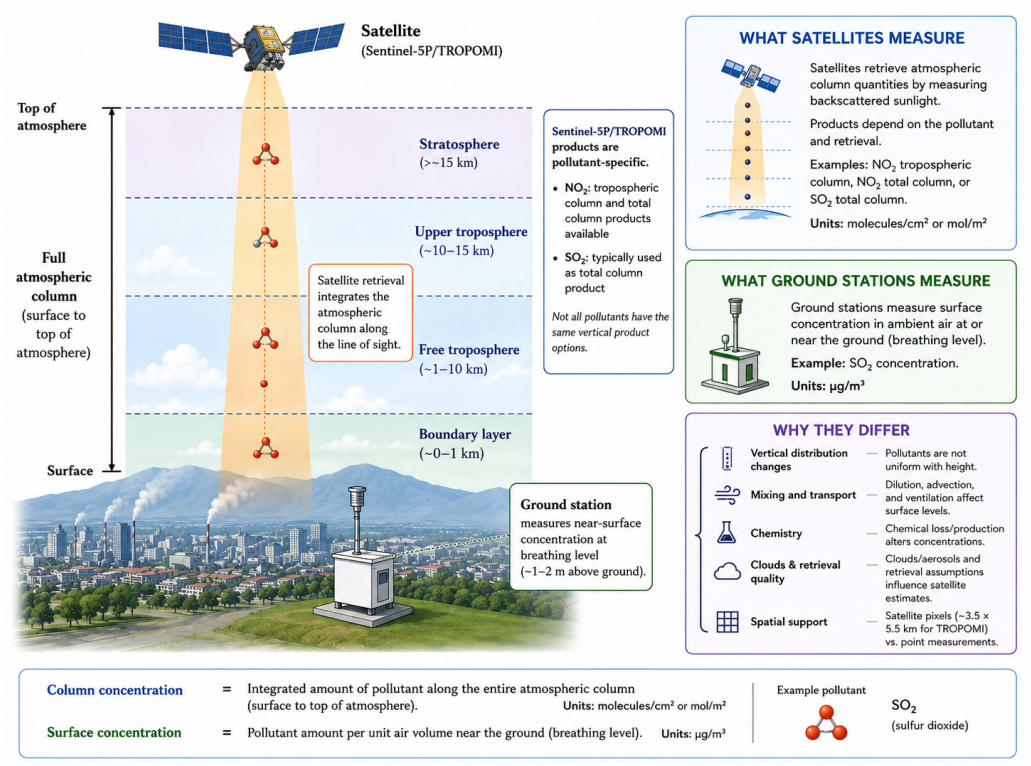

Figure 1.1: Conceptual difference between satellite-derived atmospheric column measurements and surface-level pollutant concentrations. Satellite instruments observe the integrated pollutant amount throughout the atmospheric column, whereas air-quality monitoring stations measure concentrations directly at ground level. The relationship between both quantities is influenced by boundary-layer dynamics, vertical mixing, atmospheric transport, and meteorological conditions.

Consequently, atmospheric reanalysis products have become essential complementary datasets for environmental analysis. Reanalysis systems combine numerical weather prediction models with large volumes of observational data through data assimilation techniques in order to reconstruct physically consistent atmospheric states over time. Among the most widely used products is ERA5, developed by the European Centre for Medium-Range Weather Forecasts (ECMWF). ERA5 provides hourly global meteorological variables such as temperature, wind speed, precipitation, surface pressure, and boundary-layer height, which are highly relevant for understanding pollutant transport, dispersion, atmospheric stability, and accumulation processes [15]. The integration of satellite observations with meteorological reanalysis products enables a more comprehensive representation of atmospheric dynamics and pollutant behavior, while also helping bridge the gap between atmospheric column observations and near-surface pollutant concentrations.

Simultaneously, the increasing availability of large-scale environmental datasets has accelerated the application of Artificial Intelligence (AI) and machine learning methodologies within atmospheric science. Traditional deterministic atmospheric models are computationally expensive and often require detailed physical parameterizations, while machine learning approaches can learn complex non-linear relationships directly from data. Algorithms such as Random Forest (RF), Gradient Boosting (GTB), Support Vector Regression (SVR), deep neural networks, and transformer-based Time Series Foundation Models (TSFMs) have demonstrated strong capabilities for atmospheric pollutant estimation and forecasting [2, 25]. These models are capable of integrating heterogeneous datasets, including satellite observations, meteorological reanalysis variables, temporal descriptors, and ground-based measurements, in order to estimate surface pollutant concentrations with improved spatial and temporal resolution. Consequently, environmental machine learning has emerged as a powerful interdisciplinary field that combines geoinformatics, remote sensing, atmospheric science, and artificial intelligence for scalable environmental monitoring and air quality assessment.

1.3. Research Gap

Despite the substantial progress achieved in atmospheric monitoring and environmental machine learning, several important methodological and operational limitations remain unsolved within the current literature. One of the main challenges is the spatial sparsity of ground-based air quality monitoring stations. Although monitoring networks operated by environmental agencies provide highly accurate near-surface pollutant measurements, their spatial distribution remains heterogeneous and strongly concentrated around major urban centers and densely populated regions. Consequently, large suburban, rural, mountainous, and industrial transition areas remain insufficiently monitored, limiting the capability to fully characterize pollutant variability and human exposure across broad geographic domains [10].

This sparse monitoring infrastructure directly contributes to weak spatial coverage within conventional atmospheric analyses. Pollutant concentrations exhibit strong spatial heterogeneity due to local emission sources, meteorological conditions, topography, and land-use characteristics. However, relying exclusively on station-based observations introduces substantial uncertainty when attempting to interpolate pollutant distributions over regions lacking monitoring infrastructure. Although satellite remote sensing platforms such as Sentinel-5P partially address this limitation through spatially continuous observations, satellite products measure atmospheric column concentrations rather than direct surface-

level pollutant exposure. As a result, important challenges remain regarding the spatial harmonization and integration of heterogeneous atmospheric datasets [34].

At the same time, many traditional atmospheric modeling approaches continue to present methodological limitations when dealing with the complexity of environmental systems. Classical statistical models and deterministic chemical transport models frequently struggle to capture the highly non-linear interactions between meteorological conditions, pollutant emissions, atmospheric chemistry, and temporal variability. Furthermore, several machine learning studies still rely on simplified experimental configurations, geographically limited datasets, or random train-test partitioning strategies that introduce temporal leakage into evaluation procedures. Because atmospheric observations are strongly autocorrelated over time, random data splitting can lead to unrealistically optimistic predictive performance estimates that do not reflect real operational forecasting conditions [37].

Another important limitation within current atmospheric machine learning research concerns the reproducibility and scalability of environmental data engineering workflows. While numerous studies have demonstrated the predictive potential of machine learning for air-quality estimation, the associated data preparation procedures are often described only briefly and are highly tailored to specific study areas, pollutants, or datasets. As a result, challenges related to environmental data harmonization, asynchronous data retrieval, metadata inconsistencies, temporal synchronization, spatial integration, and workflow reproducibility frequently receive less attention than model development itself.

Furthermore, relatively few studies explicitly address the integration of heterogeneous environmental datasets, such as EEA ground-based observations, ERA5 meteorological reanalysis products, and Sentinel-5P atmospheric observations, within a unified framework that can be systematically applied across different European regions [23]. Consequently, a research gap remains in the development of reproducible geoinformatics-based frameworks capable of supporting scalable environmental machine learning and atmospheric pollutant estimation at continental scale.

1.4. Research Objectives

The primary objective of this research is to develop a scalable and reproducible geoinformatics and machine learning framework for the integration of heterogeneous environmental datasets. The framework is designed for European-scale atmospheric data harmonization, while the predictive modeling component is evaluated through a localized proof-of-concept case study focused on daily surface-level SO_2 estimation in the Milan Metropolitan Area.

The specific objectives of this thesis are:

- To develop a reproducible environmental data engineering framework capable of harmonizing, integrating, and managing heterogeneous atmospheric datasets from the European Environment Agency (EEA), ERA5 meteorological reanalysis, and Sentinel-5P Earth Observation products across European monitoring domains.
- To implement a multi-source environmental data fusion workflow that supports spatial and temporal harmonization, feature generation, and environmental analysis for atmospheric monitoring applications.
- To evaluate the capability of machine learning and Time Series Foundation Models to estimate daily surface-level SO_2 concentrations in the Milan Metropolitan Area using integrated environmental observations, atmospheric variables, and engineered temporal features.
- To assess the influence of feature engineering, environmental covariates, validation strategies, and model adaptation techniques on forecasting performance, while identifying methodological requirements for scalable atmospheric prediction systems.

This research investigates whether the integration of Earth Observation data and meteorological reanalysis variables can improve the estimation of near-surface air pollutant concentrations over heterogeneous European environments. In particular, the study evaluates the extent to which Sentinel-5P atmospheric column observations can be used to infer ground-level pollutant concentrations when combined with meteorological and temporal predictors. The thesis further examines which environmental variables contribute most significantly to predictive performance, with special attention given to feature engineering strategies, autoregressive lag structures, and temporal validation methodologies designed to prevent information leakage. Additionally, the research compares the performance of conventional machine learning approaches with recent Time Series Foundation Models (TSFMs), including Chronos and TimesFM, in order to determine whether pretrained temporal architectures can outperform classical supervised learning methods for atmospheric forecasting tasks. Finally, this work investigates whether a reproducible geoinformatics framework can effectively harmonize heterogeneous environmental datasets at a European scale while addressing the limitations associated with sparse monitoring networks and uneven spatial coverage within air quality monitoring systems.

The remainder of this thesis is organized as follows. Chapter 2 reviews the state of the art related to satellite–ground data fusion, column-to-surface pollutant estimation, geoinformatics challenges in multi-source atmospheric data integration, meteorological covari-

ates, machine learning approaches for air-quality estimation, and Time Series Foundation Models. Chapter 3 presents the proposed methodology, including the study domains, environmental data acquisition, spatial and temporal harmonization, feature engineering, feature selection, model development, training strategy, evaluation metrics, and implementation setup. Chapter 4 presents the experimental results obtained from dataset exploration, feature-importance analysis, classical machine learning models, ensemble approaches, AutoML configurations, and Time Series Foundation Models. Chapter 5 discusses the scientific interpretation of the results, their relationship with previous studies, implications for environmental monitoring, limitations, and future work. Finally, Chapter 6 summarizes the principal conclusions and contributions of this research.

2 State of the Art

2.1. Satellite–Ground Data Fusion for Surface-Level Air Quality Estimation

Ground-based monitoring stations remain the reference source for regulatory air-quality assessment because they provide direct near-surface pollutant measurements with high temporal continuity. However, previous studies consistently identify the sparse and uneven spatial distribution of monitoring networks as a major limitation for exposure assessment and large-scale air-quality characterization [4, 24]. Monitoring stations are commonly concentrated in urban and economically developed regions, while suburban, rural, mountainous, and industrial transition areas are often less densely represented. This spatial imbalance limits the capacity of station-only approaches to describe pollutant variability continuously across space [16, 20].

Satellite Earth Observation has therefore been widely explored as a complementary source of atmospheric information. Unlike point-based monitoring stations, satellite products provide spatially continuous observations over regional, continental, and global domains, supporting the analysis of pollutant transport, emission hotspots, transboundary pollution, wildfire plumes, volcanic emissions, and areas where in-situ infrastructure is limited [12, 34]. Nevertheless, the literature also emphasizes that satellite observations cannot be interpreted as direct replacements for ground-based measurements because they differ from monitoring stations in spatial support, temporal sampling, and atmospheric representativeness. The specific problem of translating satellite column observations into surface-level concentrations is discussed in the following section [3, 23, 31].

For this reason, recent air-quality studies increasingly adopt satellite–ground data fusion strategies. In these approaches, satellite products provide spatially continuous atmospheric information, while ground-based stations provide the surface-level reference required for calibration and validation [23, 34]. Machine learning models are then used to learn empirical relationships between satellite observations, meteorological variables, land-use descriptors, temporal features, and measured pollutant concentrations [5, 6, 11].

These studies show that the integration of Earth Observation data with in-situ measurements can improve pollutant estimation compared with approaches based on a single data source, particularly when meteorological and temporal covariates are included.

However, the effectiveness of satellite–ground fusion depends strongly on how spatial and temporal inconsistencies are handled. A monitoring station represents a point-based measurement influenced by local emissions, street configuration, surrounding land use, and station typology [26, 31]. In contrast, a satellite pixel represents an areal atmospheric column over a much larger spatial support [20]. Reanalysis products introduce an additional scale because they are provided on regular meteorological grids that may be coarser than both station locations and satellite pixels. Therefore, the relationship between these datasets is not purely statistical, but also geoinformatics-dependent: co-location strategy, spatial aggregation, temporal synchronization, quality filtering, and station representativeness directly affect model reliability [3, 6, 16].

Several studies address these issues through quality-assurance filtering, spatial aggregation around monitoring sites, temporal averaging, and the integration of meteorological reanalysis variables [6, 22]. Nevertheless, many workflows remain tailored to specific cities, pollutants, or experimental periods, which limits their portability to broader geographic domains. In addition, methodological choices such as the spatial matching radius, aggregation period, treatment of missing satellite pixels, and temporal alignment between satellite overpass time and ground observations are not always reported in sufficient detail. This reduces reproducibility and complicates direct comparison between studies [6, 16, 23].

Consequently, the current literature indicates that satellite–ground fusion is a promising direction for surface-level air-quality estimation, but also shows that predictive performance depends on transparent and reproducible geoinformatics procedures [34]. The challenge is not only to combine satellite and station data, but also to harmonize heterogeneous environmental datasets with different spatial supports, temporal frequencies, units, uncertainties, and representativeness [22, 23]. This motivates the need for scalable data-fusion workflows capable of integrating ground observations, Earth Observation products, meteorological reanalysis, and machine learning under explicit spatial and temporal harmonization rules, as further discussed in the following sections [6, 16].

2.2. Column-to-Surface Estimation Using Satellite Atmospheric Observations

A central challenge in satellite-based air-quality research is the estimation of near-surface pollutant concentrations from atmospheric column observations. Sentinel-5P TROPOMI and related atmospheric satellite missions provide valuable information on the vertical abundance of trace gases, but regulatory air-quality assessment and exposure analysis require concentrations close to the ground [3, 23]. Therefore, the main methodological issue is not the availability of satellite observations themselves, but the conversion of column-integrated atmospheric information into surface-level estimates [20].

Previous studies have addressed this problem by combining satellite retrievals with ground-based monitoring data and auxiliary environmental predictors. In these data-fusion approaches, the satellite product contributes spatially continuous atmospheric information, while the ground station measurements provide the surface-level reference needed for calibration. Machine learning models are then used to estimate empirical relationships between satellite columns, meteorological conditions, temporal patterns, and measured pollutant concentrations [5, 6, 11]. This strategy has been particularly relevant for pollutants such as NO_2, CO, and SO_2, for which satellite products can describe regional atmospheric variability but cannot directly represent human exposure at ground level.

The strength of column-to-surface modeling depends on the degree to which the selected predictors describe the physical processes connecting atmospheric columns with surface concentrations [3, 6]. Meteorological variables are especially important because vertical mixing, wind transport, precipitation, atmospheric stability, and boundary-layer dynamics influence whether pollutants remain near the surface or are dispersed through the atmospheric column. For this reason, several studies integrate satellite observations with meteorological or reanalysis products in order to reduce the mismatch between column measurements and ground-level concentrations [6, 11, 15]. The inclusion of such covariates is not only a statistical improvement, but also a physically motivated strategy for representing atmospheric processes that satellite columns alone cannot resolve.

However, the literature also shows that column-to-surface estimation remains affected by several sources of uncertainty. First, the spatial support of a satellite pixel does not correspond to the local representativeness of a monitoring station [11, 16]. A ground station may be strongly influenced by nearby roads, industrial facilities, urban morphology, or station type, whereas a satellite pixel averages atmospheric information over a broader area. Second, satellite retrievals are affected by missing data and quality degradation caused

by cloud cover, aerosols, surface reflectance, snow or ice, and low solar illumination conditions [3, 34]. Third, the temporal sampling of polar-orbiting satellites provides only a limited representation of daily pollutant variability, while ground stations record continuous or hourly concentrations. These inconsistencies require explicit spatial co-location, temporal aggregation, and quality-filtering procedures before satellite observations can be used in predictive models [6, 22].

Several studies apply quality-assurance filtering, pixel aggregation, station-buffer matching, daily averaging, or temporal alignment with satellite overpass times to reduce these uncertainties [6, 22]. Nevertheless, the implementation of these procedures varies substantially across studies. Some works prioritize spatial matching between satellite pixels and stations, while others emphasize temporal aggregation, pollutant-specific filtering, or feature engineering. As a result, reported model performance is often difficult to compare directly because differences in preprocessing and harmonization can influence the predictive task as much as the selected algorithm [3, 34, 37].

This limitation is particularly relevant for geoinformatics-based air-quality modeling. Column-to-surface estimation is not only a machine learning problem, but also a spatial and temporal data harmonization problem [3, 22]. The reliability of the final model depends on how heterogeneous datasets with different units, spatial resolutions, temporal frequencies, and uncertainty structures are aligned [20]. Therefore, the literature supports the use of Sentinel-5P as a complementary atmospheric predictor, but also indicates that its contribution to surface-level pollutant estimation depends on transparent geoinformatics procedures and integration with ground observations and meteorological covariates [6, 11, 16].

2.3. Geoinformatics Challenges in Multi-Source Atmospheric Data Integration

The integration of ground-based monitoring data, satellite atmospheric products, and meteorological reanalysis variables requires more than the simple combination of environmental predictors. These datasets are generated under different spatial supports, temporal sampling schemes, coordinate structures, measurement units, metadata conventions, and uncertainty conditions. From a geoinformatics perspective, these differences define the structure of the modeling problem before any machine learning algorithm is applied [6, 16, 23].

A central challenge is spatial scale mismatch. Ground stations provide point-based mea-

surements that reflect local environmental conditions and may be strongly influenced by nearby roads, industrial sources, urban morphology, or station classification. Satellite retrievals represent areal atmospheric columns over pixel footprints that are substantially larger than a monitoring site, while reanalysis products describe meteorological fields on regular grids. The predictive relationship between these sources therefore depends on how point, pixel, and grid-based observations are spatially co-located and aggregated. Inadequate spatial matching can introduce representativeness errors, especially when local surface emissions are compared with broader satellite or reanalysis fields [3, 11, 20].

Temporal synchronization introduces an additional source of uncertainty. Ground monitoring networks often provide hourly or sub-hourly observations, satellite instruments acquire measurements at specific overpass times, and reanalysis products are available at regular temporal intervals. When these data sources are aggregated to daily values, methodological choices regarding averaging windows, overpass alignment, missing observations, and lag construction can affect both the predictor variables and the target concentration. These decisions are particularly relevant for pollutants with strong diurnal cycles, episodic emissions, or rapid meteorological responses [6, 22, 37].

Station representativeness is also critical in satellite–ground fusion studies. Traffic, industrial, urban background, suburban, and rural stations do not describe the same atmospheric processes, even when they measure the same pollutant. A model trained using heterogeneous station types may therefore combine observations influenced by different emission regimes and spatial footprints. Previous studies often account for this issue through station classification, spatial filtering, land-use descriptors, or localized calibration, but the treatment of station representativeness remains inconsistent across the literature [16, 26, 31].

Data harmonization is consequently a core methodological requirement in environmental machine learning. Coordinate reference systems, pollutant units, temporal indexes, missing-data structures, quality flags, aggregation rules, and metadata conventions must be standardized before multi-source modeling can be performed. These operations directly influence the final analytical dataset and should therefore be considered part of the scientific methodology rather than secondary preprocessing. For scalable air-quality estimation, reproducible geoinformatics workflows are necessary to ensure that satellite, station, and reanalysis datasets are integrated under transparent and transferable rules [6, 16, 23].

2.4. Meteorological and Reanalysis Covariates in Pollutant Estimation

Meteorological conditions are widely recognized in the literature as key drivers of surface-level pollutant variability. Even when emissions remain relatively stable, changes in atmospheric circulation, vertical mixing, precipitation, temperature, and boundary-layer dynamics can produce substantial differences in measured pollutant concentrations [6, 17, 26]. For this reason, recent air-quality estimation studies increasingly integrate meteorological variables or reanalysis products as covariates in satellite–ground data-fusion and machine learning frameworks [11, 20].

The role of meteorological covariates is particularly important in surface-level estimation from satellite atmospheric observations. Satellite products describe vertically integrated atmospheric columns, whereas ground stations measure near-surface concentrations. The relationship between these two quantities depends strongly on atmospheric mixing and transport conditions. For example, wind speed and wind direction influence horizontal transport and dispersion; precipitation contributes to wet removal processes; temperature affects atmospheric stability and chemical transformation; and boundary layer height controls the vertical volume available for pollutant dilution or accumulation [17, 26]. Therefore, meteorological variables provide physically meaningful information that helps machine learning models interpret whether atmospheric pollutants are likely to remain near the surface or be distributed through a deeper atmospheric column [3, 11, 25].

Reanalysis datasets such as ERA5 are frequently used in this context because they provide gridded and temporally continuous meteorological fields over large geographic domains [15]. In contrast to local weather-station observations, reanalysis products offer standardized meteorological variables that can be integrated with satellite pixels and ground monitoring locations. This makes them especially useful for geoinformatics-based workflows that require consistent environmental covariates across multiple regions, pollutants, and temporal periods. Previous studies have shown that the inclusion of meteorological or reanalysis variables can improve pollutant estimation performance when combined with satellite observations and in-situ measurements [6, 11].

However, the integration of reanalysis covariates also introduces geoinformatics challenges. ERA5 and similar products are provided on regular spatial grids and hourly temporal resolutions, while ground stations are point-based and satellite retrievals have different pixel footprints and overpass times. Consequently, reanalysis variables must be spatially co-located, temporally aggregated, and synchronized with both satellite observations and

ground measurements before being used in predictive models. These operations are not neutral preprocessing steps: the choice of spatial interpolation method, temporal aggregation window, and alignment strategy can influence the resulting feature values and, therefore, model performance [11, 22, 23].

The literature also indicates that the importance of meteorological covariates may vary by pollutant, season, study area, and modeling objective. Pollutants dominated by local traffic emissions may respond differently to meteorological forcing than pollutants influenced by industrial emissions, regional transport, or atmospheric chemistry [17, 20, 36]. Similarly, variables such as wind speed, precipitation, and boundary layer height may become more relevant under stagnation episodes, wintertime accumulation, or transport-dominated conditions. This variability suggests that meteorological covariates should not be treated as generic auxiliary features, but as physically interpretable predictors whose contribution depends on atmospheric context [11, 26].

Overall, previous studies support the integration of meteorological and reanalysis variables in air-quality estimation, especially when satellite observations are used to infer surface-level concentrations. Nevertheless, many works provide limited detail on how these covariates are harmonized with ground and satellite datasets, and few explicitly discuss how spatial scale, temporal resolution, and interpolation choices affect downstream modeling. This remains an important limitation for reproducible geoinformatics-based environmental machine learning [17, 23, 26].

2.5. Machine Learning Approaches for Air Quality Estimation and Forecasting

Machine learning has become a central methodological approach in air-quality estimation because it can integrate heterogeneous environmental predictors and model nonlinear relationships between emissions, meteorology, satellite observations, temporal variability, and surface pollutant concentrations. Compared with deterministic atmospheric chemistry models, data-driven approaches require less explicit numerical representation of physical and chemical processes, since they learn empirical relationships from historical observations [16, 31, 37]. This has made machine learning especially attractive for satellite–ground data-fusion studies, where predictors may include ground monitoring data, atmospheric column observations, meteorological reanalysis variables, land-use descriptors, traffic-related indicators, and engineered temporal features [6, 11, 25].

Early and benchmark air-quality studies have commonly relied on regression-based ap-

proaches because of their interpretability, computational efficiency, and usefulness for identifying first-order relationships between predictors and pollutant concentrations. Linear and regularized regression models are often used as baseline methods against which more flexible models can be evaluated [16, 21, 22]. However, their capacity to represent atmospheric pollutant dynamics is limited by the non-linear interactions between emissions, meteorological forcing, atmospheric transport, boundary-layer evolution, wet deposition, and chemical transformation processes. For this reason, regression models are generally more useful as reference models than as final predictive systems in complex air-quality estimation tasks [11, 19, 26].

Tree-based ensemble methods have become some of the most frequently used models in environmental machine learning because they perform well on structured tabular datasets and can represent non-linear interactions without requiring strong distributional assumptions. Random Forest, XGBoost, LightGBM, and related gradient boosting methods have been widely applied to pollutant estimation and forecasting using combinations of ground observations, meteorological variables, satellite products, and temporal descriptors [14, 25, 32]. Comparative studies often report that tree-based models outperform simpler regression approaches, especially when the input space includes heterogeneous covariates and non-linear dependencies [33]. Their additional capacity to provide feature-importance measures has also made them useful for interpreting the relative contribution of meteorological, temporal, and satellite-derived predictors.

Nevertheless, the strong performance of tree-based models does not eliminate broader methodological limitations in air-quality studies. Reported performance can depend substantially on preprocessing choices, missing-data treatment, feature engineering, spatial aggregation, and train–test splitting strategy. In particular, models trained on pollutant time series may appear highly accurate when evaluated with random data splits, because temporal autocorrelation allows information from nearby future observations to leak into the training process. Therefore, model comparison is only meaningful when the same input data, feature engineering strategy, and leakage-aware validation protocol are applied consistently across algorithms [4, 6, 16].

Deep learning models have also been investigated for air-quality forecasting, especially when temporal dependencies or high-dimensional environmental inputs are involved. Multilayer perceptrons, recurrent neural networks, LSTM architectures, convolutional models, and Transformer-based approaches have been used to capture non-linear relationships, lagged pollutant responses, seasonal patterns, and long-range temporal dependencies [21, 23, 31]. These models can be advantageous when large training datasets are available and when pollutant dynamics are influenced by complex sequential patterns. However,

their performance is not automatically superior to tree-based models in smaller or tabular environmental datasets. Deep learning approaches may require larger data volumes, careful hyperparameter tuning, and stronger computational resources, while also reducing interpretability compared with simpler models [16, 24, 33].

The literature therefore suggests that no single model family is universally optimal for air-quality estimation. Regression models provide interpretable baselines, tree-based ensembles often perform strongly on structured environmental predictors, and deep learning methods may be useful when temporal or high-dimensional patterns dominate the forecasting task. However, many studies evaluate these models under different datasets, validation schemes, preprocessing workflows, or spatial domains, making direct comparison difficult. This limits the ability to determine whether performance gains are produced by the algorithm itself or by differences in data preparation, feature engineering, temporal splitting, and spatial harmonization [11, 16, 21, 37].

For this reason, recent environmental machine learning research increasingly emphasizes the need for standardized and reproducible benchmarking. In the context of surface-level pollutant estimation, model evaluation should consider not only predictive accuracy but also temporal generalization, leakage prevention, interpretability, data availability, and scalability across geographic domains. These issues are particularly important for geoinformatics-based workflows, where the modeling stage depends on the prior harmonization of ground stations, satellite observations, meteorological grids, and temporal features. A critical comparison of machine learning approaches therefore requires consistent input data, explicit spatial-temporal preprocessing, and chronological validation conditions [6, 22, 23, 25].

2.6. Time Series Foundation Models for Environmental Forecasting

Time Series Foundation Models (TSFMs) represent a recent direction in forecasting research, in which large pretrained models are designed to generalize across heterogeneous temporal datasets and downstream prediction tasks. Instead of training a forecasting model from scratch for each individual time series, TSFMs aim to transfer temporal representations learned during large-scale pretraining to new domains through zero-shot inference, covariate conditioning, or task-specific adaptation. Models such as Chronos and TimesFM have shown promising results in general forecasting benchmarks [8, 18], and recent atmospheric applications suggest potential for environmental time series.

In environmental forecasting, the potential value of TSFMs lies in their capacity to model temporal dependence, seasonal structure, persistence, and nonlinear sequential patterns. These properties are relevant for air-quality applications because pollutant concentrations often exhibit autocorrelation, weekly cycles, seasonal variability, and delayed responses to meteorological conditions [21, 25]. In principle, pretrained temporal models may reduce the need for extensive manual feature engineering and may provide stronger generalization when historical data are limited [8, 18]. This makes TSFMs an attractive candidate for atmospheric pollutant forecasting, especially when compared with conventional machine learning models that require explicitly engineered lag variables, rolling statistics, and temporal encodings.

However, the direct transfer of TSFMs to surface-level pollutant estimation remains methodologically uncertain. Atmospheric pollutant concentrations are not governed by temporal persistence alone. They are influenced by exogenous drivers such as wind transport, precipitation, boundary-layer dynamics, temperature, atmospheric stability, regional background concentrations, emission variability, and retrieval uncertainty in satellite products. Therefore, a model trained only on historical pollutant observations may fail to represent the environmental processes that determine short-term concentration changes. This limitation is particularly relevant for pollutants such as SO_2, whose surface-level variability may be strongly affected by episodic emissions, transport processes, and meteorological dispersion conditions [6, 23, 26, 36].

The literature on TSFMs in air-quality applications remains relatively limited compared with the broader literature on classical machine learning and deep learning for pollutant estimation. Existing TSFM studies and surveys mainly evaluate general forecasting benchmarks or broad time-series domains [8, 18], while only recent work has begun to explore atmospheric applications such as greenhouse-gas concentration forecasting. This emerging evidence suggests that TSFMs may be useful for environmental time series, particularly under zero-shot or fine-tuned configurations, but it does not yet resolve their suitability for surface-level air-quality estimation from heterogeneous satellite, meteorological, and ground-based observations. In particular, it remains unclear whether zero-shot forecasting based only on historical pollutant values can compete with conventional machine learning models that use meteorological, satellite, and engineered temporal predictors.

A further issue concerns evaluation comparability. TSFMs are often assessed using forecasting protocols that differ from the validation strategies used in tabular environmental machine learning studies. For a meaningful comparison, foundation models and conventional machine learning algorithms should be evaluated under consistent chronological splits, equivalent prediction horizons, comparable input information, and leakage-aware

validation procedures. Without this consistency, it is difficult to determine whether observed performance differences are caused by the model architecture, the use of covariates, the temporal validation design, or the amount of information available at prediction time [1, 37].

Recent approaches attempt to improve TSFM performance through the integration of covariates and parameter-efficient adaptation strategies. Covariate-aware forecasting allows environmental variables such as meteorology, atmospheric composition, and temporal descriptors to inform the prediction process, while fine-tuning techniques such as Low-Rank Adaptation (LoRA) enable model adaptation with lower computational cost than full retraining. These strategies are particularly relevant for atmospheric applications because they allow pretrained temporal models to incorporate domain-specific environmental information while preserving the benefits of large-scale pretraining [8, 18, 39].

Overall, TSFMs offer a promising but still insufficiently validated direction for environmental forecasting. Their usefulness for surface-level air-quality estimation depends on whether pretrained temporal representations can capture pollutant dynamics beyond simple persistence, and whether environmental covariates and domain adaptation are required to achieve reliable performance. This motivates a systematic comparison between conventional machine learning models, AutoML forecasting systems, zero-shot TSFMs, covariate-aware configurations, and fine-tuned foundation models under the same leakage-aware experimental framework [8, 18, 37].

2.7. Synthesis of Research Gaps

The reviewed literature shows that satellite observations, ground-based monitoring networks, meteorological reanalysis products, and machine learning models have substantially advanced surface-level air-quality estimation. However, this literature also reveals several methodological limitations that remain insufficiently addressed, particularly from a geoinformatics perspective.

First, although satellite–ground data fusion has become a common strategy for estimating surface-level pollutant concentrations, many studies remain focused on specific cities, pollutants, or short experimental periods [6, 11, 23]. This limits the portability of their workflows to broader geographic domains. The integration of point-based monitoring stations, satellite pixels, and gridded meteorological products requires explicit decisions about spatial co-location, aggregation, interpolation, and representativeness. Nevertheless, these operations are often treated as technical preprocessing steps rather than as methodological components that directly influence model reliability. As a result, the geoinformatics

dimension of air-quality data fusion remains underdeveloped in many environmental machine learning studies, even though spatial support, temporal synchronization, metadata consistency, and station representativeness directly condition the reliability of downstream predictive models. This gap is especially pronounced for SO_2, for which surface-level estimation using TROPOMI retrievals remains comparatively less explored in the data-fusion literature than more extensively studied pollutants such as NO_2 and particulate matter.

Second, the column-to-surface relationship remains a central unresolved challenge. Satellite atmospheric products provide spatially continuous information, but they describe vertically integrated column quantities rather than direct near-surface concentrations [34]. Previous studies address this limitation by combining satellite observations with ground measurements, meteorological variables, and machine learning models. However, the effectiveness of this transformation depends strongly on atmospheric mixing, transport, boundary-layer dynamics, retrieval quality, and station representativeness. Therefore, surface-level estimation cannot be reduced to a simple statistical mapping between satellite columns and monitoring stations; it requires a harmonized multi-source framework capable of representing both spatial scale mismatch and temporal synchronization.

Third, meteorological and reanalysis covariates are widely recognized as important predictors in pollutant estimation, but their integration remains inconsistent across studies. ERA5 and similar products provide standardized atmospheric variables that can support large-scale modeling [15]. However, their spatial and temporal resolutions differ from both satellite retrievals and ground-based observations. Choices related to resampling, temporal aggregation, overpass alignment, and feature construction can affect the resulting predictive dataset. Many studies report the use of meteorological predictors, but fewer provide a fully reproducible description of how these variables are harmonized with station and satellite data.

Fourth, model comparison in environmental machine learning remains limited by inconsistent validation designs. Regression models, tree-based ensembles, deep learning architectures, and AutoML systems have all been applied to air-quality estimation and forecasting [14, 25, 32]. However, reported performance is often difficult to compare because studies use different input datasets, feature engineering procedures, missing-data treatments, spatial domains, and evaluation metrics. In atmospheric time series, this issue is intensified by temporal autocorrelation. Random train–test splitting can introduce temporal leakage and produce overly optimistic results because observations close in time may share information across training and testing subsets [37]. Leakage-aware chronological validation is therefore necessary for evaluating realistic forecasting performance.

Fifth, the role of Time Series Foundation Models in air-quality forecasting remains insufficiently established. Models such as Chronos and TimesFM have shown promising results in general forecasting benchmarks [8, 18], and recent atmospheric applications suggest potential for environmental time series. However, atmospheric pollutant concentrations are influenced by exogenous environmental drivers that may not be captured from historical target values alone. Few studies evaluate whether zero-shot TSFMs can outperform conventional machine learning baselines for surface-level pollutant estimation, or whether their performance depends on meteorological covariates, satellite-derived atmospheric information, and domain-specific adaptation. This creates a need for systematic comparison between classical machine learning, AutoML forecasting systems, zero-shot foundation models, covariate-aware configurations, and fine-tuned TSFMs under consistent experimental conditions.

These gaps motivate the contribution of the present thesis. The proposed work addresses the need for a reproducible geoinformatics-based data fusion framework capable of integrating EEA ground observations, Sentinel-5P atmospheric products, ERA5 meteorological reanalysis, and additional atmospheric covariates within a harmonized spatial-temporal workflow. The Milan case study is used to evaluate the predictive implications of this integration for daily surface-level SO_2 estimation under leakage-aware chronological validation. By comparing conventional machine learning models, ensemble approaches, AutoML configurations, and Time Series Foundation Models under consistent experimental conditions, this thesis contributes both to scalable environmental data harmonization and to the methodological evaluation of emerging forecasting architectures for surface-level air-quality estimation.

3 | Methodology

3.1. Area of Study and Experimental Scope

This research was developed on two complementary spatial scales. The first corresponds to the European Environmental Agency (EEA) air quality monitoring network, which constitutes the continental-scale study domain used for metadata harmonization, environmental data integration, and exploratory statistical analysis. The second corresponds to the Milan metropolitan area, which was selected as the localized experimental domain for the machine learning and Time Series Foundation Model (TSFM) evaluation.

3.1.1. European Environmental Monitoring Domain

The principal study domain of this research corresponds to the European air quality monitoring network operated under the European Environment Agency (EEA) Air Quality e-Reporting framework. This network includes thousands of monitoring stations distributed across approximately 30 European countries, covering urban, suburban, industrial, traffic, and rural environments under heterogeneous climatic and atmospheric conditions, as shown in Figure 3.1. [10].

The continental-scale scope of the EEA air quality monitoring infrastructure makes it particularly suitable for environmental data engineering and large-scale geoinformatics analysis [26]. Although the EEA Air Quality e-Reporting framework provides a standardized structure for reporting air quality observations across Europe, its use within a multi-country and multi-source modeling workflow still requires study-specific preprocessing. In this research, this preprocessing was not intended to reorganize the EEA reporting system itself, but to construct an analysis-ready dataset compatible with the proposed geoinformatics and machine learning framework. The required operations included pollutant-specific filtering, station metadata consolidation, temporal aggregation, spatial reference checking, and the alignment of ground-based observations with external environmental datasets such as ERA5 meteorological reanalysis and Sentinel-5P atmospheric products [3].

The pollutants considered within the continental analysis include SO_2, NO_2, nitrogen oxides expressed as NO_2 (NO_x as NO_2), and CO. These pollutants were selected due to their regulatory relevance, direct association with anthropogenic combustion processes, and compatibility with Sentinel-5P atmospheric products.

The resulting harmonized framework supports exploratory statistical analysis of pollutant variability, station distributions, monitoring density, temporal trends, and cross-country atmospheric behavior across Europe. Additionally, the pipeline was designed to operate over arbitrary European countries or user-defined geographic domains, enabling scalable environmental machine learning applications beyond the experimental case study implemented in this thesis.

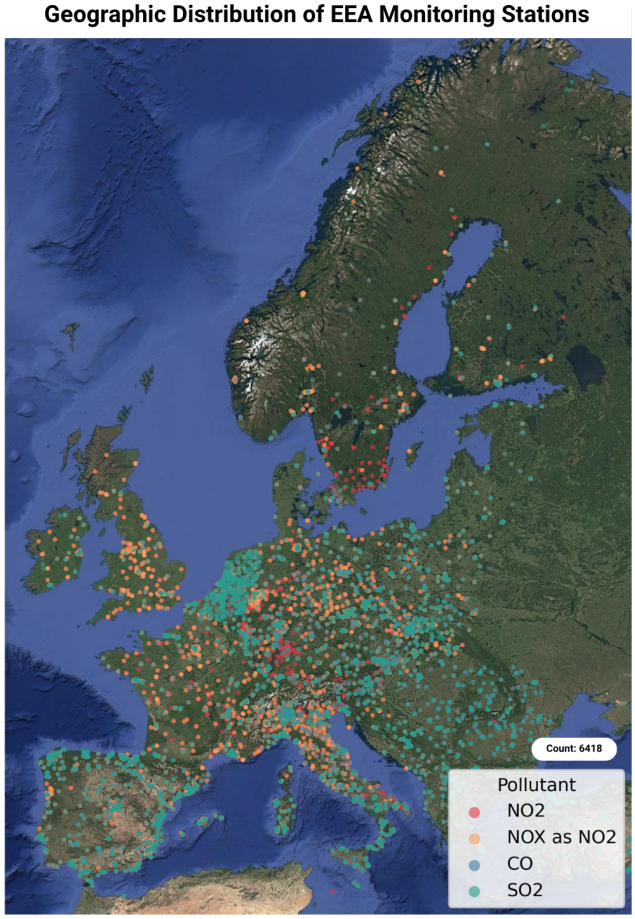

Figure 3.1: Spatial distribution of the EEA air quality monitoring network across Europe after metadata harmonization and pollutant filtering.

3.1.2. Milan Metropolitan Area

Although the environmental preprocessing and harmonization stages were developed at a continental scale, the machine learning experiments presented in this thesis were conducted using the Milan metropolitan area as a localized proof-of-concept case study as evidenced in figure 3.2.

Milan was selected because it contains a dense monitoring infrastructure, strong anthropogenic emission sources, and significant atmospheric variability suitable for pollutant estimation experiments. As one of the largest metropolitan and economic centers in Europe, the city presents elevated traffic intensity, residential energy consumption, industrial activity, and complex urban atmospheric conditions that directly influence near-surface pollutant concentrations.

The pollutants analyzed in the machine learning experiments are strongly associated with combustion-related activities. NO_2 is predominantly linked to vehicular traffic and urban transportation systems, while SO_2 is influenced by industrial emissions, fuel combustion, and regional atmospheric transport processes [21]. Additionally, temporal pollutant variability is affected by meteorological conditions including wind dynamics, precipitation, atmospheric stability, temperature, and boundary layer evolution .

Milan presents a humid subtropical climate characterized by warm summers and colder winters with marked seasonal atmospheric variability. Winter periods frequently exhibit reduced atmospheric mixing and more stable atmospheric conditions, contributing to increased pollutant accumulation near the surface. In contrast, warmer periods generally present stronger vertical mixing and improved atmospheric dispersion conditions.

The metropolitan monitoring infrastructure includes traffic stations, urban background stations, and industrial monitoring locations distributed across different urban environments. This heterogeneous station network provides suitable conditions for evaluating machine learning generalization across spatially variable atmospheric conditions while maintaining a manageable experimental domain for model development and validation.

The Milan case study was therefore used as a controlled experimental environment for evaluating the capability of machine learning algorithms and Time Series Foundation Models to estimate surface-level pollutant concentrations using harmonized EEA observations, ERA5 meteorological variables, and Sentinel-5P atmospheric products.

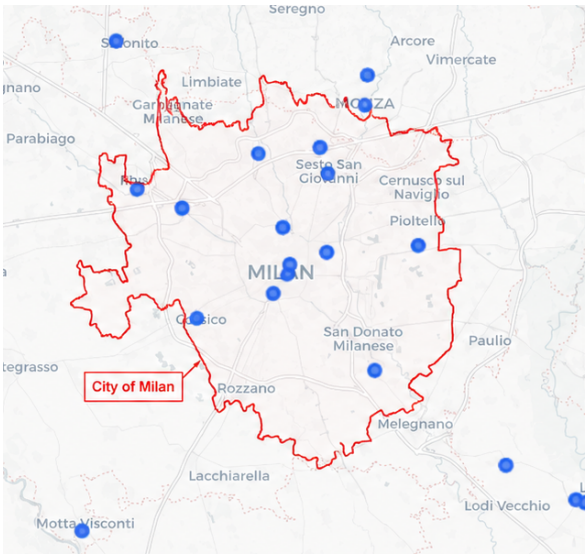

Figure 3.2: Monitoring stations and metropolitan study area used for the machine learning experiments in Milan.

3.1.3. Environmental Data Integration Context

An important characteristic of this research is the separation between the spatial scale used for environmental data engineering and the scale used for predictive modeling. While the harmonization pipeline was designed to operate across the complete European monitoring infrastructure, the machine learning experiments were intentionally constrained to a localized metropolitan case study in order to evaluate methodological robustness under controlled atmospheric conditions.

This separation enables the proposed framework to simultaneously address two complementary objectives: reproducible large-scale environmental data harmonization and localized atmospheric pollutant estimation through machine learning. Consequently, the research should not be interpreted solely as a city-level forecasting study, but rather as a scalable environmental machine learning framework capable of supporting future continental-scale atmospheric analysis and operational air quality applications.

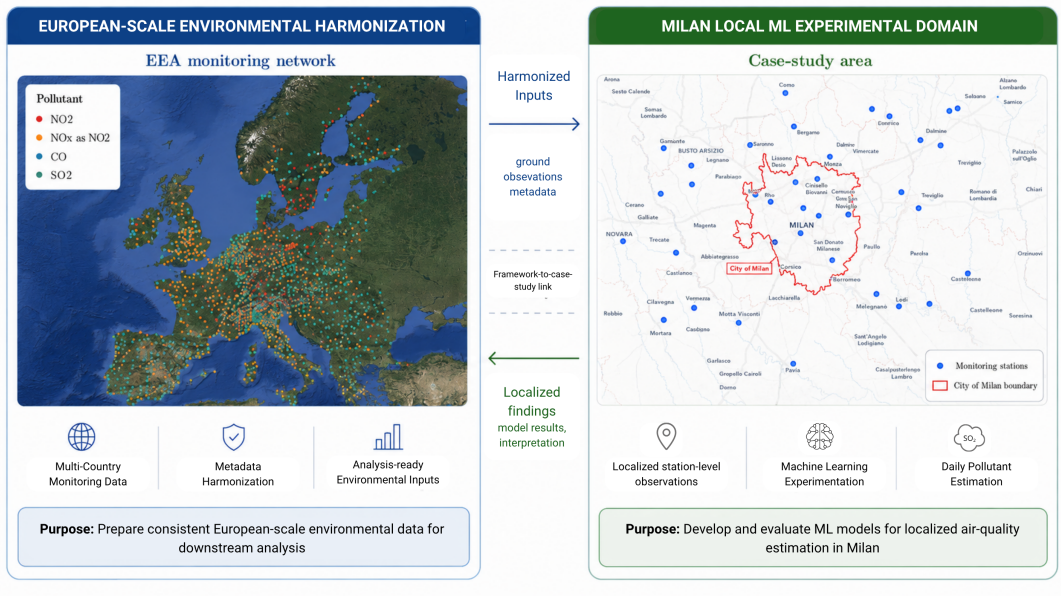

Figure 3.3: Conceptual separation between the European-scale environmental harmonization framework and the localized Milan machine learning experimental domain.

3.2. Framework Overview

This research developed a scalable environmental machine learning framework for the estimation of atmospheric pollutant concentrations through the integration of heterogeneous environmental datasets originating from independent observational systems. The framework was structured around two complementary components. The first component addresses European-scale environmental data engineering, including station metadata harmonization, pollutant filtering, spatial co-location, temporal synchronization, and multi-source data fusion. The second component evaluates predictive modeling under a localized case study in Milan, where daily NO_2 and SO_2 concentrations are estimated using classical machine learning models, AutoML configurations, and Time Series Foundation Models.

The proposed methodology combines:

- European Environment Agency (EEA) ground-based monitoring stations;
- ERA5 meteorological reanalysis products;
- Sentinel-5P atmospheric composition satellite observations;
- machine learning and Time Series Foundation Models (TSFMs).

The complete framework was designed as a modular and reproducible environmental data

pipeline capable of:

- automated atmospheric data acquisition;
- spatial and temporal harmonization;
- environmental feature engineering;
- multi-source atmospheric data fusion;
- machine learning experimentation;
- TSFM adaptation and fine-tuning.

Unlike conventional atmospheric forecasting workflows based on manually curated datasets, the proposed methodology dynamically reconstructs harmonized environmental datasets from raw atmospheric observations and geospatial products.

The implementation workflow consists of six principal stages:

1. environmental data acquisition;
2. spatial harmonization;
3. temporal synchronization;
4. environmental data fusion;
5. feature engineering and selection;
6. machine learning and TSFM experimentation.

A general overview of the complete methodology is presented in Figure 3.4

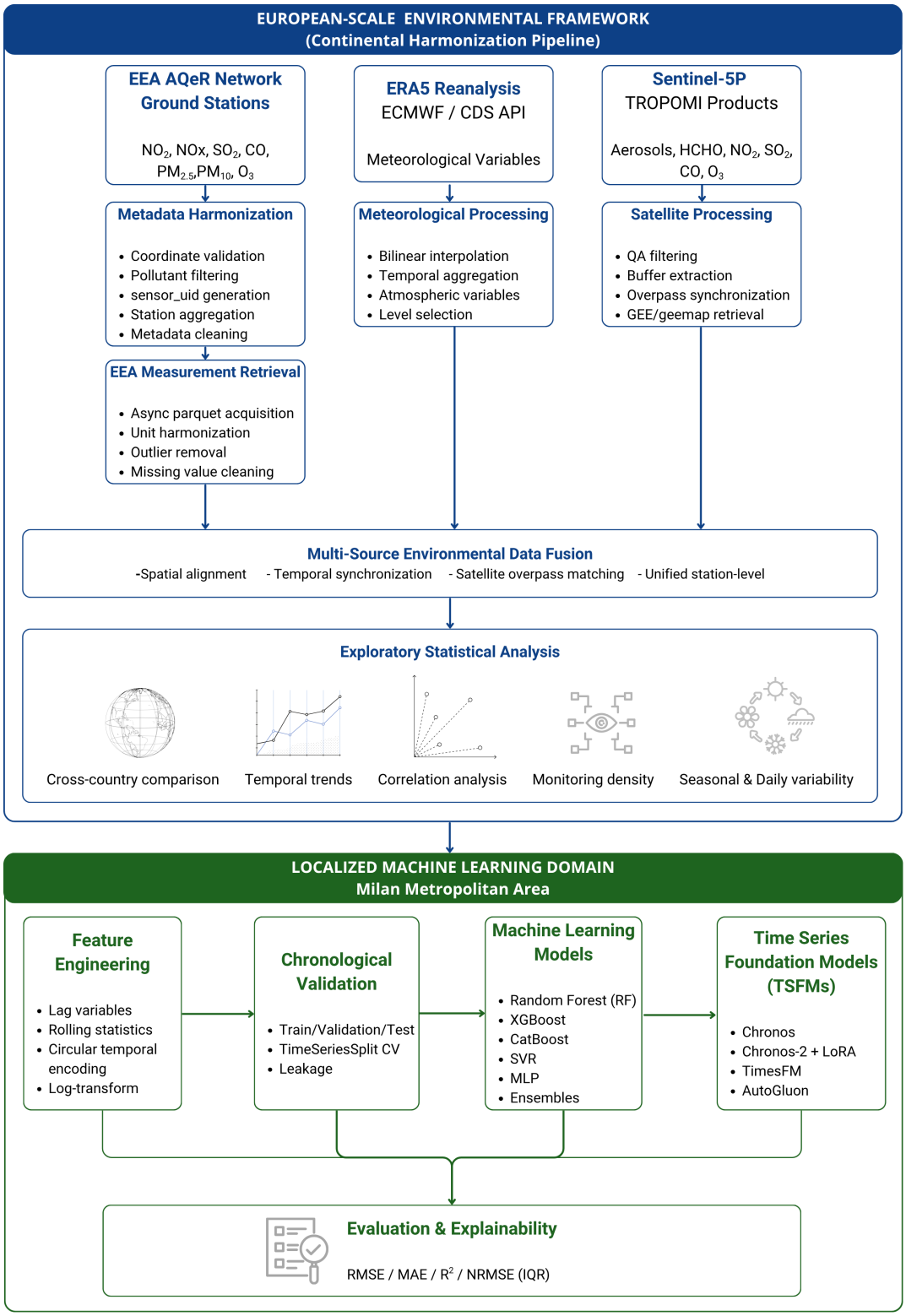

Figure 3.4: Overall methodological workflow of the proposed environmental machine learning framework.

3.3. Environmental Data Acquisition

The environmental machine learning framework relies on the integration of heterogeneous atmospheric datasets characterized by different spatial resolutions, temporal frequencies, physical interpretations, and acquisition mechanisms.

Three principal environmental sources were integrated throughout the workflow:

- European Environment Agency (EEA) atmospheric monitoring stations;
- ERA5 meteorological reanalysis products generated by ECMWF;
- Sentinel-5P TROPOMI atmospheric composition products.

The acquisition stage required the implementation of a reproducible preprocessing workflow capable of resolving inconsistencies associated with:

- heterogeneous metadata structures;
- temporal synchronization;
- spatial co-location;
- atmospheric retrieval availability;
- station-level fragmentation;
- coordinate inconsistencies.

All acquisition procedures were implemented in Python using:

- pandas;
- GeoPandas;
- xarray;
- earthengine-api;
- cdsapi;
- NumPy.

3.3.1. European Environment Agency (EEA) Ground Measurements

Ground-level atmospheric observations were obtained from the European Environment Agency (EEA) Download Service, which provides air-quality measurements collected from

national atmospheric monitoring networks across Europe.

The acquisition workflow targeted the following atmospheric pollutants:

- SO_2;
- NO_2;
- NOX as NO_2;
- CO.

A major methodological challenge emerged from the internal organization of the EEA Download Service. Instead of distributing a unified atmospheric database, observations are stored as independent Parquet files associated with individual sampling points and reporting streams. Consequently, environmental reconstruction required the implementation of a fully automated ingestion workflow capable of handling:

- asynchronous download management;
- metadata normalization;
- pollutant harmonization;
- station identity reconstruction;
- temporal aggregation;
- geospatial validation.

The EEA Download Service additionally distinguishes multiple reporting streams associated with different validation stages of the atmospheric observations. In this work, two principal reporting categories were employed:

- **E1a**: officially verified atmospheric observations;
- **E2a**: near-real-time Up-To-Date (UTD) observations.

E1a products correspond to atmospheric measurements that have undergone national quality-control procedures and formal validation before annual publication by the EEA. These datasets constitute the authoritative historical archive used for long-term atmospheric analysis.

In contrast, E2a products contain continuously transmitted near-real-time atmospheric observations that become available shortly after acquisition. Although these measurements provide more recent temporal coverage, they have not yet completed the full val-

idation process and may therefore contain provisional values, delayed corrections, or incomplete reporting.

Because verified E1a observations are released with a temporal delay, the acquisition workflow dynamically switched between reporting streams according to the target analysis period in order to maximize both historical reliability and temporal completeness.

EEA Reporting Streams Used Throughout the Acquisition Workflow

Stream	Validation Status	Temporal Coverage	Role in This Work
E1a	Officially verified atmospheric observations	Historical archive (2013–2023)	Used for validated long-term environmental analysis and model training.
E2a	Near-real-time Up-To-Date (UTD) observations	Recent observations (2024+)	Used to preserve temporal continuity before official validation becomes available.

Table 3.1: Principal EEA reporting streams integrated throughout the atmospheric acquisition workflow.

Hourly temporal aggregation was selected throughout the workflow because daily pre-aggregated products would remove the temporal variability required for synchronization with Sentinel-5P satellite overpass conditions.

Several metadata inconsistencies were identified across national reporting systems, including:

- duplicated station names;
- inconsistent text encodings;
- changing national identifiers;
- heterogeneous temporal completeness;
- pipe-delimited sampling-point identifiers.

To resolve these inconsistencies, a harmonized pollutant-agnostic identifier denoted as `sensor_uid` was constructed using normalized metadata attributes including:

- station name;
- latitude;
- longitude;
- pollutant category;
- temporal validity.

This strategy enabled reproducible merging across pollutants, countries, and EEA releases while preserving spatial consistency throughout the environmental reconstruction workflow.

Rows containing invalid coordinates, duplicated stations, impossible geographic locations, or missing latitude–longitude values were removed during preprocessing. Additionally, coordinate consistency checks were performed to ensure that all monitoring stations remained within plausible European geographic boundaries.

Because metadata files originate from heterogeneous national reporting systems, the ingestion stage additionally supported multiple text encodings, including UTF-8 and Latin-1, during CSV parsing and metadata reconstruction procedures.

The resulting harmonized station catalogue constituted the spatial backbone of the complete environmental framework and enabled subsequent integration with ERA5 meteorological variables and Sentinel-5P atmospheric products.

EEA Atmospheric Dataset Acquisition Challenges

Challenge	Implemented Solution
Independent Parquet files per station	Automated asynchronous reconstruction workflow.
Multiple reporting streams	Dynamic switching logic between E1a and E2a datasets.
Inconsistent station identifiers	Construction of harmonized `sensor_uid` identifiers.
Pipe-delimited sampling IDs	Parsing and normalization during merging operations.
Mixed national encodings	UTF-8 and Latin-1 fallback ingestion procedures.
Missing or invalid coordinates	Geospatial validation and coordinate filtering.
Heterogeneous pollutant naming	Pollutant normalization and standardized labels.

Table 3.2: Principal preprocessing and harmonization challenges encountered during EEA atmospheric data acquisition.

3.3.2. ERA5 Meteorological Reanalysis

Meteorological variables were obtained from the ERA5 atmospheric reanalysis generated by the European Centre for Medium-Range Weather Forecasts (ECMWF). ERA5 provides globally consistent atmospheric reconstructions by combining numerical weather prediction model outputs with multi-source observational data through data assimilation [15]. The dataset provides hourly deterministic meteorological fields on a regular latitude–longitude grid with a horizontal resolution of approximately $0.25° \times 0.25°$, corresponding to about 25–30 km at mid-latitudes. Vertically, ERA5 resolves the atmosphere into 137 model levels extending from the surface to 0.01 hPa, approximately 80 km above the surface, and its historical archive extends from 1950 to the present. Compared with ERA-Interim, ERA5 provides higher temporal and vertical resolution, making it particularly suitable for environmental and atmospheric machine learning applications requiring continuous meteorological information [15].

In addition to the deterministic reconstruction, ERA5 includes an uncertainty component based on a 10-member Ensemble of Data Assimilations (EDA) [15]. This ensemble product provides 3-hourly atmospheric fields at a coarser horizontal resolution of approximately 63 km, allowing uncertainty in the atmospheric reconstruction to be assessed when required [23]. In this thesis, the deterministic hourly ERA5 product was used as the primary meteorological input because the objective was to construct daily station-level predictors compatible with the air-quality forecasting workflow.

The ERA5 acquisition workflow targeted meteorological variables known to influence pollutant transport, dispersion, accumulation, removal, and chemical transformation. The extracted variables included:

- 2-m temperature;
- surface pressure;
- 10-m zonal and meridional wind components;
- boundary layer height;
- total precipitation;
- relative humidity, where available or derived from atmospheric humidity variables;
- surface solar radiation downwards.

These variables were selected because they describe key atmospheric processes affecting near-surface pollutant variability. Wind components control horizontal transport and

ventilation, boundary layer height defines the vertical volume available for pollutant dispersion, precipitation contributes to wet deposition, temperature and humidity influence atmospheric stability and chemical processes, and solar radiation affects photochemical reactions [35].

ERA5 products were retrieved from the Copernicus Climate Data Store (CDS) using the `cdsapi` Python library, typically in NetCDF format for automated processing. For the European-scale preprocessing workflow, ERA5 data were extracted over user-defined spatial bounding boxes corresponding to the selected countries or regions. For the Milan case study, the extraction domain covered the Metropolitan City of Milan and its surrounding area in order to capture the meteorological conditions influencing local pollutant transport, dispersion, and accumulation.

A major implementation challenge was the spatial mismatch between the regular ERA5 grid and the irregular geometries of the air-quality monitoring network, which are represented as point-based EEA station locations. To address this issue, ERA5 grid-cell values were spatially co-located with each monitoring station using nearest-neighbor extraction based on station coordinates. These spatial operations were implemented using `xarray`, `NumPy`, and `GeoPandas`.

Temporal harmonization was subsequently applied to transform the hourly ERA5 fields into daily station-level predictors compatible with the daily pollutant forecasting framework. Depending on the variable, daily aggregation was performed using mean or accumulated values. Temperature, pressure, wind components, boundary layer height, relative humidity, and surface solar radiation were summarized using daily averages, whereas total precipitation was aggregated as daily accumulated precipitation. The resulting daily ERA5-derived variables were then aligned with the daily EEA ground-level pollutant observations and integrated into the final analysis-ready dataset used for feature engineering, model training, and evaluation.

3.3.3. Sentinel-5P Atmospheric Products

Atmospheric composition products were obtained from Sentinel-5P TROPOMI observations. Sentinel-5P is the first Copernicus mission dedicated specifically to atmospheric monitoring, and its main objective is to provide high spatio-temporal resolution measurements for air quality, ozone, ultraviolet radiation, climate monitoring, and forecasting applications [7, 34]. The mission carries TROPOMI, a passive multispectral imaging spectrometer that measures backscattered solar radiation in the ultraviolet, visible, near-infrared, and shortwave infrared spectral ranges. These spectral measurements are used

to retrieve atmospheric trace gases and aerosol-related products.

In this study, Sentinel-5P products were accessed through two complementary data ecosystems: the Copernicus Data Space Ecosystem (CDSE) and Google Earth Engine (GEE). CDSE represents the official Copernicus infrastructure for accessing Sentinel mission products, including original Sentinel-5P Level-1B and Level-2 atmospheric products. It provides authoritative product metadata, sensing-time information, processing baselines, orbit information, and access to native product files. For this reason, CDSE and the Copernicus Browser were used to inspect product availability, verify overpass timing, and identify the appropriate sensing windows over the selected Area of Interest [9].

Google Earth Engine was used as the operational extraction environment because it provides cloud-based access to gridded Sentinel-5P collections and allows spatial filtering, temporal filtering, quality masking, clipping, and regional reduction to be performed without downloading large volumes of raw satellite files locally [23]. In particular, GEE provides Sentinel-5P offline Level-3 collections derived from the original Level-2 products. These Level-3 products are spatially gridded and therefore more suitable for automated integration with geospatial workflows and station-based extraction procedures [13, 27].

A key technical distinction in the Sentinel-5P workflow concerns the processing level of the data product. Native Sentinel-5P Level-2 products are orbit-based geophysical retrievals derived from TROPOMI spectral measurements. They are not distributed on a regular latitude–longitude grid, but as swath-based observations with product-specific pixel footprints. For the SO_2 Level-2 product, the nominal spatial resolution is approximately 3.5 × 7.0 km at the beginning of the mission and approximately 3.5 × 5.5 km after 6 August 2019. The SO_2 Level-2 product provides sulphur dioxide vertical column density, expressed in mol m^{-2}, together with retrieval uncertainty and product-specific quality information [7, 34].

In contrast, the Sentinel-5P products used operationally in GEE correspond to offline Level-3 collections derived from the original Level-2 retrievals. These Level-3 products are generated by spatially binning and gridding the orbit-based Level-2 observations into a regular geographic raster. In GEE, the Sentinel-5P Level-3 collections are provided at approximately 0.01° spatial resolution, corresponding to about 1 km at the equator. Therefore, the apparent grid spacing of the GEE product should not be interpreted as the native TROPOMI footprint. It represents a resampled gridded product derived from coarser satellite retrieval pixels [13].

For this thesis, the `SO2_column_number_density` band from the Sentinel-5P OFFL/L3 SO_2 collection was used as the main Sentinel-5P SO_2 product used from GEE. This

variable represents the vertically integrated SO_2 column amount and is expressed in mol m^{-2}. It is therefore not a surface concentration in μg m^{-3} and cannot be directly compared with EEA ground-based SO_2 measurements without additional modeling. In the final dataset, this satellite-derived column quantity was used as an atmospheric covariate within the multi-source machine learning framework, rather than as the target variable or as a direct substitute for station observations [13].

From a spatio-temporal perspective, Sentinel-5P observations differ substantially from both EEA ground measurements and ERA5 meteorological fields. Spatially, each Sentinel-5P value represents an atmospheric column over a finite satellite pixel or gridded cell, whereas EEA observations correspond to point-based monitoring stations. Consequently, spatial harmonization was required to relate satellite-derived atmospheric information to the station-level air-quality dataset [3].

Spatial extraction was adapted to the geographic scale of each workflow component. At the European scale, Sentinel-5P products were retrieved over selected countries or regional bounding boxes to support atmospheric data integration across multiple monitoring contexts. In the Milan case study, the extraction domain was restricted to the Metropolitan City of Milan and its surrounding area so that the satellite-derived atmospheric columns remained representative of the local SO_2 forecasting domain. The selected products were then spatially filtered, clipped to the Area of Interest, quality-screened, and aggregated or co-located with EEA monitoring stations before integration with the ground-based observations.

Temporally, Sentinel-5P is a polar-orbiting mission and therefore provides approximately one overpass opportunity per day over a given location, typically in the early afternoon local solar time. However, this does not imply the availability of one valid retrieval for every day, since cloud contamination, low-quality retrievals, viewing geometry, snow or ice conditions, and product-specific quality flags may remove observations from the usable dataset. For this reason, the workflow distinguished between the nominal overpass frequency and the actual availability of valid satellite-derived values after quality filtering [16, 23, 34].

The temporal extraction period was defined to match the availability of the corresponding EEA ground observations used in the modeling workflow. In the Milan case study, Sentinel-5P retrievals were synchronized with the daily SO_2 forecasting dataset by assigning valid satellite observations to their corresponding acquisition dates. The satellite extraction windows were configured around the approximate Sentinel-5P overpass interval identified from Copernicus Browser and product sensing metadata. After quality filtering,

the resulting satellite-derived time series were aggregated to daily values and aligned with the daily EEA pollutant observations and ERA5-derived meteorological predictors.

The Sentinel-5P variables considered in the acquisition workflow corresponded to vertically integrated atmospheric column products for:

- SO_2;
- NO_2;
- CO.

These variables were selected because they are directly relevant to atmospheric composition analysis and because they can complement ground-based air-quality observations in multi-source environmental modeling. However, Sentinel-5P retrievals do not represent direct near-surface concentrations. Most products are expressed as vertically integrated atmospheric column quantities, commonly reported in units such as mol m^{-2} or molecules cm^{-2}, whereas ground monitoring stations report surface-level concentrations, commonly in μg m^{-3}. Therefore, Sentinel-5P observations were used as explanatory atmospheric covariates rather than as direct substitutes for EEA ground-based measurements [20, 34].

The preprocessing requirements depend on the product level. Level-1B products contain calibrated and geolocated radiance measurements, whereas Level-2 products contain geophysical atmospheric retrievals such as trace-gas vertical column densities, retrieval uncertainties, and product-specific quality information [9]. Level-3 products, such as those available in GEE, are gridded derivatives of Level-2 retrievals designed for spatial aggregation and raster-based geospatial analysis [13]. Since the objective of this thesis was to construct an analysis-ready dataset for environmental machine learning, the workflow relied primarily on GEE Level-3 products for operational extraction, while CDSE was used to support product verification, metadata inspection, and overpass-time definition [3].

Several methodological limitations were considered during Sentinel-5P integration. Retrieval quality can be affected by cloud cover, aerosol interference, surface reflectance, snow or ice conditions, low solar illumination angles, and product-specific uncertainty thresholds [6]. These effects may produce missing observations or spatially discontinuous valid pixels, particularly during winter periods and for products with lower signal-to-noise ratios such as SO_2. In addition, the spatial footprint of Sentinel-5P pixels does not correspond directly to point-based monitoring stations, creating a spatial co-location problem when satellite observations are integrated with ground measurements. A further temporal limitation arises from the polar-orbiting nature of the mission, which provides approxi-

mately one observation opportunity per day over a given location and therefore cannot capture intra-day pollution variability without additional data sources [22].

To reduce these limitations, quality-assurance filtering was applied before downstream analysis. Product-specific quality bands, including the `qa_value` parameter when available, were used to exclude unreliable retrievals. Spatial filtering was performed using user-defined bounding boxes corresponding to the selected study areas, and temporal filtering was applied according to the acquisition period of the corresponding EEA ground observations. For the Milan case study, satellite extraction windows were synchronized with the approximate Sentinel-5P overpass interval identified from Copernicus Browser and product sensing metadata. This ensured that satellite atmospheric columns were temporally consistent with the daily ground-level pollutant observations used in the forecasting framework [3, 6].

The Sentinel-5P extraction workflow in GEE included:

- selection of the atmospheric product and variable of interest;
- filtering by acquisition date;
- filtering by geographic Area of Interest;
- application of product-specific quality masks;
- spatial clipping to the study domain;
- extraction or aggregation of valid satellite pixels over station locations or regional geometries;
- export of the resulting time series for integration with EEA and ERA5 datasets.

The resulting Sentinel-5P-derived variables were then harmonized with EEA ground-based observations and ERA5 meteorological predictors. Within the final modeling dataset, these satellite products provided spatially continuous atmospheric composition information, while the EEA observations provided the surface-level reference values and ERA5 described the meteorological conditions controlling pollutant transport, dispersion, and accumulation.

3.4. Spatial Harmonization

The environmental datasets integrated throughout this work possess fundamentally different spatial structures. EEA observations correspond to irregular point-based monitoring

stations, ERA5 variables are distributed on regular atmospheric grids, and Sentinel-5P products represent atmospheric retrievals associated with satellite pixel footprints.

Consequently, spatial harmonization was required before environmental data fusion could be performed.

All datasets were standardized under the WGS84 geographic coordinate reference system (EPSG:4326). This CRS was selected because it corresponds to the native spatial reference system used by:

- Sentinel-5P products;
- Google Earth Engine geometries;
- ERA5 latitude–longitude grids.

The harmonization workflow included:

- coordinate normalization;
- geospatial validation;
- station filtering;
- geometry construction;
- spatial co-location procedures.

After validation, monitoring stations were converted into geospatial point objects using `GeoPandas` geometry structures. This transformation enabled direct interoperability with ERA5 grids and Sentinel-5P atmospheric products.

3.5. Temporal Harmonization

Environmental datasets operate under heterogeneous temporal resolutions and acquisition schedules. EEA ground-based observations may be reported at hourly or daily resolution, ERA5 meteorological products are distributed hourly, and Sentinel-5P observations are constrained by the satellite overpass schedule. Since Sentinel-5P is a polar-orbiting mission, it provides approximately one overpass opportunity per day over a given location [16]. However, this does not imply the availability of one valid atmospheric retrieval per day, because cloud cover, low-quality retrievals, unfavorable viewing conditions, or product-specific quality-assurance flags may lead to missing or discarded observations.

To address these inconsistencies, all datasets were synchronized into a common daily temporal framework. EEA hourly observations were aggregated to daily pollutant con-

centrations when required, ERA5 hourly meteorological variables were converted into daily summaries, and Sentinel-5P retrievals were filtered and aggregated according to the valid observations available within the daily acquisition window.

Temporal alignment was implemented using:

- `pandas` datetime indexing;
- UTC-standardized timestamps;
- chronological ordering procedures.

Particular attention was devoted to Sentinel-5P synchronization. Atmospheric extraction windows were configured around the approximate satellite overpass time over the selected Area of Interest in order to reduce the temporal mismatch between atmospheric column retrievals and ground-level pollutant concentrations [6]. When no valid Sentinel-5P pixels were available for a given day after quality filtering, the corresponding satellite-derived variable was treated as missing and handled during the subsequent dataset construction and modeling stages [22].

Additionally, all downstream temporal operations were implemented under strict chronological ordering to prevent temporal leakage during machine learning modeling. Lag features, rolling statistics, train–validation–test partitions, and forecasting windows were computed using only information available prior to the target timestamp [36].

Lag features, rolling statistics, train-validation-test partitions, and forecasting windows were computed using only information available prior to the target timestamp, and are further discussed in Section 3.6 and Section 3.9.

3.6. Feature Engineering

After completing the harmonization and cleaning stages, the integrated environmental dataset was transformed into a structured predictor space suitable for atmospheric machine learning and Time Series Foundation Model (TSFM) experimentation. The feature engineering stage was designed to explicitly represent temporal atmospheric persistence, seasonal variability, meteorological forcing, and satellite-derived atmospheric composition while preserving strict chronological consistency throughout the workflow.

All feature engineering operations were implemented using the `pandas`, `NumPy`, and `scikit-learn` Python libraries after chronologically sorting the dataset and defining the observation date as the primary temporal index.

A critical methodological requirement throughout this stage was the prevention of temporal leakage. Consequently, all lagged variables, rolling statistics, scaling operations, and aggregated descriptors were constructed exclusively using information available prior to the target prediction timestamp.

The feature engineering workflow was implemented sequentially according to:

1. chronological sorting and temporal indexing;
2. lag feature generation;
3. rolling statistical aggregation;
4. cyclical temporal encoding;
5. meteorological transformation;
6. satellite feature integration;
7. feature normalization;
8. removal of incomplete temporal structures.

Figure 3.5 presents the complete workflow implemented during the feature construction stage.

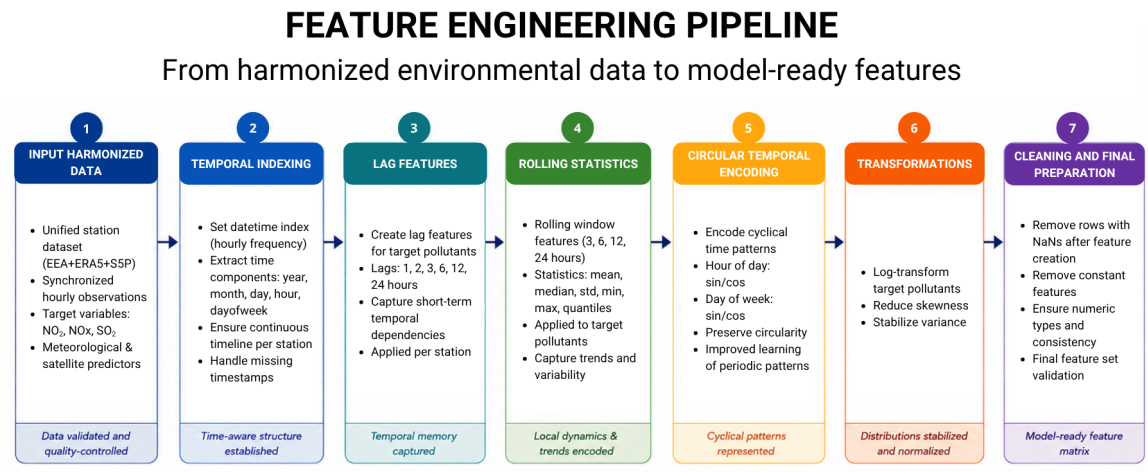

Figure 3.5: Feature engineering workflow including temporal lag generation, rolling statistics, cyclical encoding, meteorological transformations, satellite-derived predictors, and feature normalization.

3.6.1. Lag Features

Atmospheric pollutant concentrations exhibit strong temporal autocorrelation due to the persistence of meteorological conditions, emission patterns, and atmospheric transport processes. Consequently, autoregressive lag variables constitute highly informative predictors for surface-level atmospheric forecasting [21, 24].

Lag variables were generated using the `pandas.shift()` operation applied to chronologically ordered pollutant series. The target variable used throughout the experiments corresponds to ground-level SO_2 concentrations expressed in $\mu g\ m^{-3}$.

The engineered lag structure is summarized in Table 3.3.

Autoregressive Lag Features

Feature	Purpose
`so2_lag1`	Captures short-term daily atmospheric persistence.
`so2_lag2` and `so2_lag3`	Represent multi-day pollutant evolution patterns.
`so2_lag7`	Represents weekly atmospheric recurrence.
`so2_lag14`	Captures medium-term temporal persistence.
Lagged co-pollutants	Incorporates delayed interactions between SO_2, NO_2, NO_x, and CO.

Table 3.3: Principal autoregressive lag features incorporated into the atmospheric predictor space.

The general lag construction follows:

$$x_{\text{lag1}}(t) = x(t-1) \tag{3.1}$$

where x represents the corresponding atmospheric variable.

Because lag generation inherently produces incomplete observations at the beginning of the temporal series, rows containing missing lag structures were removed. This was done only after all engineered variables had been generated, in order to preserve temporal alignment consistency across the complete feature matrix.

3.6.2. Rolling Statistics

Although discrete lag variables capture direct temporal persistence, atmospheric pollutant dynamics depend on smoothed concentration trends and temporal variability evolving across broader temporal windows. To represent these dynamics, rolling statistical descriptors were incorporated into the predictor space [25].

These were computed exclusively on lagged pollutant series in order to preserve strict chronological consistency. The target series was first shifted by one timestep before applying any rolling aggregation:

$$SO_2^{\text{shifted}}(t) = SO_2(t-1) \tag{3.2}$$

This guarantees that rolling windows never include same-day or future observations.

The rolling statistical structure is summarized in Table 3.4.

Rolling Statistical Features

Feature	Purpose
`so2_roll3_mean`	Short-term atmospheric smoothing and persistence.
`so2_roll7_mean`	Weekly pollutant trend representation.
`so2_roll14_mean`	Medium-term atmospheric trend aggregation.
`so2_roll7_std`	Atmospheric volatility and instability descriptor.

Table 3.4: Rolling statistical descriptors integrated into the feature space.

Rolling windows were implemented using the `pandas.rolling()` framework with flexible minimum-period constraints in order to reduce excessive data loss during the initial temporal windows while preserving statistically meaningful aggregation behavior.

3.6.3. Cyclical Encoding

Atmospheric pollutant concentrations exhibit strong seasonal and weekly periodicity associated with meteorological variability, anthropogenic activity, and emission cycles. Initial experiments implemented conventional one-hot temporal encoding; however, itera-

tive model refinement revealed that the resulting dummy variables excessively increased feature-space dimensionality and dominated the TOP_K feature-selection process, limiting the inclusion of more informative atmospheric predictors [36].

The transition from one-hot temporal encoding to cyclical representations during the iterative feature-engineering refinement process is illustrated in Figure 3.6.

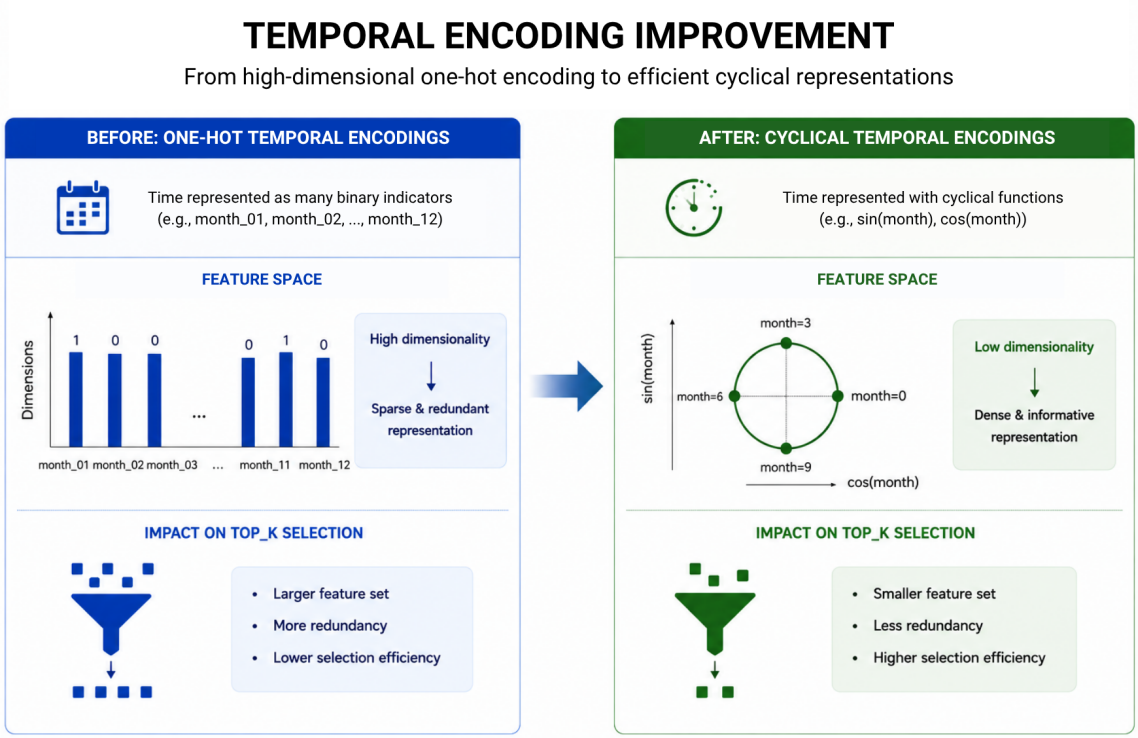

Figure 3.6: Comparison between initial one-hot temporal encoding and the final cyclical temporal representation adopted after iterative feature-engineering refinement. The cyclical encoding reduced feature dimensionality while preserving temporal continuity and improving TOP_K feature-selection efficiency.

To preserve cyclical continuity while reducing dimensionality, temporal descriptors were subsequently transformed using sine and cosine representations according to:

$$x_{\sin} = \sin\left(\frac{2\pi x}{P}\right) \tag{3.3}$$

$$x_{\cos} = \cos\left(\frac{2\pi x}{P}\right) \tag{3.4}$$

where P corresponds to the periodicity of the temporal cycle.

The engineered cyclical variables are summarized in Table 3.5.

Cyclical Temporal Features

Feature Group	Purpose
`month_sin`, `month_cos`	Represents annual atmospheric seasonality.
`doy_sin`, `doy_cos`	Encodes day-of-year temporal continuity.
`dow_sin`, `dow_cos`	Captures weekly anthropogenic activity cycles.

Table 3.5: Cyclical temporal variables integrated into the predictor space.

Replacing one-hot temporal encodings with cyclical representations reduced feature-space dimensionality while preserving periodic atmospheric behavior, improving the efficiency of subsequent TOP_K feature-selection procedures [6, 37].

3.6.4. Meteorological Features

Meteorological conditions strongly influence pollutant transport, atmospheric dispersion, deposition, and chemical transformation processes [26]. Consequently, ERA5 meteorological reanalysis variables were incorporated into the predictor space as physically meaningful atmospheric descriptors.

The integrated meteorological variables are summarized in Table 3.6.

Meteorological Predictors

Variable	Atmospheric Relevance
Temperature	Influences atmospheric chemistry and pollutant formation.
Surface pressure	Associated with atmospheric stability conditions.
Total precipitation	Represents wet deposition and pollutant removal.
Boundary layer height	Controls vertical pollutant dispersion and accumulation.
Wind components (U/V)	Represent atmospheric transport dynamics.
Solar radiation	Influences photochemical atmospheric reactions.
Thermal radiation	Associated with surface-atmosphere energy exchange.

Table 3.6: ERA5 meteorological variables integrated into the environmental predictor space.

Wind direction required additional preprocessing because angular atmospheric variables exhibit circular topology. Initial experiments represented wind direction using conventional 8-sector categorical bins; however, iterative refinement of the machine learning pipeline revealed that discrete sectorization introduced artificial directional discontinuities and reduced the continuity of atmospheric transport representation.

To improve directional consistency, the final workflow replaced the 8-sector categorical encoding with continuous sine and cosine transformations of the wind angle. Wind direction aggregation was therefore performed using circular statistics according to:

$$\theta_{\text{mean}} = \text{atan2}\left(\overline{\sin(\theta)}, \overline{\cos(\theta)}\right) \tag{3.5}$$

where:

- θ represents the wind direction angle;
- the overline denotes temporal averaging.

The resulting directional representation preserves angular continuity while eliminating artificial discontinuities near the $0°/360°$ transition boundary, improving the physical representation of atmospheric transport processes within the machine learning feature space.

3.6.5. Satellite Features

Satellite-derived atmospheric variables obtained from Sentinel-5P TROPOMI products were incorporated into the feature space in order to provide spatially continuous atmospheric composition information complementary to point-based monitoring observations. The principal satellite-derived predictors included atmospheric column retrievals of SO_2, NO_2, and CO.

Because Sentinel-5P products represent atmospheric column densities rather than direct surface-level concentrations, satellite variables were interpreted as explanatory atmospheric descriptors associated with regional atmospheric composition and pollutant transport behavior.

Buffer-based spatial extraction procedures were implemented to mitigate geometric mismatch between irregular satellite pixel footprints and point-based monitoring stations. Atmospheric statistics were therefore aggregated within predefined spatial neighborhoods surrounding each monitoring location.

Finally, all Sentinel-5P retrievals were filtered using:

$$qa\_value \geq 0.5$$

before spatial aggregation in order to exclude low-quality atmospheric observations affected by cloud contamination, retrieval instability, unfavorable viewing geometry, or insufficient radiometric confidence [34].

The `qa_value` parameter provided within Sentinel-5P Level-2 products represents a quality assurance indicator ranging from 0 to 1, where larger values correspond to more reliable atmospheric retrievals. Applying this filtering threshold improves the physical consistency and reliability of the satellite-derived atmospheric predictors integrated into the machine learning workflow.

3.6.6. Feature Scaling and Normalization

Prior to model training, predictor variables and target pollutant concentrations were normalized using the `MinMaxScaler` implementation from `scikit-learn.preprocessing`. Separate scaling functions were fitted for predictors and target variables using only the training subset in order to prevent information leakage into validation and testing observations.

The normalization procedure follows:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3.6}$$

where x' represents the normalized variable.

After prediction, inverse transformations were applied to restore pollutant concentrations to their original physical units before evaluation.

3.7. Feature Selection

After constructing the complete predictor space described previously, a feature selection stage was implemented to identify the subset of variables that contributed most to predictive performance while reducing redundancy, dimensionality, and model instability. This step was particularly important because the environmental dataset combined heterogeneous predictors, including pollutant lag variables, rolling statistics, meteorological descriptors, satellite-derived atmospheric products, and temporal features.

Feature selection is especially relevant in atmospheric machine learning because environmental predictors are often highly correlated and may encode overlapping physical processes. For example, lagged pollutant concentrations and rolling averages may describe similar temporal persistence patterns, while meteorological variables such as pressure, boundary layer height, wind components, precipitation, and radiation may interact non-linearly with pollutant accumulation, dispersion, and removal. If all variables are retained without filtering, the resulting feature space may increase computational cost, reduce interpretability, amplify overfitting risk, and degrade model generalization performance [20].

The feature selection workflow implemented in this research followed a progressive strategy, moving from simple linear screening to model-based predictive importance. First, Pearson correlation analysis was used as an initial screening step to identify variables with a measurable linear association with the SO_2 target variable. Features satisfying the correlation threshold were retained as an initial subset. This step provided a transparent and interpretable first approximation of predictor relevance, but it was limited to linear relationships and continuous variables.

The performance of the Pearson-selected feature subset was then evaluated within the modelling workflow. This intermediate evaluation was necessary because correlation-based filtering may exclude variables that are weakly correlated with the target individually but useful in combination with other predictors. It may also fail to capture non-linear effects,

interaction effects, and redundant or collinear predictors. Therefore, when the correlation-selected subset did not provide sufficient predictive performance, a second feature selection strategy based on Random Forest feature importance was implemented.

Random Forest feature importance was used to rank predictors according to their contribution to predictive performance within a non-linear tree-based model. Unlike Pearson correlation, this approach can capture non-linear relationships and interaction effects between predictors. The resulting importance scores were used to select the TOP_K most informative variables, producing a compact and more robust predictor subset for subsequent modelling experiments. This final feature set was intended to balance predictive accuracy, interpretability, and computational efficiency.

Figure 3.7 summarizes this feature selection workflow, showing the transition from Pearson correlation screening to Random Forest importance ranking after the evaluation of the initial feature subset.

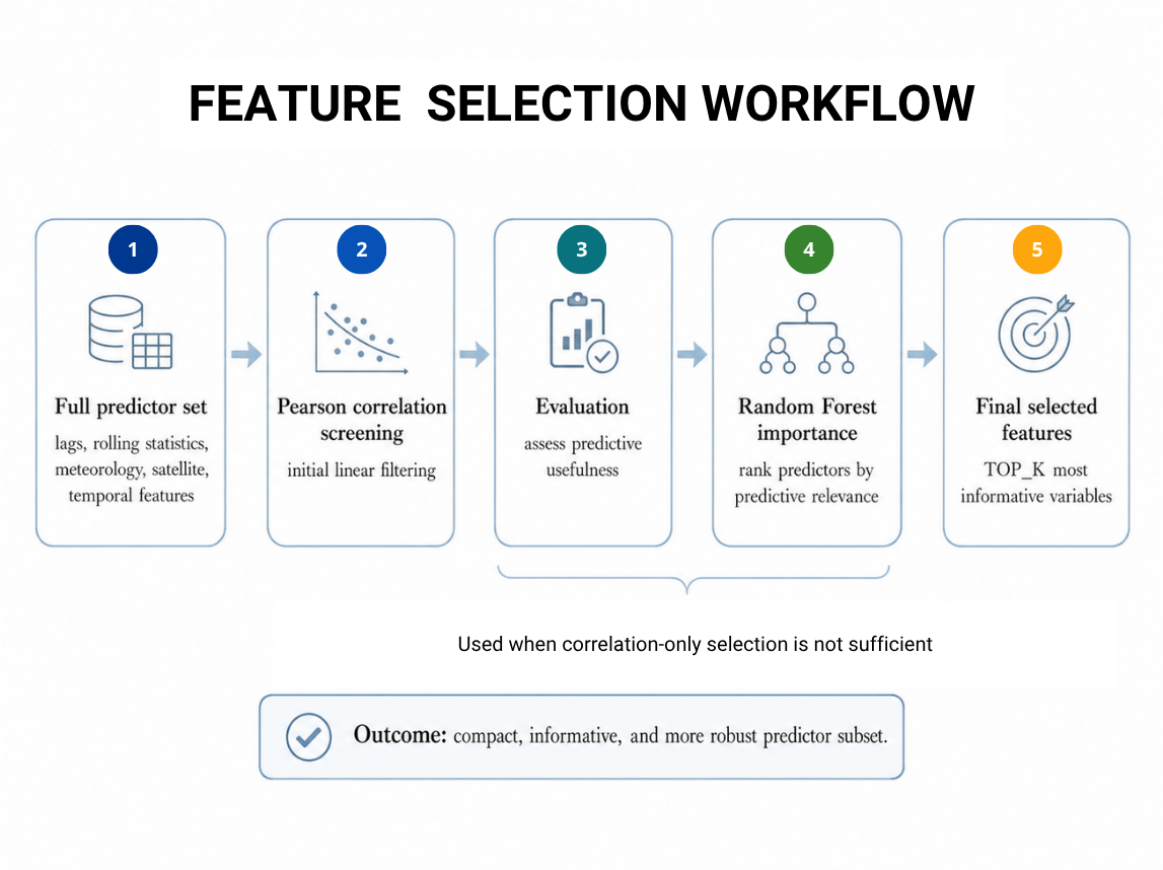

Figure 3.7: Feature selection pipeline implemented in this research. The workflow begins with Pearson correlation screening to identify predictors with linear association to the SO_2 target variable. The resulting subset is then evaluated in terms of predictive performance. Since correlation-based selection may miss non-linear relationships, interaction effects, and redundant predictor structures, Random Forest feature importance is subsequently used to rank variables by predictive relevance and select the TOP_K most informative features.

3.7.1. Correlation Analysis

Before applying model-based feature selection, an exploratory correlation analysis was performed to evaluate the statistical relationships between predictors and identify highly redundant variables within the engineered feature space.

The analysis was implemented using Pearson correlation coefficients computed through the `pandas.corr()` functionality. Correlation matrices were generated for pollutant variables, lag structures, meteorological descriptors, and satellite-derived atmospheric features.

The principal objective of this stage was not to directly select predictors exclusively based on correlation magnitude, but rather to characterize the structure of the predictor space before downstream model training.

Although Pearson correlation analysis provides useful exploratory insight, it presents important limitations for atmospheric machine learning applications because it only measures linear dependencies between individual variables. Atmospheric systems exhibit highly nonlinear interactions involving coupled meteorological processes, pollutant transport dynamics, and seasonal atmospheric variability.

For example, the influence of wind speed on pollutant dispersion or the interaction between temperature and atmospheric stability cannot be adequately represented through simple linear correlation metrics [32].

Consequently, correlation analysis was treated primarily as a diagnostic and interpretability tool rather than as the principal feature selection mechanism.

3.7.2. Random Forest Feature Importance

To overcome the limitations of linear correlation-based selection, the final predictor ranking was performed using a Random Forest Regressor feature importance strategy implemented through the `scikit-learn` library.

Tree-based importance ranking was selected because ensemble decision-tree methods are capable of capturing nonlinear predictor-target relationships, variable interactions, threshold effects, and non-monotonic atmospheric behavior.

The Random Forest model estimates feature importance by measuring how much each variable contributes to reducing prediction uncertainty across the ensemble of decision trees. During training, the algorithm repeatedly splits the data according to the variables that most effectively reduce prediction error within each node. Features that consistently produce larger reductions in impurity across multiple trees receive higher importance scores.

The feature-importance analysis was computed exclusively on the training subset in order to prevent information leakage into the validation and testing stages. This constraint is particularly important in atmospheric time-series modeling because incorporating future observations during preprocessing may artificially inflate predictive performance and compromise the validity of the experimental evaluation [1].

The Random Forest selector was trained using the complete engineered feature space

including:

- lag variables;
- rolling statistics;
- cyclical temporal descriptors;
- meteorological variables;
- satellite-derived atmospheric features.

After training, the predictors were ranked according to their normalized importance scores.

3.7.3. TOP_K Feature Selection Strategy

After computing the Random Forest importance ranking, a fixed-size feature selection strategy was applied in order to constrain the final predictor space used during model training.

The workflow retained the:

$$TOP\_K = 15$$

highest-ranked predictors according to the Random Forest importance scores.

The selection of a fixed TOP_K strategy was motivated by several engineering considerations:

- reducing feature-space dimensionality;
- improving computational efficiency;
- limiting overfitting risk;
- improving model interpretability;
- maintaining consistent predictor dimensionality across experiments.

The TOP_K filtering stage was implemented dynamically after feature ranking, allowing the selected predictors to vary depending on the engineered feature space available within each experiment.

An important methodological limitation emerged during the initial baseline experiments due to the interaction between one-hot temporal encoding and the fixed TOP_K con-

straint. Specifically, the twelve binary month variables systematically dominated the Random Forest ranking and occupied most of the available feature slots. As a consequence, physically meaningful meteorological and autoregressive predictors were excluded from the final selected feature space.

To address this issue, the enhanced pipeline replaced one-hot temporal encoding with cyclical sine-cosine representations described in Section 3.5.3. This modification reduced the temporal dimensionality from twelve variables to two while preserving seasonal continuity. The reduction in temporal feature-space occupancy freed additional TOP_K capacity for more informative atmospheric predictors.

3.7.4. Feature Normalization

After completing feature selection, the retained predictors were normalized before model training using Min-Max scaling implemented through the `scikit-learn` preprocessing framework.

Each selected feature was transformed into the interval:

$$[0, 1]$$

according to:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3.7}$$

where:

- x corresponds to the original feature value;
- x_{min} and x_{max} represent the minimum and maximum values computed exclusively on the training dataset.

Scaling parameters were fitted exclusively on the training subset and subsequently applied to the validation and testing datasets in order to preserve strict chronological separation.

Normalization was important for models sensitive to feature magnitude, including Support Vector Regression, neural architectures, and attention-based models.

Finally, the selected feature lists, fitted scalers, and normalization parameters were serialized using the `joblib` [1] framework in order to preserve reproducibility and enable future

[1] `https://joblib.readthedocs.io/en/stable/`

inference using identical preprocessing configurations.

3.8. Machine Learning Models

This section describes the model families evaluated in the thesis. Quantitative performance comparisons are not reported here, but are presented in Chapter 5.

After completing the environmental harmonization, cleaning, feature engineering, and feature-selection stages, multiple machine learning paradigms were evaluated for surface-level SO_2 estimation. The objective of this stage was not only to compare predictive performance, but also to analyze how different learning strategies respond to heterogeneous atmospheric predictors, temporal autocorrelation structures, and nonlinear environmental interactions.

The experimental framework combined:

- parametric statistical regression;
- bagging-based ensemble learning;
- gradient-boosting algorithms;
- automated ensemble forecasting;
- pretrained Time Series Foundation Models (TSFMs).

It is important to state that experiments were implemented in Python using:

- `scikit-learn`[2];
- `xgboost`[3];
- `lightgbm`[4];
- `catboost`[5];
- `AutoGluon-TimeSeries`[6];
- `PyTorch`[7];
- `Transformers`[8].

[2] https://scikit-learn.org/
[3] https://xgboost.readthedocs.io/
[4] https://lightgbm.readthedocs.io/
[5] https://catboost.ai/
[6] https://auto.gluon.ai/
[7] https://pytorch.org/
[8] https://huggingface.co/docs/transformers/index

All supervised learning models were trained under a strictly chronological workflow in order to prevent temporal leakage. First, the harmonized environmental dataset was chronologically sorted by timestamp. Subsequently, feature selection and normalization operations were computed exclusively on the training subset before being applied to validation and testing observations. Finally, models were trained using only historical information available prior to the prediction horizon.

Figure 3.8 summarizes the complete machine learning workflow implemented in this thesis, from environmental data harmonization and temporal feature construction to model development, validation, and SO_2 prediction output.

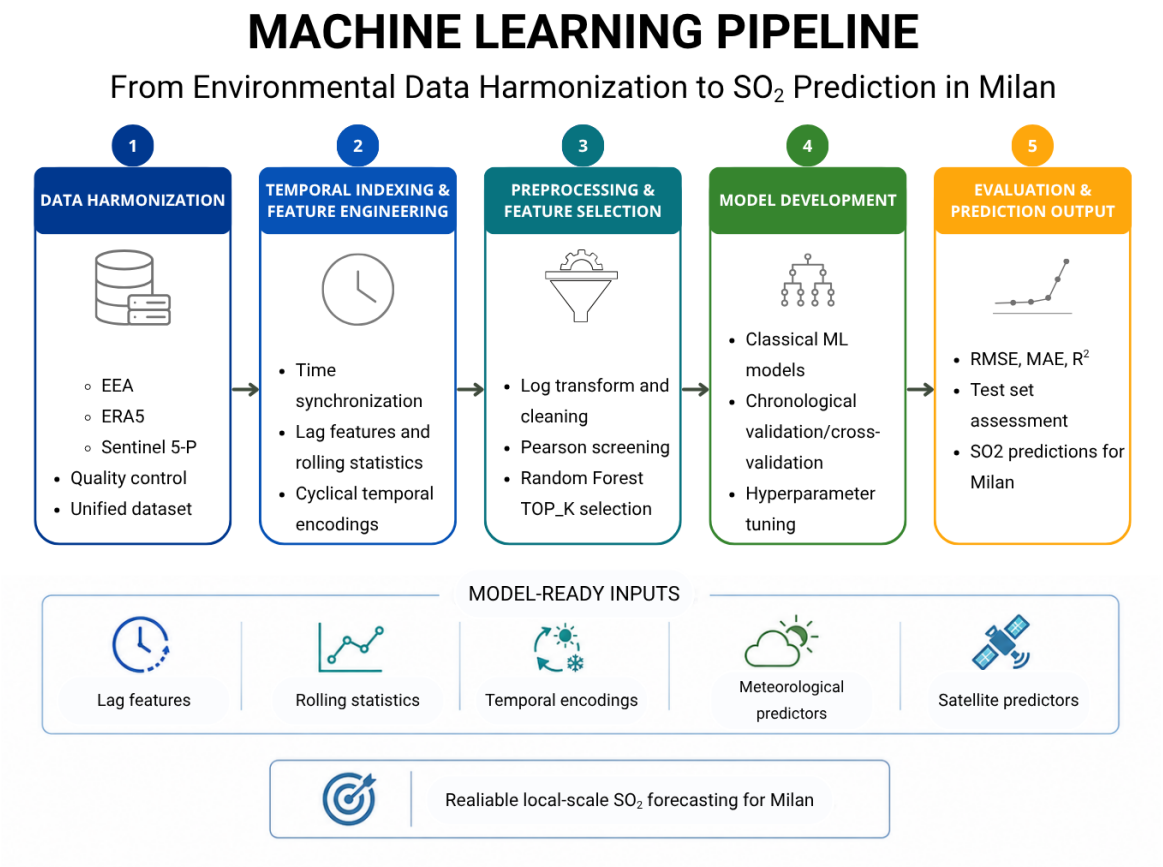

Figure 3.8: Compact machine learning pipeline implemented for daily SO_2 forecasting in the Milan Metropolitan Area. The workflow summarizes the transition from multi-source environmental data harmonization to temporal indexing, feature engineering, preprocessing, feature selection, model development, and final evaluation. The diagram also highlights the main model-ready inputs used in the forecasting experiments, including lag features, rolling statistics, temporal encodings, meteorological predictors, and satellite-derived atmospheric variables.

The evaluated machine learning families and benchmark roles are summarized in Table 3.7.

Machine Learning Model Families and Training Characteristics

Model	Family	Learning Strategy	Benchmark Role
Linear Regression	Parametric statistical regression	Closed-form linear optimization	Baseline interpretable benchmark
Random Forest	Bagging ensemble trees	Independent bootstrap aggregation	Nonlinear ensemble benchmark
XGBoost	Gradient-boosting ensemble	Sequential residual learning	High-performance boosting benchmark
LightGBM	Histogram-based boosting	Leaf-wise gradient optimization	Computationally efficient boosting benchmark
CatBoost	Ordered boosting ensemble	Permutation-based gradient boosting	Robust heterogeneous predictor benchmark

Table 3.7: Machine learning model families and training paradigms evaluated throughout the atmospheric forecasting experiments.

3.8.1. Linear Regression

Linear Regression was implemented using `sklearn.linear_model.LinearRegression`. The model belongs to the family of parametric statistical regression methods and was included as the principal interpretable baseline benchmark throughout the experimentation workflow. This model assumes a linear relationship between the engineered predictors and the target atmospheric concentration:

$$\hat{y} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j \tag{3.8}$$

where:

- $\hat{y}$ corresponds to the predicted SO_2 concentration;
- β_0 represents the intercept term;
- β_j are the fitted regression coefficients;

- x_j are the engineered atmospheric predictors.

On the other hand, the model training was performed through Ordinary Least Squares (OLS) optimization, where the regression coefficients are estimated by minimizing the residual sum of squared errors between predictions and observations.

Because linear regression is sensitive to feature magnitude differences, all predictors were normalized using `MinMaxScaler` fitted exclusively on the training subset. The final model complexity is directly associated with the number of fitted coefficients corresponding to the selected atmospheric predictors.

Although atmospheric systems involve highly nonlinear interactions between emissions, meteorology, atmospheric transport, and chemical transformation, Linear Regression was intentionally retained for three principal reasons:

- to provide an interpretable statistical baseline;
- to evaluate whether the engineered feature space contained predictive information;
- to establish comparison against nonlinear ensemble learning methods.

The model therefore serves as the lowest-complexity benchmark within the complete atmospheric forecasting framework [25].

3.8.2. Random Forest

Random Forest was implemented using `sklearn.ensemble.RandomForestRegressor`. The model belongs to the family of bagging-based ensemble learning algorithms and constructs multiple independent decision trees trained on bootstrap samples of the training data.

The final prediction is computed through ensemble averaging:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(x) \tag{3.9}$$

where:

- N is the number of decision trees;
- $T_i(x)$ corresponds to the prediction generated by the i-th tree.

Unlike linear models, Random Forest does not require explicit assumptions regarding:

- linearity;
- feature independence;
- Gaussian distributions;
- stationary relationships.

This characteristic is particularly important for atmospheric datasets, which frequently exhibit nonlinear dependencies, heterogeneous predictor interactions, threshold-driven atmospheric responses, and multicollinearity among meteorological variables.

Regarding the implemented configuration, it was applied:

- `n_estimators = 300`;
- `max_depth = None`;
- `random_state = 42`;
- `n_jobs = -1`.

A relatively large ensemble of 300 trees was selected to stabilize variance and improve robustness across heterogeneous atmospheric conditions. The unrestricted depth configuration allowed the model to learn highly nonlinear interactions between covariates.

Additionally, Random Forest served as the principal nonlinear benchmark for evaluating the effectiveness of the engineered atmospheric feature space. Furthermore, impurity-based feature importance derived from the ensemble was subsequently used during the TOP_K feature-selection stage because it provides robust ranking behavior under nonlinear predictor spaces [33].

3.8.3. XGBoost

XGBoost was implemented using `xgboost.XGBRegressor`. The model belongs to the family of gradient-boosted decision tree ensembles and constructs trees sequentially through residual-error minimization.

Unlike Random Forest, where trees are trained independently, XGBoost iteratively adds new trees designed to correct the prediction residuals generated by the previous ensemble.

The optimization objective follows:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.10}$$

where:

- $l(y_i, \hat{y}_i)$ represents the prediction loss;
- $\Omega(f_k)$ corresponds to the regularization term applied to tree f_k.

XGBoost was evaluated as the principal high-performance boosting benchmark because gradient-boosting methods consistently achieve strong performance on structured environmental datasets involving nonlinear atmospheric interactions [14, 32].

The implemented configuration used:

- `n_estimators = 500`;
- `learning_rate = 0.03`;
- `max_depth = 6`;
- `subsample = 0.8`;
- `colsample_bytree = 0.8`;
- `objective = reg:squarederror`.

The ensemble capacity was controlled primarily through the number of boosting iterations, tree depth, and residual-learning complexity.

A low learning rate combined with a relatively large number of boosting iterations was intentionally selected to improve gradient stability and reduce overfitting. Subsampling and feature-column sampling were additionally incorporated to improve generalization under correlated atmospheric predictors.

3.8.4. LightGBM

LightGBM was implemented using `lightgbm.LGBMRegressor`. The model belongs to the family of histogram-based gradient boosting algorithms and was designed to improve computational efficiency during large-scale boosting operations.

Different from conventional level-wise boosting approaches, LightGBM uses leaf-wise tree growth, where the branch producing the largest reduction in loss is expanded first.

This strategy enables efficient learning of localized nonlinear atmospheric behavior while reducing computational overhead during ensemble construction.

The implementation used:

- `n_estimators = 500`;

- `learning_rate = 0.03`;
- `num_leaves = 31`;
- `max_depth = -1`;
- `subsample = 0.8`;
- `colsample_bytree = 0.8`.

A value of `num_leaves = 31` was selected to balance nonlinear flexibility and overfitting control under relatively limited atmospheric datasets.

LightGBM was particularly suitable for the heterogeneous predictor space constructed throughout this work because the feature matrix simultaneously contained the autoregressive features, the rolling statistics, the cyclical temporal encoding and the meteorological variables as well as the satellite retrieving. The leaf-wise optimization strategy enables efficient partitioning of this multidimensional atmospheric feature space while maintaining comparatively low computational cost.

3.8.5. CatBoost

CatBoost was implemented using `catboost.CatBoostRegressor`. The model belongs to the family of ordered gradient-boosting ensemble methods and incorporates permutation-driven training procedures designed to reduce prediction shift and overfitting during sequential boosting.

Although CatBoost was originally developed for categorical feature processing, it also performs robustly on numerical environmental datasets characterized by nonlinear atmospheric dependencies and heterogeneous predictor distributions [34].

The implemented configuration used:

- `iterations = 500`;
- `learning_rate = 0.03`;
- `depth = 6`;
- `loss_function = RMSE`;
- `verbose = False`.

The model capacity was controlled through the number of boosting iterations, controlling the symmetric tree depth, and by ordered boosting complexity. Likewise, it was trained

using the same chronological splits and feature-selected predictor space adopted throughout the remaining supervised learning experiments in order to preserve methodological consistency [32].

3.8.6. Time Series Foundation Models (TSFMs)

In addition to the classical machine learning models described previously, this research evaluated pretrained Time Series Foundation Models (TSFMs) for atmospheric SO_2 forecasting. TSFMs differ from conventional supervised regression models because they are pretrained on large collections of heterogeneous time series and can generate forecasts for unseen temporal sequences without being trained from scratch on the target dataset [8].

The objective of this stage was to evaluate whether general temporal representations learned during large-scale pretraining can transfer to atmospheric pollutant forecasting, and whether their performance improves when explicit environmental covariates and domain-adaptation strategies are introduced.

The TSFM experiments followed three progressively more complex configurations as evidenced in Figure 3.9:

1. zero-shot forecasting using only the historical SO_2 sequence;
2. covariate-aware forecasting using engineered atmospheric predictors;
3. parameter-efficient domain adaptation through LoRA fine-tuning.

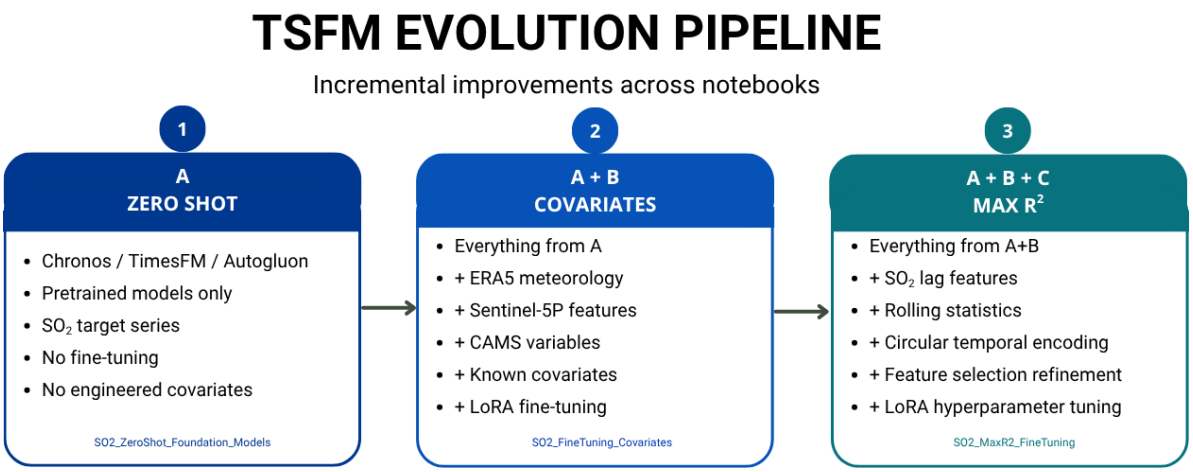

Figure 3.9: Time Series Foundation Model workflow used for daily SO_2 forecasting. The pipeline illustrates the progression from zero-shot forecasting with pretrained models to covariate-based configurations, AutoGluon forecasting, and Chronos-2 LoRA fine-tuning. The workflow evaluates whether progressively adding environmental covariates and lightweight model adaptation improves forecasting performance in the Milan SO_2 case study.

This progressive design enabled the evaluation of three methodological questions: whether pretrained models can forecast SO_2 without local training, whether meteorological and satellite covariates improve forecasting skill, and whether lightweight fine-tuning can adapt a general-purpose TSFM to the Milan atmospheric domain.

Model Sources, Pretraining Data, and Benchmark Role

The pretrained models used in this research were obtained from public model repositories, principally Hugging Face and the AutoGluon-TimeSeries model interface. Chronos models are pretrained on large collections of real and synthetic time series, including publicly available datasets and synthetic Gaussian-process-generated series, but the available documentation does not indicate that the pretraining corpus was specifically constructed from atmospheric pollutant observations [2]. Similarly, TimesFM was pretrained on a large time-series corpus containing approximately 100 billion time points from diverse domains, but pollutant-specific pretraining is not explicitly reported [8].

Consequently, the zero-shot experiments should be interpreted as a transfer-learning test: the models were not originally trained specifically for SO_2, NO_2, or air-quality forecasting, but were evaluated to determine whether their general temporal representations could generalize to atmospheric pollutant dynamics.

The foundation models and their experimental roles are summarized in Table 3.8.

Time Series Foundation Models Evaluated in This Research

Model	Model Family	Pretraining Data	Role in This Research
Chronos-Bolt-Small	T5-based encoder–decoder TSFM	Large real and synthetic time-series corpus; no pollutant-specific pretraining reported	Zero-shot benchmark using only historical SO_2.
TimesFM-1.0-200m	Decoder-only patched time-series foundation model	Approximately 100 billion time points from diverse real-world and synthetic sources	Additional zero-shot benchmark using only historical SO_2.
AutoGluon-TimeSeries	Automated forecasting framework	Not a single pretrained model; trains and ensembles multiple forecasting models	Covariate-aware forecasting framework using engineered atmospheric predictors.
Chronos-2	Universal TSFM with covariate support	Large-scale temporal pretraining; accessed through AutoGluon and Chronos interfaces	Covariate-aware forecasting and LoRA fine-tuning.

Table 3.8: Foundation forecasting models and AutoML framework used during the TSFM experimental stage.

Chronological Splitting Strategy

All TSFM experiments used strict chronological splitting instead of random train–test partitioning. This decision was necessary because atmospheric forecasting is a temporal prediction task, and random splitting would allow future atmospheric information to influence model training or preprocessing.

The complete SO_2 series was partitioned chronologically into:

- 64% training;
- 16% validation;
- 20% testing.

The temporal order was preserved as:

$$\texttt{train} \rightarrow \texttt{validation} \rightarrow \texttt{test}$$

without shuffling.

For zero-shot experiments, the forecasting context consisted of the concatenated training and validation periods:

$$\texttt{context} = \texttt{train + validation}$$

while the prediction horizon corresponded to the complete unseen test period. This design simulated realistic forecasting conditions in which only past observations are available at prediction time.

3.8.7. Zero-Shot Foundation Models

The first TSFM configuration evaluated pretrained models without any domain-specific retraining, covariates, or fine-tuning. In this setup, the models received only the historical SO_2 concentration sequence:

$$x = [SO_2(t-n), ..., SO_2(t-1)]$$

and generated forecasts for the test horizon.

No meteorological variables, satellite observations, lag-engineered predictors, or temporal encodings were supplied. Therefore, this experiment isolated the intrinsic zero-shot forecasting capability of the pretrained temporal models.

Chronos-Bolt-Small

Chronos-Bolt-Small was implemented using the `chronos`, `torch`, and `transformers` libraries. The pretrained checkpoint was loaded from Hugging Face using:

`amazon/chronos-bolt-small`

Chronos-Bolt belongs to the Chronos family of pretrained probabilistic forecasting models. The Chronos framework tokenizes continuous time-series values through scaling and quantization and trains transformer-based architectures to model temporal sequences as

token prediction problems [2]. Chronos-Bolt is a more computationally efficient variant designed for faster inference while maintaining strong zero-shot forecasting performance.

The implementation used:

- `ChronosBoltPipeline.from_pretrained()`;
- `device_map = "cpu"`;
- `torch_dtype = torch.float32`;
- `prediction_length = len(test)`;
- `quantile_levels = [0.1, 0.5, 0.9]`.

The quantile configuration enabled probabilistic forecasting by generating lower, median, and upper prediction intervals. The mean forecast was used as the principal deterministic prediction for evaluation.

Because Chronos-Bolt-Small was used in zero-shot mode, no model weights were updated using the Milan SO_2 dataset. Therefore, the model was not trained on pollutant data during this research; it was only applied to the historical SO_2 sequence at inference time.

TimesFM-1.0

TimesFM was evaluated as a second zero-shot foundation model benchmark using the PyTorch backend. The pretrained checkpoint was loaded from Hugging Face as:

`google/timesfm-1.0-200m-pytorch`

TimesFM is a decoder-only time-series foundation model based on patched temporal input representations. Instead of processing each timestep independently, the model groups temporal observations into patches and learns long-range forecasting patterns through a decoder-style attention architecture [8].

The implementation used:

- `backend = "pytorch"`;
- `per_core_batch_size = 32`;
- `horizon_len = prediction_length`;
- `freq = [1]`.

The frequency code `freq = [1]` was used because the input SO_2 series was daily, corresponding to a low-frequency temporal sequence.

As with Chronos-Bolt-Small, TimesFM was not fine-tuned and did not use meteorological or satellite covariates. It therefore served as an additional zero-shot benchmark for evaluating whether general-purpose temporal pretraining could transfer directly to atmospheric pollutant forecasting.

3.8.8. AutoGluon Forecasting with Atmospheric Covariates

After evaluating pure zero-shot forecasting, the second experimental stage introduced engineered atmospheric covariates using AutoGluon-TimeSeries. Unlike Chronos-Bolt-Small and TimesFM zero-shot inference, AutoGluon was used as an automated forecasting framework capable of training, validating, ranking, and ensembling multiple time-series models.

The objective of this stage was to determine whether explicit atmospheric context improves forecasting performance relative to history-only TSFM inference.

The covariate-aware configuration integrated:

- SO_2 lag features;
- rolling SO_2 statistics;
- cyclical temporal encodings;
- ERA5 meteorological variables;
- Sentinel-5P atmospheric column observations.

AutoGluon was implemented through:

`TimeSeriesPredictor`

with the dataset converted into the long-format structure required by the framework:

- `item_id`;
- `timestamp`;
- `target`;
- `known_covariates`.

Because AutoGluon requires a regular temporal index, missing dates were reconstructed using:

```
pd.date_range(freq = "D")
```

and interpolated using:

```
interpolate(method = "time")
```

before constructing the `TimeSeriesDataFrame`. This regularization step was applied only within the AutoGluon forecasting branch.

The AutoGluon forecasting configuration is summarized in Table 3.9.

AutoGluon Forecasting Configuration

Parameter	Configuration
`freq`	`"D"`, indicating daily temporal observations.
`prediction_length`	Forecasting horizon defined according to the experimental setup.
`eval_metric`	`"RMSE"`, selected to penalize large forecasting errors.
`presets`	`"medium_quality"` for computationally feasible automated model training.
`known_covariates_names`	List of engineered atmospheric, meteorological, temporal, and satellite predictors.

Table 3.9: AutoGluon-TimeSeries configuration used for covariate-aware atmospheric forecasting.

The known covariates supplied to AutoGluon are summarized in Table 3.10.

Known Covariates Used in AutoGluon and Chronos-2 Experiments

Covariate Group	Variables
SO_2 lag features	$t-1$, $t-2$, $t-3$, $t-7$, $t-14$.
Rolling statistics	7-day mean, 7-day standard deviation, and 14-day mean.
Cyclical temporal variables	`month_sin`, `month_cos`, `doy_sin`, `doy_cos`.
Meteorological variables	Temperature, precipitation, boundary layer height, wind direction, wind components, surface pressure, and radiation variables.
Sentinel-5P atmospheric columns	NO_2, SO_2, and CO column retrievals.

Table 3.10: Atmospheric covariates supplied to the covariate-aware forecasting models.

3.8.9. Chronos-2 with Covariates

Chronos-2 was evaluated as a covariate-aware TSFM through the AutoGluon-TimeSeries interface. Unlike the zero-shot Chronos-Bolt and TimesFM experiments, this configuration allowed the model to condition its forecasts on engineered atmospheric predictors.

Chronos-2 was used to evaluate whether a pretrained foundation model can exploit external environmental information without requiring full model retraining. The covariates were supplied as known covariates during the forecasting horizon, allowing the model to condition predictions on atmospheric states rather than relying exclusively on SO_2 history.

This configuration represents an intermediate stage between pure zero-shot inference and domain-adapted fine-tuning.

3.8.10. LoRA Fine-Tuning

The final experimental stage introduced parameter-efficient fine-tuning of Chronos-2 through Low-Rank Adaptation (LoRA). Although pretrained TSFMs contain generalized temporal representations, these representations are not specifically optimized for pollutant dynamics in the Milan Metropolitan Area. LoRA fine-tuning was therefore implemented to adapt the model to local SO_2 behavior while avoiding the computational cost of full transformer retraining.

Instead of updating all pretrained model parameters, LoRA injects trainable low-rank

matrices into selected transformer layers:

$$W' = W + BA \tag{3.11}$$

where:

- W is the original pretrained weight matrix;
- A and B are trainable low-rank matrices.

This strategy substantially reduces the number of trainable parameters while preserving the pretrained temporal representations.

The LoRA configuration used in this research was:

- `fine_tune = True`;
- `fine_tune_mode = "lora"`;
- `fine_tune_steps = 500`;
- `fine_tune_lr = 5e-5`;
- `fine_tune_batch_size = 8`.
- `lora_rank` = 8;
- `lora_alpha` = 16.

A low learning rate was selected to reduce the risk of catastrophic forgetting, while the limited number of update steps constrained overfitting under the available atmospheric training data.

Unlike the zero-shot experiments, the LoRA stage did train model adaptation parameters using the Milan SO_2 training and validation data together with the engineered atmospheric covariates. However, the original pretrained Chronos-2 weights were not fully updated; only the low-rank adaptation matrices were optimized.

This final configuration therefore represents the most domain-adapted TSFM pipeline implemented in this research.

3.9. Training Strategy

The training strategy was designed to ensure methodological reproducibility, temporal consistency, and realistic atmospheric forecasting conditions across all ML and TSFM

experiments. Because atmospheric pollutant concentrations exhibit strong temporal autocorrelation and seasonal structure, random train–test partitioning would introduce temporal leakage and produce unrealistic forecasting performance estimates.

Consequently, all experiments implemented strict chronological partitioning together with temporally consistent preprocessing and validation procedures.

Figure 3.10 presents the complete training and evaluation workflow.

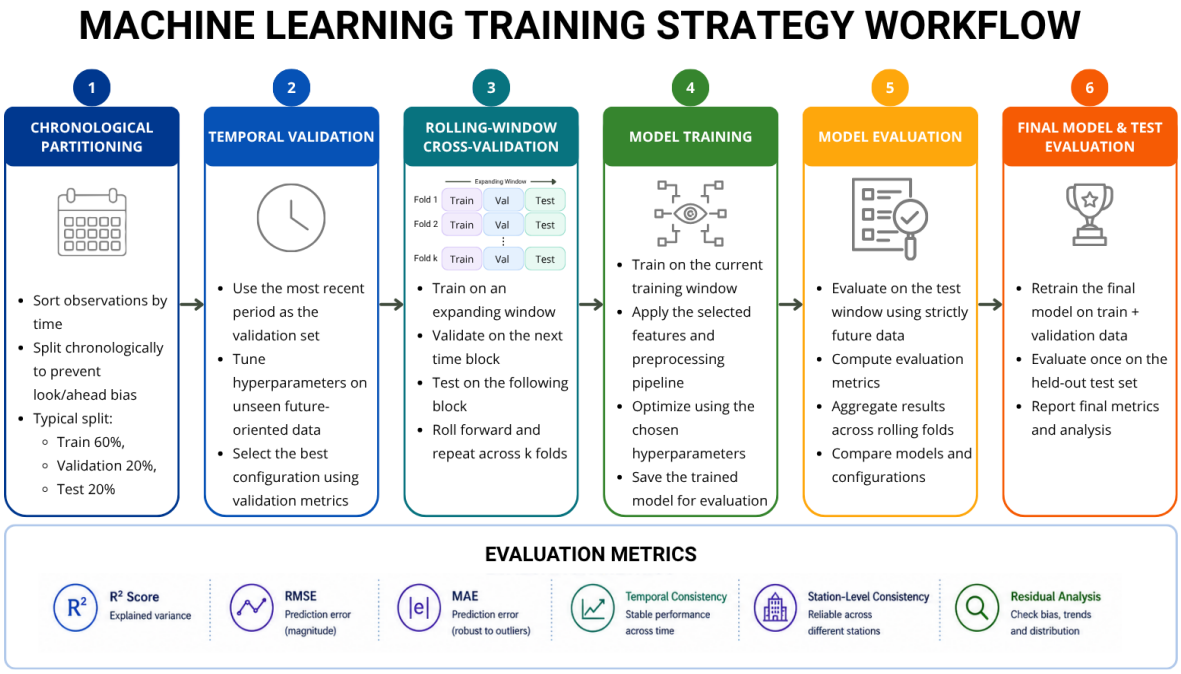

Figure 3.10: Training strategy workflow including chronological partitioning, temporal validation, rolling-window evaluation, and final testing procedures.

3.9.1. Chronological Train–Validation–Test Split

All datasets were partitioned chronologically in order to preserve temporal causality and simulate realistic atmospheric forecasting conditions.

The complete time series was divided into:

- 64% training subset;
- 16% validation subset;
- 20% testing subset.

The partitioning structure followed:

$$\texttt{train} \rightarrow \texttt{validation} \rightarrow \texttt{test}$$

without temporal shuffling.

The training subset was used for:

- model fitting;
- feature selection;
- normalization fitting;
- hyperparameter adjustment.

The validation subset was used for:

- intermediate model evaluation;
- configuration comparison;
- TSFM adaptation monitoring.

Finally, the testing subset was reserved exclusively for final unseen evaluation.

No testing observations were used during:

- feature engineering optimization;
- feature selection;
- scaler fitting;
- hyperparameter tuning;
- LoRA fine-tuning.

This chronological strategy reproduces operational forecasting conditions where models are trained exclusively on historical atmospheric information before being applied to future observations.

The partitioning workflow was implemented using:

- timestamp ordering through `sort_values()`;
- chronological indexing with `pandas.iloc()`.

3.9.2. Temporal Leakage Prevention

A fundamental methodological requirement throughout the entire workflow was the prevention of temporal leakage. Because atmospheric pollutant concentrations exhibit strong temporal persistence and seasonality, the inadvertent use of future information during preprocessing, feature engineering, model training, or evaluation can artificially inflate forecasting performance and produce unrealistic results.

To preserve realistic forecasting conditions, all preprocessing operations that required statistical estimation were performed exclusively on the training subset. Normalization parameters, feature-selection procedures, and model-specific transformations were fitted using only training observations and subsequently applied unchanged to the validation and testing subsets. Likewise, lag features and rolling-window statistics were constructed solely from observations preceding each prediction timestamp, ensuring that no future information was incorporated into the predictor space.

Chronological ordering was maintained throughout the complete experimental workflow. Training, validation, and testing subsets were generated using temporally ordered partitions, and all model families, including classical machine learning approaches, AutoML frameworks, and Time Series Foundation Models (TSFMs), were evaluated exclusively on future observations not available during training. This strategy guarantees temporal causality and provides a realistic assessment of forecasting performance under operational atmospheric monitoring conditions.

3.9.3. Temporal Cross-Validation and Robustness Analysis

In addition to the fixed chronological partitioning strategy, temporal cross-validation was explored during intermediate model development and robustness analysis using:

- `sklearn.model_selection.TimeSeriesSplit`

Unlike conventional random K-Fold validation, this approach preserves chronological consistency while evaluating the models across multiple temporal forecasting windows.

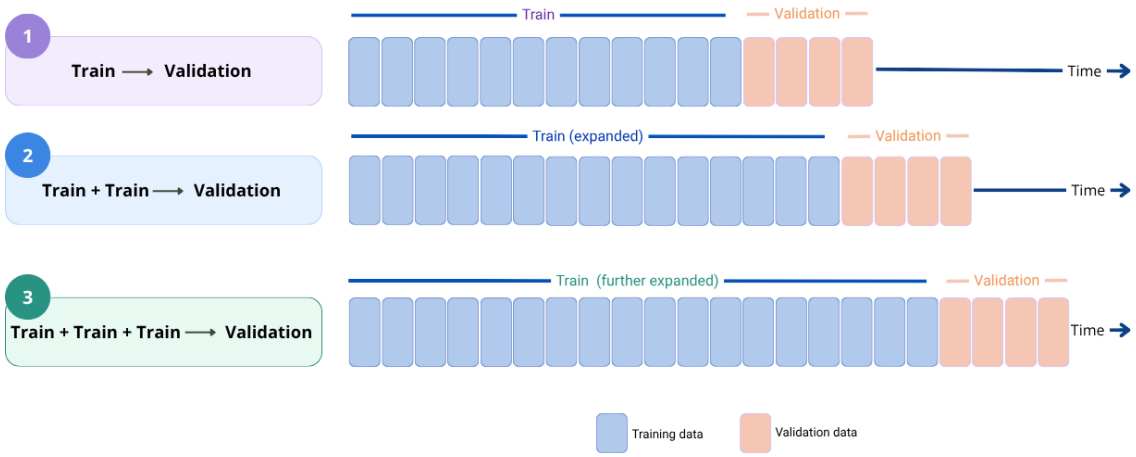

Figure 3.11: Expanding-window temporal cross-validation strategy used during robustness analysis. The training subset progressively expands while validation windows advance chronologically, preserving temporal causality and preventing future information leakage.

The expanding-window structure progressively increases the amount of historical data available for training while moving the validation subset forward chronologically across multiple atmospheric regimes.

This strategy is particularly relevant for atmospheric forecasting because pollutant dynamics vary substantially across seasons, meteorological conditions, and emission patterns [26]. Evaluating the models across multiple temporal windows therefore improves robustness assessment and reduces sensitivity to isolated atmospheric periods.

The implementation used `n_splits = 5` during intermediate robustness analysis and configuration testing. However, the final reported results throughout this thesis were always computed exclusively on the held-out testing subset.

3.9.4. Training Workflow per Model Family

Although all models followed the same chronological partitioning strategy, the training workflow differed according to the model family.

For:

- Linear Regression;
- Random Forest;
- XGBoost;
- LightGBM;

- CatBoost;

following the same workflow mentioned previously in the chapter. Nonetheless, AutoGluon and TSFM experiments used sequence-oriented forecasting workflows operating on temporally indexed atmospheric series.

For AutoGluon forecasting the dataset was converted into long-format forecasting structures, the atmospheric covariates were supplied through `known_covariates`, and the validation forecasting windows were internally generated by the framework.

While for Chronos and TimesFM, the forecasting context windows were extracted chronologically, the models operated autoregressively on historical SO_2 sequences, and the forecasts were generated sequentially across the prediction horizon.

Finally, the LoRA experiments introduced an additional adaptation stage where the pretrained transformer representations were partially updated using only the training subset before evaluating the unseen testing period.

3.9.5. Reproducibility and Random Seed Control

To ensure reproducibility across all experiments, fixed random seeds were applied whenever stochastic training operations were involved.

The implementation used:

- `random_state = 42`

for:

- Random Forest;
- XGBoost;
- LightGBM;
- train-validation procedures.

Additional seed initialization was applied whenever supported by the corresponding frameworks, including:

- NumPy;
- PyTorch;
- AutoGluon.

This reproducibility strategy ensures that differences between experiments originate primarily from methodological modifications rather than stochastic training variability.

3.10. Evaluation Metrics

The predictive performance of the machine learning models and Time Series Foundation Models (TSFMs) was evaluated using multiple complementary regression metrics designed to quantify forecasting accuracy, variance explanation capability, and normalized prediction error under heterogeneous atmospheric conditions.

Because of the nature of atmospheric pollutants forecasting involving strong temporal variability, seasonal fluctuations, and episodic concentration spikes; relying on a single evaluation metric could produce incomplete interpretation of model behavior. For this reason, the evaluation framework combined both absolute-error and normalized-error metrics in order to assess forecasting robustness under different atmospheric regimes.

The following metrics were computed throughout the experimental analysis:

Performance Metrics Stored in Model Leaderboards

Metric	Description
RMSE	Root Mean Squared Error. Measures the average magnitude of prediction errors while assigning greater weight to large deviations.
MAE	Mean Absolute Error. Represents the average absolute difference between observed and predicted SO_2 concentrations.
R^2	Coefficient of determination. Quantifies the proportion of variance in the observed SO_2 series explained by the model.
NRMSE(IQR)	Normalized Root Mean Squared Error using the interquartile range (IQR) of the observations as the normalization factor, enabling scale-independent comparison.
NRMSE(std)	Normalized Root Mean Squared Error using the standard deviation of the observations as the normalization factor.

Table 3.11: Performance metrics recorded in the model leaderboard files and used for comparative evaluation across all experiments.

All metrics were calculated exclusively on the held-out testing subset after model training, feature selection, and hyperparameter optimization had been completed.

3.10.1. Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) measures the square root of the mean squared prediction residuals:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.12}$$

where:

- y_i represents the observed SO_2 concentration;
- $\hat{y}_i$ corresponds to the predicted concentration;

- n is the total number of testing observations.

RMSE was selected as one of the principal evaluation metrics because the squared residual term penalizes large forecasting errors more strongly than linear-error metrics. This property is particularly important for atmospheric pollutant estimation since severe pollution episodes are environmentally relevant and operationally critical.

Additionally, RMSE preserves the physical units of the target variable ($\mu g\ m^{-3}$), facilitating direct interpretation of forecasting errors.

3.10.2. Normalized RMSE using Interquartile Range (NRMSE(IQR))

Although RMSE provides physically interpretable forecasting errors, direct RMSE comparison between datasets with different concentration variability may be misleading [11].

To address this limitation, a normalized RMSE metric based on the interquartile range (IQR) was additionally computed:

$$NRMSE_{IQR} = \frac{RMSE}{Q_{75} - Q_{25}} \tag{3.13}$$

where:

- Q_{75} corresponds to the 75th percentile of the observed series;
- Q_{25} corresponds to the 25th percentile.

The interquartile-range normalization reduces sensitivity to extreme outliers and provides a robust relative error estimate under skewed pollutant distributions.

This metric is particularly useful for atmospheric datasets because pollutant concentrations frequently exhibit:

- non-Gaussian distributions;
- episodic spikes;
- seasonal asymmetry;
- highly variable concentration ranges.

Consequently, NRMSE(IQR) allows more stable comparison between forecasting models operating under heterogeneous atmospheric conditions.

3.10.3. NRMSE using Standard Deviation (NRMSE(std))

A second normalized error metric was computed using the standard deviation of the observed pollutant series:

$$NRMSE_{std} = \frac{RMSE}{\sigma_y} \tag{3.14}$$

where:

- σ_y represents the standard deviation of the observed SO_2 concentrations.

NRMSE(std) expresses the prediction error relative to the intrinsic variability of the atmospheric series. Lower values indicate that the forecasting error remains small compared to the natural variability of the pollutant concentrations.

Unlike IQR normalization, standard deviation normalization is more sensitive to extreme atmospheric events because the variance term incorporates the complete distribution of the series.

Using both NRMSE(IQR) and NRMSE(std) provides complementary interpretation of forecasting robustness under highly variable atmospheric conditions.

It is important to state that the normalized RMSE metrics were computed manually after calculating the corresponding RMSE values and statistical descriptors from the testing subset. Therefore, all reported metrics correspond exclusively to unseen testing observations and were never computed using the training data.

3.11. Computational Environment and Implementation Setup

This chapter describes the computational environment, software ecosystem, architectural design, and execution workflow adopted throughout the SO_2 forecasting experiments. Unlike the previous sections in the chapter, which focuses on the scientific rationale and methodological principles of the research, the present section documents the practical set up details required to reproduce the experimental framework.

The implementation evolved progressively through six successive notebook versions, each addressing methodological limitations identified in the previous iteration. This iterative development strategy enabled progressive refinement of:

- preprocessing procedures;
- feature engineering strategies;
- evaluation methodologies;
- machine learning architectures;
- Time Series Foundation Model (TSFM) integration.

The complete implementation developed during this research is publicly available through dedicated repositories covering environmental harmonization [28], European-scale statistical analysis [29], and machine learning forecasting experiments [30].Likewise, the workflow, software dependencies, computational constraints, and automation procedures described in this chapter correspond to the complete experimental framework used throughout the thesis.

3.12. Hardware

All experiments were conducted on a single personal workstation running Microsoft Windows 11. The computational environment consisted of:

- Intel Core i7 CPU;
- 16 GB RAM;
- Solid-State Drive (SSD) storage.

No dedicated Graphics Processing Unit (GPU) was available during the experiments. Consequently, all machine learning training, transformer inference, and fine-tuning procedures were executed exclusively on CPU hardware. This constraint significantly influenced several architectural and methodological decisions across the experimental workflow.

In particular, the absence of GPU acceleration affected:

- transformer batch sizes;
- fine-tuning duration;
- forecasting horizon length;
- AutoGluon preset selection;
- model complexity budgets.

The Chronos-Bolt-Small and TimesFM foundation models were, therefore, executed using CPU inference mode. Although inference remained computationally feasible for the selected forecasting horizons, larger transformer configurations and extended fine-tuning schedules would not be practical without hardware acceleration.

Similarly, LoRA fine-tuning experiments were intentionally constrained to:

- 200–500 update steps;
- micro-batch sizes of 8;
- low learning rates.

These restrictions were necessary to maintain acceptable execution times under CPU-only training conditions.

External network connectivity was additionally required during multiple stages of the workflow, including:

- EEA air quality retrieval;
- ERA5 reanalysis download;
- Sentinel-5P extraction through Google Earth Engine;
- Hugging Face pretrained model retrieval.

To reduce repeated bandwidth consumption and improve reproducibility, all downloaded datasets and pretrained model weights were cached locally after their initial retrieval.

3.13. Software Environment

3.13.1. Python Environment

Python 3.10 was used as the principal programming language throughout the implementation. The entire workflow was developed using Jupyter Notebook environments in order to facilitate iterative experimentation, modular pipeline development, intermediate result inspection, and reproducible notebook-based execution.

Two independent Conda virtual environments were maintained to isolate incompatible dependency trees and avoid runtime conflicts between TensorFlow and PyTorch libraries on Windows systems.

The first environment:

- `cmcc_env`

was used for the preprocessing and first machine learning versions. It contained:

- scikit-learn;
- TensorFlow/Keras;
- XGBoost;
- CatBoost;
- SHAP.

The second environment:

- `so2_foundation`

was used for the TSFMs versions, containing:

- AutoGluon-TimeSeries;
- Chronos;
- TimesFM;
- PyTorch.

The separation between environments was required because simultaneous TensorFlow and PyTorch installation under Windows generated Dynamic-Link Library (DLL) conflicts during AutoGluon initialization. Maintaining isolated environments resolved these compatibility issues while preserving reproducibility across all experimental versions.

3.13.2. Libraries and Frameworks

The implementation relied on a heterogeneous Python ecosystem combining:

- numerical computing;
- geospatial processing;
- atmospheric data extraction;
- machine learning;
- transformer forecasting;
- visualization;
- model explainability.

Table 3.12 summarizes the principal libraries used throughout the implementation.

Principal Software Libraries Used Throughout the Implementation Workflow

Library	Role
pandas	Tabular preprocessing and aggregation
numpy	Numerical computation
scikit-learn	Classical machine learning
tensorflow / keras	Deep learning models
xgboost	Gradient boosting
catboost	Ordered boosting
autogluon.timeseries	Time-series AutoML
chronos	Foundation forecasting models
timesfm	Decoder-only TSFM forecasting
torch	Transformer backend
geopandas	Spatial processing
xarray	ERA5 NetCDF processing
earthengine-api	Sentinel-5P extraction
cdsapi	ERA5 download

Table 3.12: Principal software libraries and frameworks used throughout the environmental preprocessing and machine learning implementation workflow.

The implementation additionally relied on:

- SHAP explainability analysis;
- Plotly interactive visualizations;
- matplotlib static charting;
- joblib model serialization;
- pathlib filesystem abstraction.

The full software inventory together with minimum compatible versions is documented in the implementation appendix.

3.14. Pipeline Architecture

The SO_2 prediction framework was implemented as a progressive sequence of six independent Jupyter notebook versions rather than as a monolithic software system. Each notebook represented a distinct architectural iteration focused on addressing limitations identified during the previous experimental stage.

The iterative pipeline evolved through the following improvements:

1. baseline machine learning pipeline;
2. improved temporal ML framework;
3. enhanced feature engineering;
4. zero-shot foundation models;
5. covariate-aware TSFMs;
6. maximum-R^2 fine-tuning architecture.

Despite the progressive architectural modifications, all notebook versions shared a common preprocessing backbone consisting of:

- CSV ingestion;
- timestamp parsing;
- circular wind preprocessing;
- daily aggregation;
- feature engineering;
- chronological partitioning;
- model training;
- leaderboard export.

The raw dataset contained:

- EEA SO_2 observations;
- ERA5 meteorological variables;
- CAMS reanalysis fields;
- Sentinel-5P atmospheric columns;

- geographic metadata.

After loading, the dataset underwent deterministic preprocessing using:

- circular wind averaging;
- station aggregation;
- duplicate removal;
- temporal alignment;
- feature generation.

The preprocessing workflow remained intentionally deterministic across all versions to ensure reproducibility and isolate the effects of model and feature-engineering modifications.

3.14.1. Source Code Availability

To support methodological transparency and reproducibility, the software developed throughout this research has been released through publicly accessible GitHub repositories.

The repositories are organized according to the principal components of the framework:

- **EEA-AQA**: Environmental harmonization and metadata reconstruction framework for the European Environment Agency air-quality monitoring network, including station metadata processing, pollutant filtering, spatial harmonization, and environmental dataset integration.

  `https://github.com/Saudisis/EEA-AQA`

- **EEA_Stats**: European-scale statistical analysis framework used for exploratory analysis, monitoring-network characterization, pollutant distributions, station typologies, and environmental variability assessment.

  `https://github.com/Saudisis/EEA_Stats`

- **SO2_Milan_Prediction**: Machine learning and Time Series Foundation Model experimentation framework used for feature engineering, model training, AutoML evaluation, Chronos adaptation, LoRA fine-tuning, forecasting experiments, and performance benchmarking.

  `https://github.com/Saudisis/SO2_Milan_Prediction`

Together, these repositories contain the preprocessing workflows, environmental data-fusion procedures, feature-engineering pipelines, forecasting experiments, and evaluation

routines required to reproduce the methodological framework and predictive analyses presented in this thesis.

3.15. Automation Workflow

Although the pipeline was not deployed through a fully automated orchestration framework, a structured execution protocol was implemented in order to maintain reproducibility and artifact traceability across all notebook versions.

The workflow generated three principal categories of artifacts:

- data artifacts;
- model artifacts;
- results artifacts.

3.15.1. Data Artifacts

The complete environmental dataset:

- `milano-gee-cams-SO2-df_complete.csv`

was produced by the companion environmental preprocessing pipeline and was consumed consistently in all six modeling notebooks.

Dataset paths were centralized using configurable variables such as:

- `DATA_PATH`
- `BASE_PATH`

allowing rapid reconfiguration of input sources without modifying downstream logic.

Intermediate preprocessing outputs, including aggregated daily datasets, station-level datasets, and transformed feature matrices, were saved using timestamped filenames in order to prevent accidental overwriting, preserve historical experimental states, and recover from interrupted executions.

3.15.2. Model Artifacts

Classical machine learning models from the first versions were serialized using:

- `joblib.dump()`

allowing direct reload without retraining. Feature-selection outputs and normalization parameters were additionally exported as:

- JSON files;
- serialized scaler objects.

On the other hand, AutoGluon predictors internally maintained trained weights, forecasting logs, leaderboard rankings, and model metadata.

These directories were preserved automatically and could be reloaded through:

- `TimeSeriesPredictor.load()`

for reproducible inference.

3.15.3. Results Artifacts

Every notebook exported:

- leaderboard CSV files 3.11
- prediction-versus-truth plots;
- residual diagnostics;
- model comparison graphics.

The exported artifacts enabled direct cross-version comparison of:

- feature engineering strategies;
- model architectures;
- transformer adaptation approaches;
- covariate integration methods.

4 | Results

This chapter presents the results obtained from the six-version environmental machine learning framework developed for daily SO_2 forecasting in the Milan Metropolitan Area. The results follow the chronological progression of the modeling pipeline described in Chapters 3 and 4.

Although analogous exploratory and modeling procedures were also tested for NO_2, the detailed presentation of results in this chapter focuses on SO_2. This decision was made because SO_2 constitutes the principal pollutant case study used to evaluate the complete machine learning and Time Series Foundation Model workflow, including feature engineering, chronological validation, covariate integration, AutoML experimentation, and Chronos-2 fine-tuning. The NO_2 analyses were therefore treated as supplementary exploratory outputs and are not discussed in detail in order to maintain the methodological focus and avoid introducing a parallel results stream.

The chapter begins with an exploratory analysis of the final SO_2 environmental dataset, including the temporal behavior of the target variable, seasonal variability, target distribution, and missing-data treatment. It then presents the feature-selection and feature-importance results obtained across the different experimental versions. Subsequently, the performance of classical machine learning models, ensemble strategies, AutoML configurations, and Time Series Foundation Models (TSFMs) is evaluated. The final sections present the results obtained through covariate integration and LoRA fine-tuning of Chronos-2, followed by an overall comparison of the main model families.

4.1. Dataset Exploration

4.1.1. Dataset Overview

The final station-aggregated dataset contained 992 daily observations spanning February 2019 to October 2023. The dataset integrated EEA ground-level SO_2 measurements, ERA5 meteorological variables, CAMS atmospheric reanalysis products, and Sentinel-5P atmospheric column observations.

After spatial aggregation of all monitoring stations within the Milan Metropolitan Area, the resulting daily SO_2 series presented the following descriptive statistics:

- mean concentration: 2.50 $\mu g/m^3$;
- median concentration: 2.40 $\mu g/m^3$;
- standard deviation: 0.83 $\mu g/m^3$;
- minimum concentration: 0.025 $\mu g/m^3$;
- maximum concentration: 6.38 $\mu g/m^3$.

These values show that the aggregated SO_2 series was characterized by relatively low daily concentrations, moderate variability, and occasional high-concentration episodes. The difference between the mean and median also indicates a moderately right-skewed distribution, which was later addressed through target transformation in Versions 2 and 3.

The chronological train, validation, and test split preserved similar statistical characteristics across the three subsets. This indicates that the test period did not correspond to an anomalous atmospheric regime when compared with the historical record. Therefore, the evaluation strategy provided a realistic assessment of forecasting performance, since the models were tested on future observations while maintaining comparable SO_2 concentration distributions between training and testing conditions.

4.1.2. Temporal Evolution of SO_2

The complete daily SO_2 time series exhibited substantial temporal variability across the study period, as shown in Figure 4.1. The signal was characterized by short-term fluctuations, seasonal variability, and episodic concentration peaks rather than by a stable monotonic increasing or decreasing trend.

The long-term mean concentration is represented by the horizontal dashed line in Figure 4.1. Several periods exceeded this reference level, indicating intermittent episodes of elevated SO_2 concentration. In particular, the winter 2019–2020 interval, highlighted in the figure, presented concentrations frequently above the long-term mean and included some of the highest values observed during the study period.

In contrast, other parts of the series presented lower concentration levels and reduced variability, suggesting that SO_2 dynamics were not uniform over time. The observed temporal behavior indicates that surface-level SO_2 variability was structured by a combination of

seasonal conditions, short-term concentration episodes, and atmospheric dispersion processes.

These patterns support the inclusion of both autoregressive predictors and meteorological covariates in the forecasting framework. Autoregressive features capture temporal persistence and short-term dependence in the SO_2 series, while meteorological variables provide information on atmospheric conditions that may influence pollutant accumulation, dispersion, and removal.

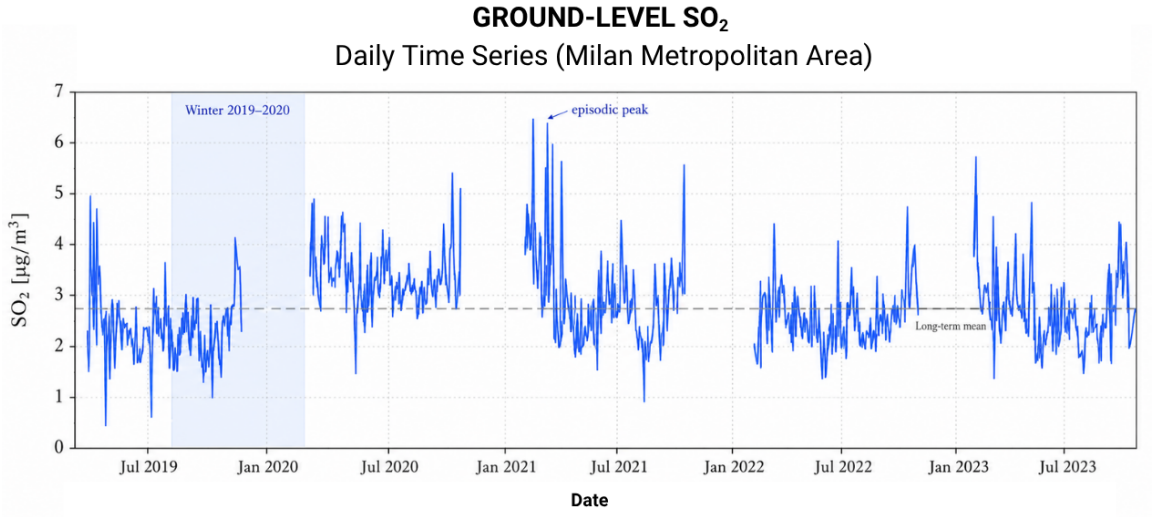

Figure 4.1: Daily temporal evolution of ground-level SO_2 concentrations in the Milan Metropolitan Area. The blue line represents the daily SO_2 time series, the horizontal dashed line indicates the long-term mean concentration, and the shaded interval highlights the winter 2019–2020 period, during which several elevated SO_2 values were observed. The annotated peak illustrates the episodic nature of high-concentration events within the study period.

4.1.3. Target Distribution

The original SO_2 target distribution exhibited positive skewness, with a right tail extending toward concentrations above 4 μg m^{-3}, as shown in Figure 4.2. This distributional pattern indicates the presence of episodic high-concentration events, which may occur under unfavorable atmospheric dispersion conditions or during periods influenced by localized or regional emission episodes.

To reduce the influence of extreme values during model optimization, a logarithmic transformation was applied to the SO_2 target variable in the enhanced modeling configurations. Specifically, the transformation was implemented using the $\log(1 + x)$ function, which is

suitable for non-negative concentration values because it compresses large observations while preserving zero values and maintaining the relative ordering of the data. As shown in Figure 4.2, the transformed distribution became more symmetric, reducing the dominance of episodic high-concentration values during training.

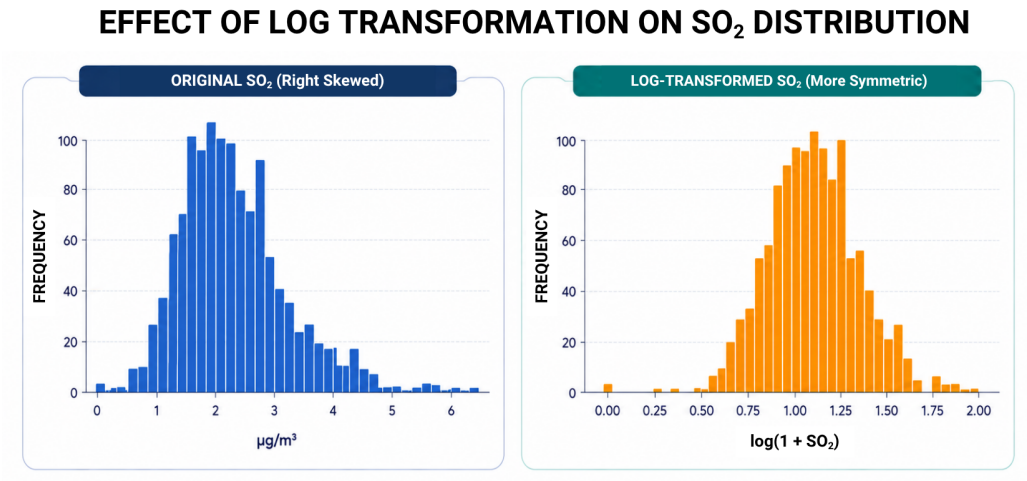

Figure 4.2: Comparison between the original SO_2 target distribution and the logarithmically transformed distribution obtained using $\log(1 + x)$.

This transformation was particularly relevant for models optimized using error-based loss functions or evaluation metrics, such as RMSE, because large residuals associated with high-concentration episodes can disproportionately influence the optimization process. By compressing the upper tail of the SO_2 distribution, the logarithmic transformation improved numerical stability and allowed the models to learn both background concentration variability and episodic pollution events more consistently.

4.1.4. Missing Data Assessment

The station-aggregated dataset contained only limited missing observations. For classical machine learning experiments, missing station values primarily reduced the number of monitoring stations contributing to the daily metropolitan average, rather than producing large temporal gaps in the final target series.

However, AutoML and TSFM frameworks required continuous temporal sequences. Missing dates were therefore reconstructed by creating a complete daily temporal index and applying time-based interpolation after reindexing.

The main preprocessing operations used to ensure temporal continuity were:

- creation of a complete daily index using `pandas.date_range()`;
- temporal reindexing using `reindex()`;
- time-based interpolation using `interpolate(method="time")`.

Lag-feature generation introduced additional missing values at the beginning of the time series due to shifting operations. These rows were removed before model training. After preprocessing and lag construction, the effective sample sizes became:

- Version 2: 985 observations;
- Version 3: 978 observations;
- Version 6: approximately 970 valid observations.

4.2. Feature Selection and Importance Analysis

The initial feature-selection strategy used absolute Pearson correlation coefficients with respect to the SO_2 target variable. This approach was implemented as a simple baseline screening method before model training, with the objective of identifying predictors showing a first-order linear association with the target.

When applying the stronger correlation threshold of $|r| > 0.30$, the only predictor variable that exceeded the threshold was `prev_boundary_layer_height`, with a correlation coefficient of:

$$r = -0.346.$$

The target variable itself, `ground_so2`, also presented a correlation of 1.000, as expected, but it was not considered a valid predictor because it corresponds to the target series. Therefore, only `prev_boundary_layer_height` was identified as a strongly correlated explanatory variable under this threshold.

The negative correlation indicates that lower planetary boundary layer heights were associated with higher surface SO_2 concentrations. This result is physically consistent because shallow boundary layers reduce the vertical volume available for pollutant dispersion and can therefore favor pollutant accumulation near the surface.

After the preliminary filtering stage, fifteen candidate variables were retained for the baseline modeling configuration, while three variables were removed. The retained variables were:

- `ground_so2`;

- satellite_so2;
- cams_so2_prev;
- cams_so2_curr;
- curr_temp_celsius;
- curr_surface_net_solar_radiation;
- curr_surface_pressure;
- curr_wind_speed;
- curr_boundary_layer_height;
- prev_temp_celsius;
- prev_surface_net_solar_radiation;
- prev_surface_pressure;
- prev_total_precipitation;
- prev_wind_speed;
- prev_boundary_layer_height.

Although Pearson screening provided an interpretable first-order assessment of linear relationships, its results also showed that most atmospheric predictors did not exhibit strong individual linear correlations with the SO_2 target. This limitation is important because pollutant variability is influenced by non-linear interactions among meteorology, atmospheric transport, temporal persistence, and regional background conditions. For this reason, the correlation-based baseline was later complemented with model-based feature selection using Random Forest feature importance.

4.2.1. Random Forest Importance: Version 2

Version 2 replaced Pearson correlation screening with Random Forest feature importance computed exclusively on the training subset. This avoided information leakage from the test period and enabled the identification of non-linear predictor contributions.

Figure 4.3 presents the Random Forest feature-importance ranking obtained in Version 2. The plot shows that autoregressive SO_2 variables dominated the predictive structure, with recent lag and rolling-window features occupying the highest positions. This confirms that

short-term concentration persistence was one of the main sources of information for the classical machine learning models.

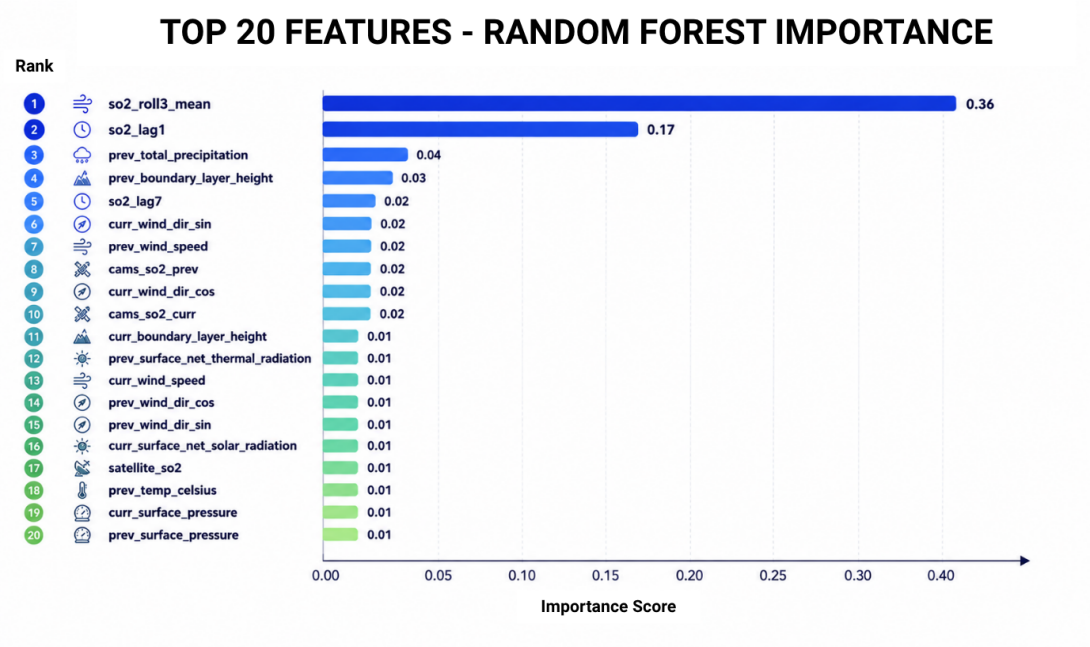

Figure 4.3: Random Forest feature-importance ranking obtained in Version 2.

Together, the dominant autoregressive predictors accounted for more than 50% of the total feature-importance contribution. This result demonstrated that short-term persistence and recent concentration history were central components of daily SO_2 forecasting.

Among the environmental covariates, the most relevant predictors included previous-day precipitation, planetary boundary layer height, wind speed, and wind-direction components. CAMS SO_2 reanalysis variables also contributed positively, indicating that large-scale atmospheric background information added predictive value beyond local monitoring-station history.

4.2.2. Enhanced Feature Space Importance: Version 3

Version 3 expanded the predictor space through the introduction of additional lag variables, rolling-window statistics, and cyclical temporal encodings. The objective was to capture atmospheric persistence across multiple temporal scales while representing seasonal variability more efficiently than in the previous feature-engineering configurations.

Figure 4.4 presents the Random Forest feature-importance ranking obtained after implementing the enhanced feature space. The results show a clear dominance of SO_2

rolling statistics and autoregressive descriptors, indicating that recent pollutant behavior remained the most informative source of predictive information. In particular, rolling-window features consistently occupied the highest-ranking positions, demonstrating that pollutant dynamics were better represented through short-, weekly-, and biweekly-scale concentration patterns rather than through isolated lag observations alone.

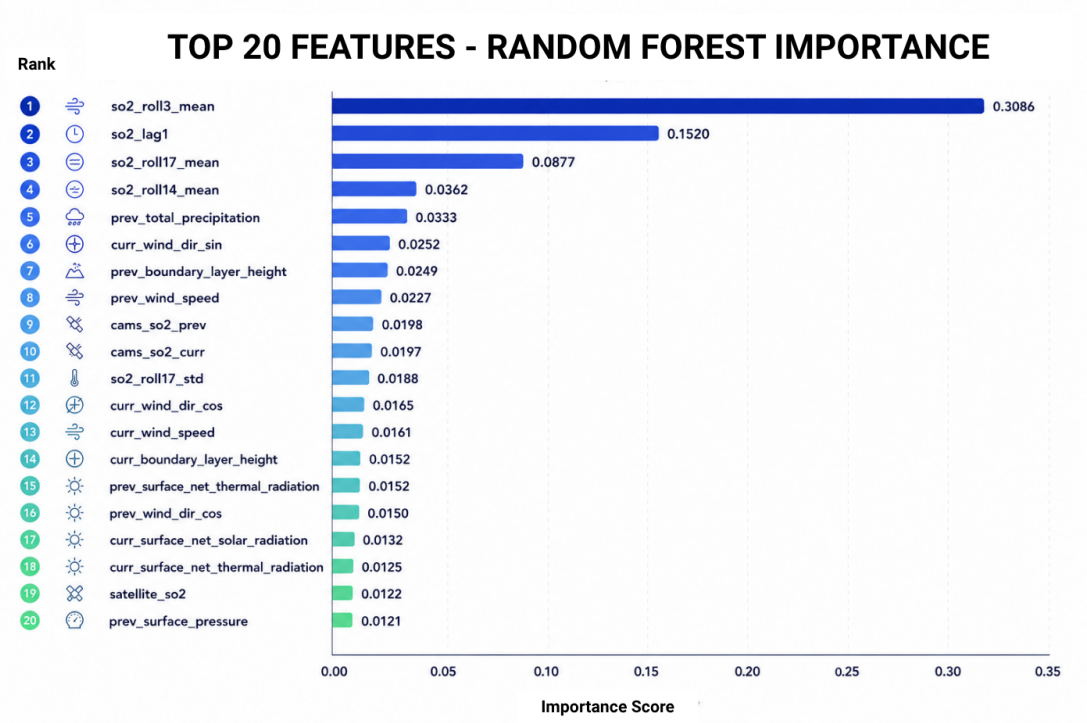

Figure 4.4: Random Forest feature-importance ranking obtained in Version 3 after the introduction of additional lag variables, rolling statistics, and cyclical temporal encodings.

The prominence of the rolling-window variables suggests that SO_2 concentrations were influenced by atmospheric conditions operating across multiple temporal horizons. The 3-day rolling mean primarily captured short-term persistence, whereas the 7-day and 14-day rolling statistics represented broader weekly and biweekly concentration tendencies. Together, these features provided a more stable and informative description of pollutant evolution than individual lag variables alone.

An additional improvement introduced in Version 3 was the replacement of one-hot month encodings with cyclical temporal representations. Previous iterations allocated a substantial portion of the available TOP_K feature-selection budget to month dummy variables, limiting the inclusion of potentially informative atmospheric predictors. The revised encoding strategy reduced feature-space dimensionality while preserving the cyclical nature of seasonal atmospheric processes.

The transition from one-hot to cyclical temporal encoding is illustrated in Figure 3.6. By representing temporal variables through sine and cosine transformations, the framework preserved seasonal continuity while reducing redundancy in the predictor space. This modification improved feature-selection efficiency and allowed a greater number of physically meaningful meteorological and autoregressive variables to be retained during model training.

The resulting feature ranking confirms that Version 3 achieved a more balanced predictor space, combining temporal persistence descriptors, seasonal representations, and environmental variables within the feature-selection framework. This enhancement established the foundation for the improved machine learning performance observed in the subsequent experiments.

4.2.3. AutoGluon Permutation Importance: Version 6

The final AutoGluon framework used permutation importance computed over five validation permutations. This analysis evaluated the effect of randomly perturbing each predictor on validation performance.

The variable `so2_lag1` emerged as the most informative predictor within the complete 36-feature space. This confirmed the importance of immediate temporal persistence in daily SO_2 forecasting.

Environmental variables also exhibited consistent predictive contribution. These included precipitation, wind direction, surface pressure, temperature, and Sentinel-5P SO_2 atmospheric-column information.

The combined feature-importance results across Versions 2, 3, and 6 showed that accurate SO_2 prediction depended on the simultaneous integration of:

- temporal autocorrelation;
- meteorological forcing;
- atmospheric transport variables;
- satellite-derived atmospheric information.

4.3. Classical Machine Learning Results

4.3.1. Version 1 - Baseline

Version 1 established the initial machine learning benchmark using random train-test splitting, Pearson-selected features, no lag engineering, and no target transformation. This configuration was used as a first reference point to evaluate the effect of the methodological improvements introduced in the subsequent versions.

Table 4.1 summarizes the main Version 1 results.

Performance of the Main Version 1 Baseline Models

Model	NRMSE(std)
Random Forest	0.944
RF + SVR Ensemble	**0.924**

Table 4.1: Performance of the principal Version 1 baseline machine learning models evaluated using the NRMSE(std) metric. Lower values indicate better predictive performance.

The best-performing individual model was Random Forest, with an NRMSE(std) of 0.944. The strongest ensemble configuration was the RF + SVR ensemble, which achieved an NRMSE(std) of 0.924. It is important to note that R^2 and RMSE are not reported for Version 1 because random train–test splitting violates the temporal structure of the dataset and may lead to overly optimistic performance estimates. Consequently, these metrics are not directly comparable with the chronological validation results reported for Versions 2–6.

Although the baseline models captured broad seasonal behavior, they systematically smoothed high-frequency concentration peaks. Therefore, Version 1 results provided a useful initial benchmark, but they also revealed several methodological weaknesses. Random splitting did not respect the chronological structure of the forecasting task, Pearson screening was limited to linear relationships, and the absence of lag features prevented the models from explicitly using recent SO_2 history.

4.3.2. Version 2 - Improved Pipeline

Version 2 corrected the principal methodological limitations identified in Version 1. The updated pipeline introduced chronological train-test splitting, lag-feature engineering, $\log_{1+p}$ target transformation, IQR-normalized evaluation metrics, and expanded ensemble strategies.

Table 4.2 summarizes the performance of the strongest Version 2 configurations.

Performance of the Main Version 2 Configurations

Model	RMSE	R^2	NRMSE(IQR)	NRMSE(std)
Voting Ensemble (GTB + MLPR)	0.575	0.434	0.616	0.752
Stacking Ensemble (GTB + RF + CAT)	0.577	0.430	0.618	0.755
XGBoost	0.581	0.421	0.623	0.761

Table 4.2: Performance comparison of the main Version 2 machine learning configurations evaluated on the testing subset. RMSE is expressed in $\mu g/m^3$.

The best-performing configuration was the Voting Ensemble combining Gradient Tree Boosting and Multi-Layer Perceptron Regressor, which achieved an RMSE of 0.575 $\mu g/m^3$, an R^2 of 0.434, and an NRMSE(IQR) of 0.616. The Stacking Ensemble ranked second, followed by XGBoost as the strongest standalone model.

Figure 4.5 presents the observed and predicted SO_2 concentrations for the best Version 2 configuration.

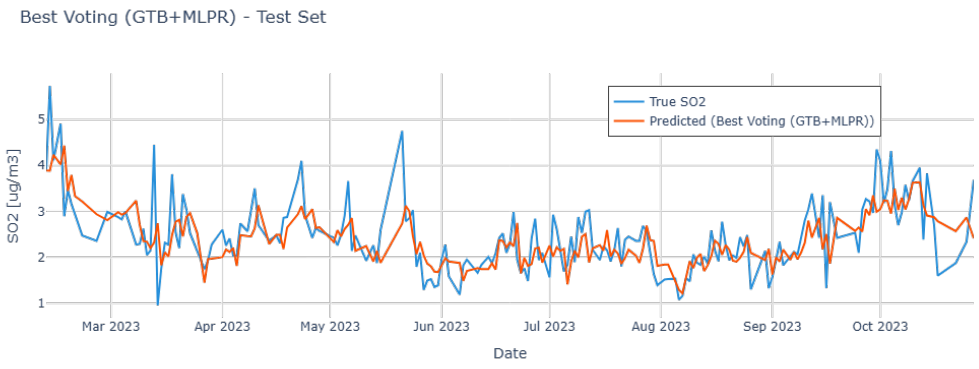

Figure 4.5: Observed and predicted SO_2 concentrations for the best Version 2 configuration.

The inclusion of lag features improved the temporal tracking capability of the models relative to Version 1. The predictions followed the general temporal evolution of the observed SO_2 series more closely, although short-lived concentration peaks remained difficult to reproduce accurately.

These results show that the methodological corrections introduced in Version 2 improved model reliability. Chronological splitting provided a more realistic forecasting evaluation, while lag features allowed the models to incorporate short-term pollutant persistence.

4.3.3. Version 3 - Enhanced Feature Engineering

Version 3 retained the same general model families used in Version 2, but extended the feature-engineering framework. The revised feature set included additional lag variables, rolling-window means, rolling-window standard deviations, and cyclical calendar features.

Table 4.3 summarizes the performance of the strongest Version 3 configurations.

Performance of the Main Version 3 Configurations

Model	RMSE	R^2	NRMSE(IQR)	NRMSE(std)
Voting Ensemble (GTB + SVR)	0.561	**0.455**	**0.600**	**0.738**
Stacking Ensemble (GTB + RF + CAT)	0.569	0.438	0.609	0.750
XGBoost	0.572	0.433	0.612	0.753

Table 4.3: Performance comparison of the principal Version 3 machine learning configurations evaluated on the testing subset. RMSE is expressed in $\mu g/m^3$.

The strongest model was the Voting Ensemble combining Gradient Tree Boosting and Support Vector Regression. This configuration achieved an RMSE of 0.561 $\mu g/m^3$, an R^2 of 0.455, and an NRMSE(IQR) of 0.600. This represented the best performance among all classical machine learning configurations evaluated in this thesis.

Figure 4.6 presents the observed and predicted SO_2 concentrations for the best Version 3 configuration.

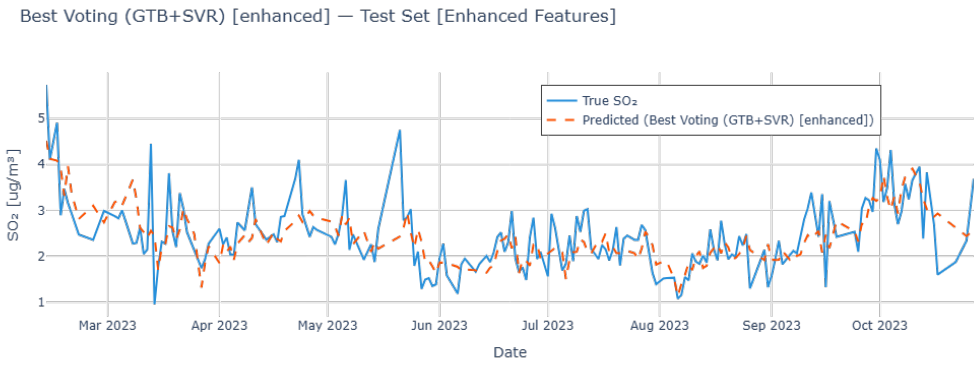

Figure 4.6: Observed and predicted SO_2 concentrations obtained using the best-performing Version 3 model, namely the Voting Ensemble (GTB + SVR). The figure illustrates the model's ability to reproduce the temporal variability of the measured pollutant concentrations across the testing period.

The predictions followed the general temporal behavior of the observed SO_2 series more closely than in Version 2. Although short-lived concentration peaks were still partially smoothed, the enhanced temporal feature set improved the representation of recent concentration dynamics and seasonal variability.

The improvement from Version 2 to Version 3 was moderate but consistent. These results demonstrate that better feature representation improved forecasting performance even without changing the underlying model families. In particular, the extended lag hierarchy, rolling-window statistics, and cyclical temporal encoding allowed the models to represent recent temporal behavior and seasonal structure more effectively.

4.3.4. Comparative Performance of Classical Models

The classical machine learning results showed a progressive improvement from Version 1 to Version 3. The transition from the initial baseline to the improved pipeline corrected key methodological weaknesses, while the enhanced feature-engineering version produced the strongest classical machine learning result.

Table 4.4 summarizes the performance progression across the three classical machine learning versions.

Performance Progression Across Classical Machine Learning Versions

Version	Best Configuration	R^2	NRMSE(std)
Version 1	RF + SVR Ensemble	—	0.924
Version 2	Voting Ensemble (GTB + MLPR)	0.434	0.752
Version 3	Voting Ensemble (GTB + SVR)	**0.455**	**0.738**

Table 4.4: Performance evolution of the best-performing classical machine learning configurations across Versions 1–3. The results demonstrate the progressive improvement obtained through data harmonization, enhanced feature engineering, and optimized ensemble design.

The progression from Version 1 to Version 3 shows that the methodological corrections introduced in the pipeline improved the reliability and predictive skill of the classical machine learning framework.

Overall, the results confirmed that classical machine learning models could capture part of the SO_2 variability when supported by temporal and environmental predictors. However, their performance remained lower than the later AutoML and TSFM configurations.

4.4. Foundation Model and AutoML Results

4.4.1. Version 4 – SO_2–Only Forecasting Baseline

Version 4 evaluated whether daily SO_2 concentrations could be forecast using only the historical target series, without additional meteorological, satellite, or atmospheric reanalysis variables. The objective of this stage was to establish a foundation-model and AutoML baseline under a purely autoregressive forecasting setting.

Table 4.5 summarizes the performance of the SO_2-only baseline forecasting configurations.

Performance of SO_2-Only Foundation Model Baselines

Model	Type	RMSE	R^2	NRMSE(IQR)	NRMSE(std)
Naive Seasonal t-7	Baseline	0.938	-0.491	0.996	1.221
TimesFM-1.0-200m	Zero-shot	0.868	-0.276	0.921	1.130
AutoGluon medium [SO_2 only]	AutoML	0.840	-0.185	0.890	1.088
Chronos-Bolt-Small	Zero-shot	**0.788**	**-0.053**	**0.837**	**1.026**

Table 4.5: Performance comparison of SO_2-only baseline forecasting configurations. RMSE is expressed in $\mu g/m^3$.

All SO_2-only configurations produced negative R^2 values, showing that historical SO_2 concentrations alone were insufficient for skillful forecasting. Among the tested models, Chronos-Bolt-Small produced the strongest result, with an RMSE of 0.788 $\mu g/m^3$, an R^2 of -0.053, and an NRMSE(IQR) of 0.837. Although this was the best performance within the SO_2-only group, it still remained below the threshold of positive predictive skill.

Figure 4.7 presents the observed and predicted SO_2 concentrations obtained with Chronos-Bolt-Small.

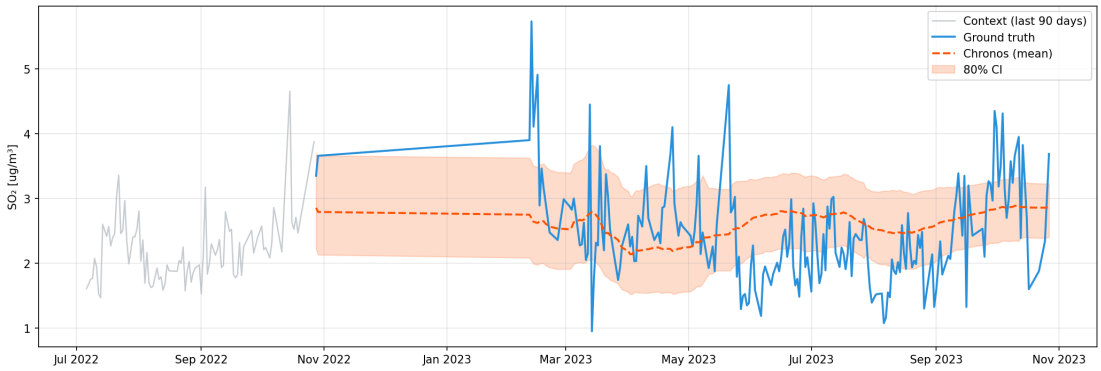

Figure 4.7: Observed and predicted SO_2 concentrations obtained with Chronos-Bolt-Small in the SO_2-only forecasting configuration.

These results indicate that SO_2 forecasting could not be treated as a purely autoregressive problem. The weak predictive performance observed in Version 4 motivated the introduction of external environmental covariates in the following stage.

4.4.2. Version 5 - First Covariate-Aware TSFM and AutoML Experiment

Version 5 introduced meteorological and environmental covariates into the TSFM and AutoML forecasting framework. The objective of this stage was to determine whether adding external atmospheric information could improve performance relative to the SO_2-only baseline.

Table 4.6 summarizes the results obtained in this first covariate-aware experiment.

Performance of Covariate-Aware Forecasting Configurations

Model	Type	RMSE	R^2	NRMSE(IQR)	NRMSE(std)
AutoGluon `medium_quality`	AutoML	0.816	-0.128	0.866	1.062
Chronos-2 zero-shot	Zero-shot	0.820	-0.141	0.871	1.068
Chronos-2 LoRA fine-tuned	Fine-tuned	0.825	-0.154	0.876	1.074

Table 4.6: Performance comparison of the first covariate-aware forecasting configurations. RMSE is expressed in $\mu g/m^3$.

The introduction of environmental covariates produced only a limited improvement relative to the SO_2-only baseline configurations. AutoGluon improved from an R^2 of -0.185 in the SO_2-only setting to -0.128 when covariates were added, and its NRMSE(IQR) decreased from 0.890 to 0.866. This indicates that the additional atmospheric variables provided some useful predictive information.

However, all covariate-aware configurations in this stage still produced negative R^2 values. Therefore, the introduction of covariates alone was not sufficient to achieve skillful forecasting. In particular, both Chronos-2 zero-shot and Chronos-2 LoRA fine-tuned configurations remained below zero in terms of explained variance.

Figure 4.8 presents the observed and predicted SO_2 concentrations obtained with the best Version 5 configuration.

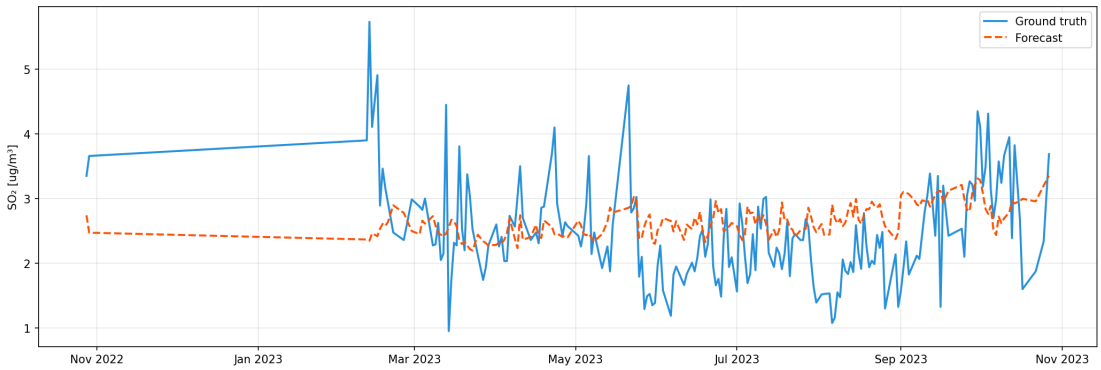

Figure 4.8: Observed and predicted SO_2 concentrations obtained with the best Version 5 configuration, namely AutoGluon `medium_quality` with covariates.

These results show that the initial covariate-injection strategy implemented in Version 5 was not yet sufficient to solve the forecasting problem. Although meteorological and environmental covariates were introduced, the experiment still relied on an early covariate-aware configuration, AutoGluon `medium_quality`, and limited foundation-model adaptation. As a result, all Version 5 configurations remained below zero in terms of explained variance, indicating that the inclusion of external variables alone was not enough to obtain skillful SO_2 forecasting.

Version 6 was therefore designed as a final optimization stage rather than a simple repetition of Version 5. The main changes were: (i) the use of the complete environmental feature suite, including the full set of engineered temporal, meteorological, satellite-derived, and atmospheric covariates; (ii) the evaluation of stronger AutoML configurations, including `best_quality`; (iii) the explicit comparison between Chronos-2 with covariates but without fine-tuning and Chronos-2 with parameter-efficient LoRA fine-tuning; and (iv) the use of domain-specific adaptation to improve the ability of the pretrained model to represent local SO_2 dynamics. In the LoRA configuration, the adaptation was performed using the selected low-rank update strategy defined in the methodology, with rank $r = 8$ and scaling factor $\alpha = 16$. This progression allowed Version 6 to test whether the performance limitation observed in Version 5 was caused by the covariates themselves, by the model configuration, or by the absence of sufficient task-specific adaptation.

4.4.3. Version 6 - Covariate-Aware Fine-Tuning Strategy

Version 6 implemented the final maximum-R^2 optimization strategy. This stage combined the complete environmental feature set with advanced AutoML configurations and parameter-efficient adaptation of Chronos-2 through Low-Rank Adaptation (LoRA) fine-

tuning. The objective was to evaluate the maximum predictive performance attainable when temporal foundation models were provided with comprehensive atmospheric context and domain-specific adaptation.

Table 4.7 summarizes the best-performing configurations obtained during the final experimental stage.

Performance of All-Features and Maximum-R^2 Configurations

Model	R^2	RMSE	NRMSE(IQR)	NRMSE(std)
AutoGluon `medium_quality`	0.816	0.372	0.226	0.489
Chronos-2 (covariates, no fine-tuning)	0.920	0.246	0.149	0.323
Chronos-2 LoRA fine-tuned	**0.976**	**0.134**	**0.081**	**0.176**
AutoGluon `best_quality + LoRA`	0.967	0.158	0.096	0.208

Table 4.7: Performance comparison of the final all-features and maximum-R^2 configurations. RMSE is expressed in $\mu g\, m^{-3}$.

The final optimization stage produced the strongest forecasting performance of the entire study. The best overall configuration was Chronos-2 with environmental covariates and LoRA fine-tuning, achieving an R^2 of 0.976, an RMSE of 0.134 $\mu g\, m^{-3}$, an NRMSE(IQR) of 0.081, and an NRMSE(std) of 0.176. AutoGluon `best_quality` combined with LoRA adaptation ranked second, while Chronos-2 using the same environmental covariates but without fine-tuning achieved the third-best performance.

Figure 4.9 presents the observed and predicted SO_2 concentrations obtained with the best-performing Version 6 configuration.

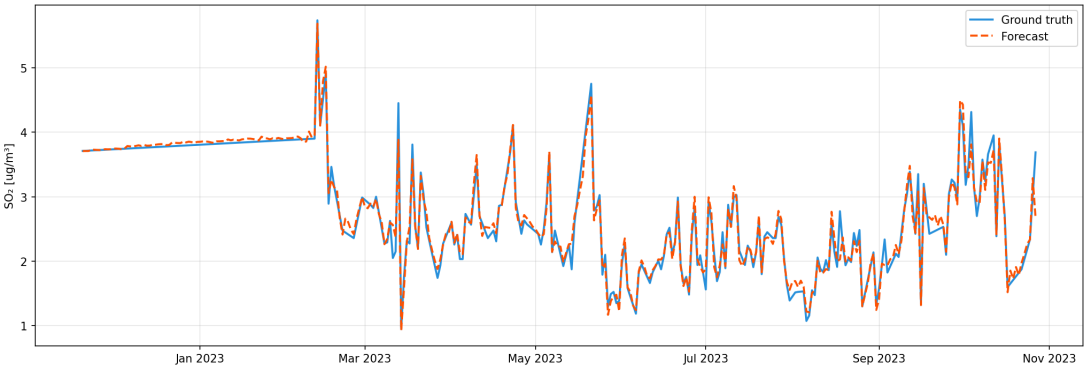

Figure 4.9: Comparison between observed and predicted SO_2 concentrations during the testing period using Chronos-2 with environmental covariates and LoRA fine-tuning.

The prediction series closely reproduces both the long-term temporal evolution and the short-term fluctuations of the observed SO_2 concentrations. Compared with previous experimental stages, the results indicate that forecasting performance improved substantially when pretrained temporal representations were complemented by physically meaningful atmospheric covariates and subsequently adapted to the local domain through parameter-efficient fine-tuning.

These findings directly support the central hypothesis of this research: temporal foundation models alone are insufficient to fully represent atmospheric pollutant dynamics, whereas the combination of pretrained temporal knowledge, environmental context, and domain adaptation yields substantially superior forecasting performance. The progressive improvement observed from zero-shot forecasting to covariate-aware modeling and finally to LoRA adaptation demonstrates the complementary contribution of each component within the proposed framework.

Because the final configuration used observed meteorological and atmospheric variables during the evaluation period, the reported results should be interpreted as a retrospective upper-bound estimate of predictive capability rather than a fully operational forecasting scenario. In a real-world deployment, these covariates would need to be replaced by forecasted meteorological and atmospheric composition variables generated by numerical weather prediction and atmospheric forecasting systems.

4.4.4. Feature Importance in the Final Configuration

To better understand the drivers of the strongest forecasting configuration, feature importance was analyzed for the final all-features experiment.

Figure 4.10 presents the 20 most important predictors for SO_2 forecasting.

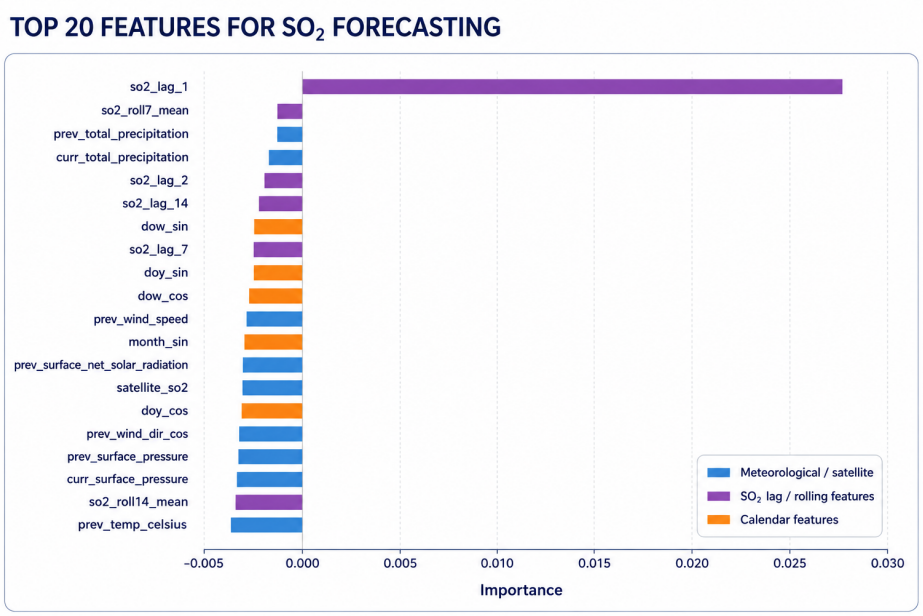

Figure 4.10: Top 20 most important predictors in the final SO_2 forecasting configuration.

The importance ranking confirms that lagged SO_2 variables remained dominant predictors, particularly `so2_lag_1`, `so2_roll7_mean`, and other lagged or rolling-window features. At the same time, meteorological variables such as precipitation, wind speed, pressure, temperature, and satellite-derived SO_2 also contributed meaningfully. Calendar variables, including cyclical day-of-week and month encodings, further supported the representation of temporal periodicity.

Overall, the feature-importance results demonstrate that the strongest forecasting performance emerged from the joint contribution of temporal persistence, meteorological forcing, atmospheric conditions, and seasonal structure.

4.5. Model Comparison

4.5.1. Overall Performance Ranking

Table 4.8 summarizes the overall ranking of the principal configurations evaluated across the six experimental versions.

Grand Leaderboard: Foundation Models vs Classical Machine Learning

Model	R^2	NRMSE(IQR)
Chronos-2 LoRA fine-tuned [all features]	**0.9762**	**0.08126**
AutoGluon `best_quality` [all features + LoRA]	0.9667	0.09601
Chronos-2 zero-shot [all features]	0.9197	0.14914
AutoGluon `medium_quality` [all features]	0.8159	0.22577
Best Voting GTB + MLPR [Classical ML]	0.4342	0.61600
XGBoost [Classical ML]	0.4214	0.62300
Random Forest [Classical ML]	0.4081	0.63000
SVR [Classical ML]	0.3851	0.64200
Chronos-Bolt-Small zero-shot	-0.0533	0.83706
AutoGluon medium [SO_2 only]	-0.1950	0.89157
TimesFM-1.0 zero-shot	-0.2762	0.92137
Naive Seasonal t-7 [no covariates]	-0.4913	0.99600

Table 4.8: Overall ranking of the principal forecasting configurations evaluated throughout the six experimental versions.

Figure 4.11 provides a graphical comparison of the same models in terms of R^2 and NRMSE(IQR).

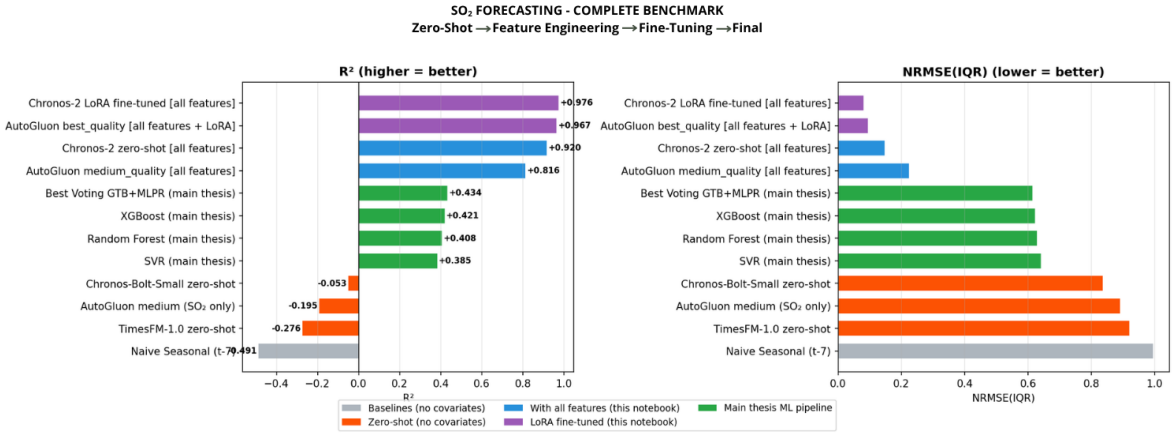

Figure 4.11: Global performance comparison across the main forecasting configurations, using R^2 and NRMSE(IQR).

The final comparison highlights a clear progression from weak autoregressive baselines to highly accurate covariate-aware and fine-tuned foundation-model configurations. The

strongest results were achieved only in Version 6, where the complete environmental feature suite and Chronos-2 LoRA fine-tuning produced near-exact agreement with the observed series. Classical machine learning models performed substantially better than the SO_2-only foundation baselines, but they remained below the final all-features configurations.

4.6. Performance Evolution Across Experimental Versions

To facilitate the interpretation of the complete experimental workflow, Figure 4.12 summarizes the progression of forecasting performance across the six experimental versions evaluated in this thesis.

The results reveal a clear methodological evolution. The first three versions focused on classical machine learning and progressively improved predictive performance through enhanced feature engineering, chronological validation, ensemble learning, and more informative temporal representations. These improvements increased the best classical machine learning performance from the initial baseline to an R^2 of 0.455.

The introduction of Time Series Foundation Models in Versions 4 and 5 demonstrated that pretrained temporal architectures alone were insufficient for skillful SO_2 forecasting. Although foundation models have shown strong performance in many forecasting domains, SO_2-only and early covariate-aware configurations produced limited predictive skill, highlighting the importance of environmental forcing variables.

The strongest performance was achieved in Version 6, where the complete environmental feature set was combined with Chronos-2 and parameter-efficient LoRA fine-tuning. This configuration produced the highest forecasting accuracy of the entire study, demonstrating that optimal performance emerged from the combination of temporal representations, environmental covariates, and domain adaptation rather than from model complexity alone.

Figure 4.12 illustrates this progression from baseline machine learning approaches to fully adapted foundation-model architectures.

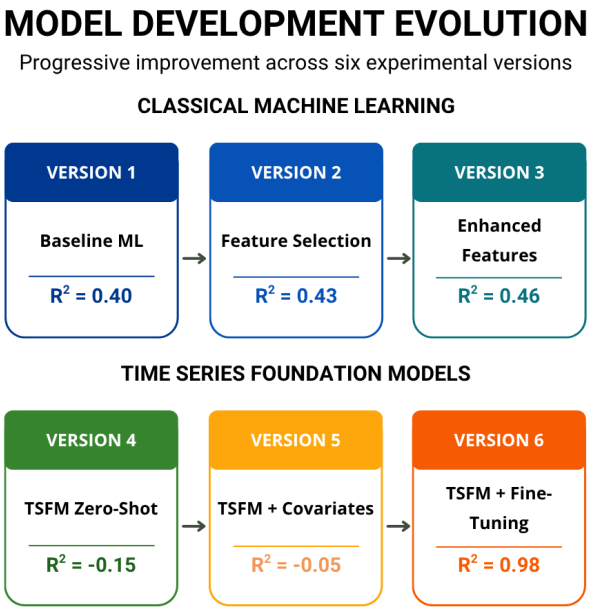

Figure 4.12: Evolution of forecasting performance across the six experimental versions evaluated in this thesis. The figure summarizes the transition from baseline classical machine learning models to covariate-aware Time Series Foundation Models with LoRA fine-tuning.

The performance trajectory demonstrates that improvements were not driven by increasingly complex model architectures alone. Instead, the largest gains were obtained when physically meaningful environmental information was integrated into the forecasting process. This finding reinforces one of the central conclusions of the thesis: accurate atmospheric pollutant forecasting requires the joint representation of temporal persistence and environmental forcing mechanisms.

4.7. Computational Challenges

Several non-trivial computational and methodological obstacles emerged throughout the implementation process. The most significant issues and their corresponding solutions are summarized below.

4.7.1. XGBoost and scikit-learn Compatibility

A compatibility break between XGBoost 3.x and scikit-learn ensemble wrappers prevented `VotingRegressor` and `StackingRegressor` from recognizing `XGBRegressor` as a valid regressor object.

The issue was resolved through a lightweight compatibility wrapper:

```
class XGBRegressorFixed(XGBRegressor, RegressorMixin)
```

which explicitly restored the `_estimator_type = 'regressor'` attribute required by scikit-learn.

4.7.2. TensorFlow and PyTorch DLL Conflicts

Simultaneous TensorFlow and PyTorch installation inside a single Conda environment generated DLL runtime conflicts under Windows systems during AutoGluon initialization.

The solution consisted of fully separating:

- TensorFlow workflows;
- PyTorch workflows

into independent virtual environments.

4.7.3. Foundation Model CPU Latency

Chronos-Bolt-Small and TimesFM exhibited substantial CPU inference latency because transformer attention operations were executed without GPU acceleration.

Chronos inference required approximately 2–5 minutes per forecast run, while TimesFM occasionally exceeded 15 minutes per inference cycle. Similarly, LoRA fine-tuning required 1–3 hours depending on the selected update budget.

These constraints limited extensive hyperparameter optimization and motivated the adoption of parameter-efficient adaptation strategies.

4.7.4. AutoGluon Temporal Gap Constraints

AutoGluon-TimeSeries requires perfectly regular temporal indices. However, the raw SO_2 series contained occasional missing dates due to:

- sensor downtime;
- transmission failures;
- incomplete observations.

The issue was resolved by reconstructing a complete daily temporal index using `pandas.reindex()` followed by `interpolate(method="time")` to fill missing timestamps prior to forecasting.

5 | Discussion

This chapter interprets the principal findings of the research within the broader context of environmental monitoring, atmospheric data fusion, geoinformatics, and Time Series Foundation Models (TSFMs). While Chapter 4 presented the quantitative results obtained across the six experimental versions, this chapter focuses on the scientific significance of those results, their implications for atmospheric forecasting, the limitations of the proposed framework, and potential directions for future research.

5.1. Scientific Interpretation

One of the most important findings of this research is that surface-level SO_2 forecasting cannot be treated as a purely autoregressive problem. Historical pollutant concentrations alone do not contain sufficient information to characterize future atmospheric conditions. This result is physically consistent with the complex processes governing atmospheric pollutant variability.

Surface-level SO_2 concentrations emerge from the interaction of multiple environmental mechanisms, including emission variability, atmospheric transport, boundary-layer dynamics, precipitation scavenging, vertical mixing, and regional background concentrations. Consequently, the observed SO_2 time series represents the outcome of a coupled atmospheric system rather than an isolated temporal sequence.

The poor performance obtained by the SO_2-only forecasting configurations supports this interpretation. Naive Seasonal forecasting, TimesFM, AutoGluon using only historical SO_2, and Chronos-Bolt-Small all produced negative R^2 values. These results indicate that temporal persistence alone was insufficient for skillful forecasting and that additional environmental information is required to represent the physical processes controlling pollutant evolution.

The progression across the experimental versions further reinforces this conclusion. Improvements in predictive performance were consistently associated with richer environmental representation rather than simply increasing model complexity. The introduction

of lag variables, rolling-window statistics, and cyclical temporal encoding improved forecasting performance by capturing short-term atmospheric persistence at daily, weekly, and biweekly timescales. However, temporal persistence alone remained insufficient because it describes how pollutant concentrations evolve but not the environmental processes responsible for those changes.

Meteorological variables, CAMS atmospheric composition products, and Sentinel-5P observations provided complementary information describing atmospheric forcing mechanisms. Variables related to precipitation, wind conditions, atmospheric pressure, temperature, and atmospheric composition supplied predictive information that could not be inferred directly from historical SO_2 measurements. The inability of the SO_2-only foundation-model experiments to achieve meaningful predictive skill demonstrates that temporal modeling capacity alone cannot compensate for the absence of environmental context.

The strongest performance emerged when temporal memory and environmental forcing were integrated within a unified forecasting framework. The final Chronos-2 configuration combined pretrained temporal representations, engineered temporal features, environmental covariates, and parameter-efficient adaptation through LoRA fine-tuning. The resulting performance improvement suggests that the combination of pretrained temporal representations, environmental covariates, and domain adaptation provides a substantially richer forecasting framework than classical machine learning approaches operating on engineered tabular features alone.

From a physical perspective, this result is particularly relevant for SO_2 forecasting in an urban-industrial environment such as Milan, where atmospheric transport, boundary-layer evolution, and meteorological variability strongly influence ground-level concentrations. The results therefore indicate that accurate forecasting requires the joint representation of temporal persistence and environmental forcing rather than reliance on either component independently.

An additional observation emerges from the performance evolution across the six experimental versions. Forecasting skill improved progressively as the framework incorporated increasingly informative environmental representations and more advanced modeling strategies. Version 1 provided an initial benchmark based on a simplified feature space and random train–test partitioning, while Versions 2 and 3 introduced chronological validation, expanded lag structures, rolling-window statistics, and improved feature selection. Versions 4 through 6 extended the framework toward Time Series Foundation Models, environmental covariate integration, AutoML forecasting systems, and parameter-efficient

fine-tuning. The resulting progression demonstrates that performance improvements were not driven by a single methodological change but rather by the cumulative contribution of improved validation design, feature engineering, environmental information, and model adaptation. This evolutionary pattern reinforces the central conclusion of the thesis that accurate atmospheric forecasting requires the joint integration of physically meaningful environmental data and advanced temporal learning architectures.

5.2. Comparison with Previous Studies

The findings obtained in this thesis are broadly consistent with previous environmental machine learning studies emphasizing the importance of meteorological forcing and atmospheric transport variables for air-quality estimation [11, 14, 32]. Similar to these studies, the present research found that precipitation, wind-related variables, atmospheric composition indicators, and temporal persistence metrics contributed substantially to forecasting performance.

The importance of lag variables and rolling-window statistics is also consistent with previous air-quality forecasting literature. Pollutant concentrations often exhibit short-term persistence because meteorological conditions, atmospheric circulation patterns, and emission sources tend to remain partially stable over consecutive days. The repeated selection of lag and rolling-window features among the most important predictors confirms the relevance of temporal memory for atmospheric forecasting.

However, the present study extends previous findings by demonstrating that temporal persistence alone is insufficient for accurate SO_2 forecasting. Although lag variables consistently improved predictive performance, they could not compensate for the absence of meteorological, atmospheric composition, and satellite-derived covariates. This distinction is important because it suggests that increasingly sophisticated forecasting architectures cannot overcome limitations associated with inadequate environmental representation.

The systematic comparison between classical machine learning approaches and Time Series Foundation Models under identical data, feature-engineering, and validation conditions also provides additional insight. The results indicate that foundation models should not be assumed to outperform traditional machine learning methods automatically. Their advantage became evident only when environmental covariates and domain adaptation strategies were incorporated into the forecasting workflow.

Consequently, the results suggest that the future value of TSFMs in atmospheric forecasting may depend less on their ability to model temporal sequences in isolation and more

on their capacity to integrate heterogeneous environmental information within flexible forecasting architectures.

5.3. Implications for Environmental Monitoring

The proposed framework demonstrates the practical value of environmental data fusion for atmospheric monitoring and forecasting applications. The methodology successfully integrated heterogeneous datasets originating from fundamentally different observation systems, including EEA ground-monitoring stations, ERA5 meteorological reanalysis products, CAMS atmospheric composition variables, and Sentinel-5P atmospheric observations.

Each data source contributed complementary information. Ground-monitoring stations provided accurate surface-level measurements but limited spatial coverage. Sentinel-5P supplied spatially continuous atmospheric observations, although these represented atmospheric columns rather than direct surface concentrations. ERA5 and CAMS products contributed physically consistent meteorological and atmospheric descriptors capturing transport, dispersion, accumulation, and removal processes. Together, these datasets provided a substantially richer representation of atmospheric conditions than any individual source could offer independently.

An important practical implication follows from this result. Machine learning should not be viewed as a replacement for environmental monitoring networks, but rather as a mechanism for increasing their informational value through integration with broader environmental datasets. This capability is particularly relevant in regions characterized by sparse monitoring coverage, incomplete observational records, or limited environmental infrastructure.

The successful adaptation of Chronos-2 additionally suggests that foundation-model architectures may become increasingly relevant for environmental intelligence applications. Potential applications extend beyond pollutant forecasting and may include missing-data reconstruction, anomaly detection, multi-pollutant estimation, environmental early-warning systems, and large-scale environmental monitoring platforms.

From a geoinformatics perspective, the framework demonstrates the value of combining heterogeneous geospatial datasets within a unified analytical workflow. The integration of satellite observations, atmospheric reanalysis products, atmospheric composition datasets, and ground-monitoring networks required consistent spatial harmonization, temporal synchronization, and environmental data-fusion procedures. These operations constitute a

central geoinformatics contribution of the thesis and provide a reproducible methodological foundation for future environmental intelligence systems operating at regional and continental scales.

5.4. Limitations

Despite the encouraging results obtained in the final forecasting configurations, several limitations should be considered when interpreting the findings.

First, the predictive experiments were conducted exclusively within the Milan Metropolitan Area. Milan provides a suitable and data-rich urban environment for methodological evaluation; however, the reported forecasting performance cannot be assumed to generalize automatically to regions characterized by different climatic conditions, atmospheric regimes, emission structures, or monitoring-network densities. The transferability of the proposed framework across Europe remains an open research question.

Second, Sentinel-5P observations introduce a representativeness limitation because TROPOMI retrieves vertically integrated atmospheric column quantities rather than direct near-surface concentrations. In this thesis, this mismatch was addressed by using satellite retrievals as complementary covariates within a multi-source framework, together with ERA5 meteorological variables and CAMS atmospheric composition products. The improved performance of covariate-rich configurations indicates that these variables provide useful information for SO_2 forecasting. Nevertheless, residual uncertainty remains due to vertical mixing processes, retrieval quality, cloud filtering, spatial footprint mismatch, and the limited temporal sampling imposed by the Sentinel-5P overpass schedule. Therefore, satellite-derived variables should be interpreted as supporting atmospheric predictors rather than direct measurements of surface-level exposure.

Third, and most importantly, the strongest forecasting configurations relied on observed meteorological and atmospheric covariates during the evaluation period. Consequently, the reported results should be interpreted as retrospective upper-bound estimates of predictive capability rather than as fully operational forecasting performance.

In practical forecasting systems, future values of meteorological and atmospheric variables are not available at prediction time. Variables derived from ERA5 reanalysis, CAMS products, and Sentinel-5P observations would therefore need to be replaced by forecasted inputs obtained from numerical weather prediction systems and operational atmospheric composition forecasts. The uncertainty associated with these forecasted covariates would introduce an additional source of error that is not represented in the retrospective results

reported in this thesis.

Finally, transformer-based forecasting architectures require greater computational resources than conventional machine learning models. Although computational requirements remained manageable within the scope of this case study, large-scale deployment involving multiple pollutants, multiple cities, or spatially explicit forecasting systems would require additional computational infrastructure and optimization.

5.5. Future Work

Several research directions emerge naturally from the findings of this thesis.

The most immediate priority is the evaluation of predictive transferability across multiple European cities characterized by different climatic conditions, atmospheric regimes, emission sources, and monitoring-network structures. Such experiments would allow rigorous assessment of spatial generalization and provide insight into the degree of domain adaptation required across heterogeneous atmospheric environments.

A second priority is the transition from retrospective estimation toward fully operational forecasting. Future implementations should replace observed environmental covariates with forecasted meteorological and atmospheric composition variables obtained from operational forecasting systems. This would enable direct assessment of operational forecasting performance and quantify the uncertainty introduced by forecasted environmental inputs.

A third direction involves the development of spatially explicit forecasting architectures. The current implementation operates on a metropolitan-scale aggregated time series and therefore does not explicitly represent interactions between monitoring stations or spatial pollutant transport processes. Graph Neural Networks, Graph Attention Networks, and spatiotemporal transformer architectures represent promising candidates for incorporating spatial relationships directly into the forecasting framework.

Finally, future research should investigate the development of environmental foundation models pretrained directly on atmospheric observations, Earth Observation products, meteorological reanalysis datasets, and chemical transport simulations. Unlike current general-purpose TSFMs, such models would learn representations specifically adapted to atmospheric variability, pollutant transport processes, seasonal cycles, and meteorological forcing mechanisms. The development of environmental foundation models therefore represents one of the most promising long-term directions for scalable environmental forecasting systems.

6 | Conclusions

The central question motivating this research was whether the spatial limitations of conventional air-quality monitoring networks could be partially addressed through the integration of satellite observations, meteorological reanalysis products, atmospheric composition datasets, and machine learning methodologies within a unified environmental framework. The results suggest that such integration can substantially improve surface-level pollutant estimation; however, the improvements observed throughout the experimental progression indicate that predictive performance depends not only on model architecture but also on the quality of environmental data harmonization, feature representation, and the incorporation of physically meaningful atmospheric information. More broadly, the findings reinforce the idea that pollutant concentrations should not be interpreted as isolated temporal signals, but rather as the observable outcome of interacting atmospheric processes operating across multiple spatial and temporal scales.

A principal contribution of this work is the development of a European-scale environmental harmonization framework capable of addressing the spatial, temporal, and metadata inconsistencies that commonly complicate the integration of multi-source atmospheric datasets. Through automated preprocessing, spatial co-location, temporal synchronization, quality control, and feature generation procedures, the framework enables the construction of consistent machine learning datasets from heterogeneous environmental sources. This contribution extends beyond the forecasting experiments themselves, providing a reproducible geoinformatics workflow that can support future environmental monitoring and machine learning applications across different geographic domains. This distinction is important because the primary contribution of the thesis is not the development of a forecasting model for Milan alone, but the design of a scalable environmental data-fusion framework capable of supporting future atmospheric estimation and forecasting applications across heterogeneous European environments.

The experimental framework evolved progressively from classical machine learning baselines toward increasingly sophisticated forecasting configurations incorporating enhanced feature engineering, environmental covariates, AutoML forecasting systems, and Time

Series Foundation Models (TSFMs). This progression enabled a systematic evaluation of the relative importance of feature representation, environmental information, model architecture, and parameter-efficient adaptation strategies. The results demonstrated that forecasting skill emerged progressively as additional atmospheric information was incorporated into the framework, highlighting the importance of environmental context in atmospheric prediction tasks.

Rather than evaluating a single modeling configuration, the research adopted an iterative six-version experimental strategy in which each version addressed limitations identified in the previous one. This progression provided a controlled assessment of how feature engineering, validation methodology, environmental covariates, and foundation-model adaptation influenced forecasting performance.

The results further demonstrate that historical SO_2 observations alone are insufficient for reliable medium-horizon forecasting. All SO_2-only forecasting configurations, including seasonal baselines, AutoML approaches, and Time Series Foundation Models, produced negative R^2 values when restricted to autoregressive pollutant information. This finding confirms that surface-level SO_2 concentrations should not be interpreted as a self-contained temporal process. Instead, pollutant variability is strongly influenced by exogenous atmospheric drivers, including meteorological forcing, atmospheric transport, boundary-layer dynamics, precipitation processes, and regional background atmospheric conditions.

Forecasting performance improved substantially when environmental information was incorporated into the modeling framework. Meteorological variables, atmospheric composition indicators, satellite-derived observations, and engineered temporal features provided complementary information that could not be extracted from pollutant history alone. Consequently, the most successful forecasting configurations were those capable of jointly representing temporal persistence and environmental forcing within a unified predictive framework.

The strongest performance of the entire study was achieved by Chronos-2 with environmental covariates and Low-Rank Adaptation (LoRA) fine-tuning, obtaining an R^2 of 0.976 and an RMSE of 0.134 $\mu g/m^3$ on the held-out test period. These results demonstrate the potential of pretrained temporal foundation models for environmental forecasting when combined with physically meaningful atmospheric predictors and parameter-efficient domain adaptation strategies. Importantly, the superiority of the final Chronos-2 configuration cannot be attributed solely to temporal autocorrelation. The same lag and rolling-window features were available to the best classical machine learning configurations, yet

those models achieved a maximum R^2 of only 0.455. This suggests that the additional predictive skill emerged from the model's ability to jointly represent temporal persistence and environmental forcing through meteorological variables, atmospheric composition indicators, and satellite-derived observations. At the same time, the findings indicate that foundation models do not automatically outperform conventional approaches. Their advantages became evident only after the incorporation of environmental covariates and task-specific adaptation. Nevertheless, because the evaluation relied on observed rather than forecasted covariates, the reported performance should be interpreted as a retrospective upper-bound estimate rather than a fully operational forecasting capability. To facilitate reproducibility and future development, the complete implementation developed during this research has been released through publicly available repositories covering environmental harmonization, statistical analysis, and forecasting experiments [28–30].

An additional contribution of this research is the demonstration that parameter-efficient fine-tuning can successfully adapt large pretrained temporal models to specialized atmospheric applications without requiring full retraining. The successful use of LoRA suggests that similar adaptation strategies may provide a practical pathway for applying future foundation-model architectures within environmental science, where computational resources and labeled datasets are often limited.

Several limitations should be considered when interpreting the results. The predictive experiments were conducted exclusively within the Milan Metropolitan Area, and therefore the reported performance does not necessarily generalize to regions characterized by different climatic conditions, emission regimes, topographic settings, or monitoring-network structures. Furthermore, Sentinel-5P observations represent atmospheric column quantities rather than direct surface-level concentrations, introducing an indirect relationship between satellite observations and the forecasting target. Most importantly, the highest-performing configurations relied on observed meteorological and atmospheric variables during the evaluation period.

These limitations highlight several directions for future research. The most immediate priority is the evaluation of the framework across multiple European cities and atmospheric environments in order to assess spatial transferability and domain adaptation requirements. Additional work is also needed to replace observed covariates with forecasted meteorological and atmospheric composition products, enabling realistic operational forecasting experiments. Beyond these extensions, future investigations may explore spatially explicit learning architectures, graph-based environmental models, multi-pollutant forecasting systems, uncertainty-aware prediction frameworks, and continental-scale environmental forecasting applications.

More broadly, the results suggest that future advances may arise from the development of environmental foundation models pretrained directly on atmospheric observations, Earth Observation products, meteorological reanalysis datasets, and chemical transport simulations. Such models would be exposed during pretraining to the physical structures and variability patterns that characterize atmospheric systems, potentially reducing the need for extensive downstream adaptation and improving transferability across environmental domains.

The challenge that motivated this thesis—transforming sparse and heterogeneous atmospheric observations into reliable estimates of surface-level air quality—remains open at continental scale. Nevertheless, the framework developed here demonstrates that this challenge is tractable when environmental data fusion, physically informed feature engineering, and advanced temporal learning architectures are combined within a reproducible geoinformatics workflow. As Earth Observation systems, environmental monitoring networks, and foundation-model technologies continue to evolve, such integrated approaches are likely to become an important component of next-generation atmospheric intelligence systems.

Bibliography

[1] F. S. d. Almeida, L. F. G. Fiscina, F. P. Silva, A. M. Rocha, J. V. B. Zanotti, and M. M. Futai. Knowledge discovery for interpretable and variability-aware permeability prediction under sparse sampling in tropical weathered soils: An amazon case study. *SSRN Electronic Journal*, 2025. doi: 10.2139/ssrn.6678914. URL `https://ssrn.com/abstract=6678914`.

[2] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, Y. Shen, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024. URL `https://arxiv.org/abs/2403.07815`.

[3] A. A. E. Ayek, M. A. Loho, and S. F. Karmoka. Satellite air quality monitoring: an interactive and accessible tool for spatiotemporal analysis using google earth engine. *Geo-spatial Information Science*, 2025. doi: 10.1080/10095020.2025.2537352. Published online: 03 September 2025.

[4] I. Belachsen and D. M. Broday. Imputation of missing pm2.5 observations in a network of air quality monitoring stations by a new knn method. *Atmosphere*, 13 (1934), 2022.

[5] A. Carbone, R. Restaino, G. Vivone, and J. Chanussot. Model-based super-resolution for sentinel-5p data. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16, 2024. doi: 10.1109/TGRS.2024.3387877.

[6] J. R. Cedeno Jimenez and M. A. Brovelli. Estimating ground-level no2 concentrations using machine learning exclusively with remote sensing and era5 data: The mexico city case study. *Remote Sensing*, 16(17):3320, 2024.

[7] Copernicus Data Space Ecosystem. Sentinel-5p, 2026. URL `https://dataspace.copernicus.eu/data-collections/copernicus-sentinel-missions/sentinel-5p`. Accessed: 2026-05-26.

[8] A. Das, W. Kong, R. Sen, and Y. Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688v4*, 2024.

[9] D. D.Kovács, J. Musial, J. Bojanowski, D. Clarijs, J. de la Mar, and A. Zlinszky. Copernicus data space ecosystem establishes public cloud processing for earth observation data. *Scientific Data*, 13:537, 2026. doi: 10.1038/s41597-026-06765-8.

[10] European Environment Agency. Harm to human health from air pollution in europe: Burden of disease status. Technical report, European Environment Agency, 2025.

[11] A. Fania, A. Monaco, E. Pantaleo, et al. Estimation of daily ground level air pollution in italian municipalities with machine learning models using sentinel-5p and era5 data. *Remote Sensing*, 16(1206), 2024.

[12] E. Gladkova et al. Applying machine learning techniques in air quality prediction. *Transportation Research Procedia*, 63:1999–2006, 2022.

[13] Google Earth Engine Data Catalog. Sentinel-5P OFFL SO2: Offline Sulfur Dioxide. `https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S5P_OFFL_L3_SO2`, 2026. Accessed: 2026-06-30.

[14] N. S. Gupta et al. Prediction of air quality index using machine learning techniques: A comparative analysis. *Journal of Environmental and Public Health*, 2023:4916267, 2023.

[15] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[16] M. A. Javadi, M. Zare Shahne, and Z. Amiri. Integration of sentinel-5p satellite data and machine learning for spatiotemporal prediction of no_2 in delhi: Impacts of covid-19 lockdown. *Atmospheric Pollution Research*, 17(1):102702, 2025. doi: 10.1016/j.apr.2025.102702.

[17] S. Karami, Z. Ghassabi, N. Khoddam, and M. Habibi. Investigating meteorological factors influencing pollutant concentrations and copernicus atmosphere monitoring service (cams) model forecasts in the tehran metropolis. *Atmosphere*, 16(3):264, 2025. doi: 10.3390/atmos16030264. URL `https://doi.org/10.3390/atmos16030264`.

[18] S. R. K. Kottapalli, K. Hubli, S. Chandrashekhara, G. Jain, S. Hubli, G. Botla, and R. Doddaiah. Foundation models for time series: A survey. *arXiv preprint arXiv:2504.04011*, 2025. URL `https://arxiv.org/abs/2504.04011`.

[19] S. Kumar and V. Bhatnagar. A review of regression models in machine learning. *JOURNAL OF INTELLIGENT SYSTEMS AND COMPUTING*, 3:40–47, 12 2022. doi: 10.51682/jiscom.v3i1.30.

[20] J. Li, K. H. Lee, K. Qin, M. S. Wong, P. W. Chan, and Z. Zhang. Synthesis of satellite and ground data provide unique perspectives for discovering the air pollution patterns: A case study in guangdong province, china. *Environmental Pollution*, 264: 114779, 2020. doi: 10.1016/j.envpol.2020.114779.

[21] J. Ma, Z. Li, J. C. Cheng, et al. Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network. *Science of the Total Environment*, 214:116885, 2019.

[22] J. D. Mejía Coronel, G. M. Faican Cabrera, R. Zalakeviciute, C. Matovelle, S. Bonilla, and J. A. Sobrino. Spatio-temporal evaluation of air pollution using ground-based and satellite data during covid-19 in ecuador. *Heliyon*, 10(7):e28152, 2024. doi: 10.1016/j.heliyon.2024.e28152.

[23] A. Moazzam, D. Oxoli, J. R. C. Jimenez, and M. A. Brovelli. Integration of satellite, models and ground sensors for city-scale air quality monitoring through the open data cube. In *ISPRS Archives/GSW25*, 2025.

[24] M. Rajesh, R. Ganesh Babu, and M. Usha. Air quality prediction using machine learning algorithms. *Scientific Reports*, 15(1):14214, 2025.

[25] G. Ravindiran, G. Hayder, K. Kanagarathinam, A. Alagumalai, and C. Sonne. Air quality prediction by machine learning models: A predictive study on the indian coastal city of visakhapatnam. *Chemosphere*, 338:139518, 2023.

[26] A. Rodriguez-Sanchez, J. Santiago, M. Vivanco, B. Sanchez, E. Rivas, A. Martilli, and F. Martín. How do meteorological conditions impact the effectiveness of various traffic measures on nox concentrations in a real hot-spot? *Science of the Total Environment*, 954:176667, 2024.

[27] S. Sameh, A. Zaki, B. Elsaka, A. A. A. Beshr, and A. G. Shehata. Air pollution mapping and monitoring using sentinel-5p data and google earth engine. *The Egyptian Journal of Remote Sensing and Space Sciences*, 29(1):158–178, 2026. doi: 10.1016/j.ejrs.2026.01.008.

[28] C. I. Saud-Miño. Eea-aqa: European environmental harmonization framework. `https://github.com/Saudisis/EEA-AQA`, 2026. GitHub repository, accessed June 2026.

[29] C. I. Saud-Miño. Eea_stats: European air quality statistical analysis framework. `https://github.com/Saudisis/EEA_Stats`, 2026. GitHub repository, accessed June 2026.

[30] C. I. Saud-Miño. So2_milan_prediction: Machine learning and foundation models for atmospheric forecasting. `https://github.com/Saudisis/SO2_Milan_Prediction`, 2026. GitHub repository, accessed June 2026.

[31] Z. Sohrabi and J. Maleki. Fusing satellite imagery and ground-based observations for $pm_{2.5}$ air pollution modeling in iran using a deep learning approach. *Scientific Reports*, 15(1):21449, 2025. doi: 10.1038/s41598-025-05332-2.

[32] S. Tırınk. Machine learning-based forecasting of air quality index under long-term environmental patterns: A comparative approach with xgboost, lightgbm, and svm. *PLoS ONE*, 20(10):e0334252, 2025.

[33] S. Uddin and H. Lu. Confirming the statistically significant superiority of tree-based machine learning algorithms over their counterparts for tabular data. *PLoS ONE*, 19 (4):e0301541, 2024.

[34] J. Veefkind, I. Aben, K. McMullan, H. Förster, J. de Vries, G. Otter, J. Claas, H. Eskes, J. de Haan, Q. Kleipool, et al. Tropomi on the esa sentinel-5 precursor: A gmes mission for global observations of the atmospheric composition for climate, air quality and ozone layer applications. *Remote Sensing of Environment*, 120:70–83, 2012.

[35] H. Virta, I. Ialongo, M. Szeląg, and H. Eskes. Estimating surface-level nitrogen dioxide concentrations from sentinel-5p/tropomi observations in finland. *Atmospheric Environment*, 312:119989, 2023. doi: 10.1016/j.atmosenv.2023.119989.

[36] Y. Wang, R. Lyu, Q. He, T. Cheng, et al. Retrieval of gridded aerosol direct radiative forcing based on multiplatform datasets. *Science of the Total Environment*, 739: 140361, 2020.

[37] P. Wong, C. Wu, et al. Temporal validation machine learning model fitted by four variable selection methods. *Environmental Technology & Innovation*, 37:103996, 2025.

[38] World Health Organization. Health impacts, 2026. URL `https://www.who.int/teams/environment-cliamte.change-and-health/air-quality-energy-and-health/health-impacts`. Accessed: 2026-05-26.

[39] L. Zhang, Z. Lou, Y. Ying, C. Yang, and H. Zhou. Efficient fine-tuning of large language models via a low-rank gradient estimator. *Applied Sciences*, 15(1):82, 2025. doi: 10.3390/app15010082.

List of Figures

List of Tables

Acknowledgements

A mí, porque desde el primer día que dejé mi país para realizar una maestría que sabía que me iba a costar, supe que estaba por afrontar uno de mis más grandes desafíos.

Pero aquí estoy,
escribiendo los agradecimientos para culminar con esta etapa y creer, al fin, lo que el resto creía desde el principio...
que sí puedo.

Al José Enrique, mi esposo y el amor de mi vida,
por cuidarme, apoyarme, creer en mí, mimarme y ser el mejor compañero de vida...
este logro también es tuyo

Al Marc Anthony, mi ba, mi papito bello,
por acompañarme horas al teléfono, mientras me dabas aliento para seguir adelante

A la Lorelaine, mi cuqui, mi mamita hermosa,
por nunca dudar ni dejar de creer en mí y soñar cuando yo no podía

Al Joseph Nicholas, mi ñañito, mi alma gemela,
por darme la guía de cómo atravesar este mundo caótico, mientras crees en mí cuando yo no lo hago

A Juan Manuel, mi hermano, mi príncipe,
porque nunca te faltó decirme “tú puedes ñaña, eres inteligente”

A la Tuca, mi hermanita,
por tu amor y compañía

A la Marfe y la Natalia,
por ser mi familia, mis hermanas, mi casa lejos de casa

A mi familia y amigos,
por su amor y apoyo.

A la Lina,
porque sí.

To professor Vasil Yordanov,
for his guidance, insights, patience, and trust.

Por esto, y mucho más,
a ustedes y a esas personas que me faltan mencionar

Gracias totales,

y que viva el Ecuador.