UDC: 004.93 https://doi.org/10.15407/jai2025.02.048

O. Ostrovska[1], V. Hura[2], R. Shuvar [3], I. Kolich[4]
[1,2,3,4]Ivan Franko National University of Lviv, Ukraine
1, University St., Lviv, 79000
[1] oksana.ostrovska@lnu.edu.ua
[2] volodymyr.gura@lnu.edu.ua
[3]roman.shuvar@lnu.edu.ua
[4]igor.kolych@lnu.edu.ua
[1]https://orcid.org/0009-0006-3376-8448
[2]https://orcid.org/0009-0007-8781-8970
[3]https://orcid.org/0000-0001-6768-4695
[4]https://orcid.org/0009-0000-2140-6665

MATHEMATICAL MODELING AND INTELLECTUAL ANALYSIS OF QUANTITATIVE CHARACTERISTICS OF AIR POLLUTION

Abstract. Atmospheric pollution constitutes a substantial environmental and public health burden globally. Effective mitigation necessitates rigorous quantitative characterization of pollutant concentrations and their complex temporal dynamics, particularly concerning emissions from localized sources. Conventional analytical approaches often exhibit limitations when confronted with the high dimensionality, non-linearity, and stochasticity inherent in atmospheric dispersion processes.

This work reviews and evaluates synergistic methodologies integrating atmospheric dispersion modeling with intelligent data analysis to elucidate quantitative air pollution characteristics. The modeling component specifically addresses advanced Gaussian plume formalisms designed for single-point emission sources (industrial funnels/stacks), incorporating complex physical phenomena such as plume rise dynamics, deposition mechanisms, and potential chemical decay. These physics-based models are employed alongside and often integrated with, intelligent analytical systems leveraging Artificial Intelligence (AI) and Machine Learning (ML) algorithms for predictive modeling, anomaly detection, pattern recognition, and the assimilation of heterogeneous data streams.

The synergistic integration of sophisticated Gaussian models for point sources with AI/ML techniques facilitates enhanced predictive capabilities for downwind pollutant concentrations and deposition fields. Discussion focuses on demonstrable improvements in forecast accuracy compared to baseline models, the ability to resolve complex plume behaviors under varying meteorological regimes, refined source term estimation capabilities, and the robust evaluation of emission control scenarios specifically targeting point sources. The fusion of deterministic dispersion physics, as captured by complex Gaussian formulations, with adaptive, data-driven AI/ML methodologies yields a more potent and nuanced analytical framework than achievable with either approach in isolation.

The integrated application of advanced atmospheric dispersion models, exemplified by complex Gaussian treatments for single-funnel emissions, coupled with intelligent data analysis techniques, represents a significant advancement in the quantitative assessment and prediction of localized air pollution events. This paradigm provides essential tools for scientifically robust impact assessment, regulatory compliance verification, and the optimization of air quality management strategies pertaining to point-source emissions.

Keywords: modeling, gaussian dispersion, point source analysis, machine learning, predictive analytics, intelligent systems, quantitative analysis.

Introduction

The quantitative analysis of atmospheric pollutants — encompassing primary emissions like particulate matter ($PM_{2.5}$, PM_{10}), nitrogen oxides (NO_x), sulfur dioxide (SO_2), carbon monoxide (CO), volatile organic compounds (VOCs), and key secondary products such as tropospheric ozone (O_3) - is fundamental to advancing atmospheric science and informing environmental protection frameworks. These chemical species, originating from heterogeneous anthropogenic and natural sources, exhibit complex temporal distributions governed by emission fluxes, meteorological transport dynamics, multi-phase chemical transformations, and deposition processes detailed in foundational atmospheric science literature [1]. Rigorously elucidating these quantitative characteristics and their underlying mechanisms is essential for validating atmospheric models, assessing human and ecological exposure risks, attributing source contributions, and evaluating the efficacy of air quality management interventions, thereby demanding the sophisticated analytical and predictive methodologies explored within this work.

Pollutant concentrations vary over short

distances and time scales and are strongly influenced by local topography, micrometeorology, and the precise nature of the emission sources, ranging from diffuse area sources to concentrated point sources, and dispersion dynamics are strongly influenced by meteorology [2]. Capturing this heterogeneity requires dense monitoring networks and analytical techniques capable of discerning subtle patterns and non-linear relationships within vast, often noisy, datasets. Traditional statistical methods are often insufficient to fully encapsulate these dynamics, limiting predictive skill and mechanistic understanding.

Atmospheric dispersion modeling provides a physics-based approach to simulate pollutant flow, covering methodologies from Eulerian grid models for regional scales to localized Lagrangian particle models and refined Gaussian plume formalisms [1]. The Gaussian approach, exemplified by regulatory models like AERMOD designed for point source applications, incorporates parameterizations for plume rise, complex terrain effects, chemical decay, and deposition [3]. The accuracy of these deterministic models is linked to the quality of input data and the fidelity of their physical parameterizations, necessitating careful validation against observations, a process highlighted in various model evaluation studies [4].

To complement physics-based models and address limitations in data assimilation and pattern recognition, intelligent analysis techniques rooted in AI and ML serve as highly complementary tools, with numerous applications reviewed in recent literature [5]. These data-driven methods possess significant capabilities in air quality forecasting, anomaly detection, sensor data fusion, data gap filling, and identifying complex source-receptor relationships directly from observational data [5, 6]. Their capacity to learn intricate non-linear patterns renders them powerful tools, particularly when integrated synergistically with traditional modeling approaches in hybrid frameworks - a rapidly developing area of research [7].

System Overview and Components

The effective modeling and intelligent analysis of quantitative air pollution characteristics necessitate an integrated computational system capable of handling diverse data streams, executing complex simulations, applying sophisticated analytical algorithms, and generating actionable outputs. Such a system comprises several interconnected components, orchestrating the flow of information from raw data acquisition through to final analysis and decision support. While the specific implementation details vary based on application scope (regional forecasting and local point-source impact assessment) and available resources, the core functional components remain largely consistent. The architecture emphasizes modularity, allowing for the incorporation or updating of different models, data sources, and analytical techniques. The applicability of such a system is universal, though data availability and specific component configurations need adaptation for urban or industrial regions, such as Variazh's, reflecting local monitoring infrastructure and emission profiles.

Initially, the system focuses on data acquisition and ingestion, gathering essential inputs from heterogeneous sources. This includes real-time and historical measurements of pollutant concentrations ($PM_{2.5}$, PM_{10}) from official ground-based monitoring stations and calibrated low-cost sensor networks. Meteorological data is incorporated from surface weather stations and numerical weather prediction model outputs, providing necessary atmospheric conditions. Ancillary data such as geographic information, temporal indicators, and relevant activity data are also collected. This ingestion process involves standardized formatting, preliminary quality checks, and routing data appropriately.

Following the acquisition, data undergoes crucial preprocessing and management steps to address inconsistencies. This involves data cleaning and validation to identify and handle outliers and missing values, often using techniques ranging from simple interpolation to advanced machine learning-based imputation. Feature engineering creates derived variables pertinent to the analysis, such as temporal lags, pollutant ratios, or atmospheric stability indices. Data fusion and integration techniques harmonize information from sources with

varying spatial and temporal resolutions onto common frameworks using methods like interpolation or aggregation [6]. Robust database systems optimized for large environmental datasets manage the storage and querying needs.

The core atmospheric simulation relies on the modeling component, which executes physics-based calculations of pollutant dispersion, transport, and transformation. For localized assessments targeting specific point or line sources, advanced Gaussian models simulate plume behavior based on source parameters and meteorological inputs [3].

The intelligent analysis engine complements the physics-based simulations, employing AI and ML algorithms. This component utilizes a suite of techniques, including deep neural networks like Long Short-Term Memory (LSTM) units for time-series forecasting [8], and ensemble methods like Random Forests or Gradient Boosting for prediction. Its functions include short-term air quality forecasting, detecting anomalous pollution events, performing model output statistics for bias correction of physical model output, facilitating data assimilation, and extracting insights from complex multi-pollutant datasets [5, 6].

Implementation

The architecture utilizes Microsoft Azure services for scalability, data management, and computational power, representing one modern approach to deploying such systems in contexts like Variazh's urban environments.

The system implementation begins with data ingestion from two primary sources. Firstly, public air quality data or data from external sensor networks are accessed via APIs. An Azure Function triggered on a schedule, receives this data, performs necessary filtering, and ingests it into Azure Data Lake Storage. Secondly, data originates from dedicated microcontroller-based sensor systems using ESP32 modules. These sensors transmit data, including sensor settings or status, via API calls to Azure IoT Hub, which provides secure device connectivity and management. Connection monitoring is a key aspect of this pipeline. Like the first flow, an Azure Function processes the data stream from IoT Hub, filters it, and stores it in the central Azure Data Lake Storage.

Azure Data Lake Storage serves as the unified repository for data from all sources. From this central storage, data flows into Azure Synapse Studio, a comprehensive analytics platform. Within Synapse Studio, further data analysis is conducted to prepare curated datasets suitable for machine learning tasks. The diagram also indicates a separate flow for direct statistical analysis of the stored data, aimed at identifying problems or trends. The entire workflow, including the deployment and management of analysis code, is managed using CI/CD practices, ensuring reproducibility and operational control.

Machine learning analysis represents a core function of this implementation. Within Azure Synapse Studio, prepared datasets feed ML models designed for specific tasks, such as predicting Air Quality Index (AQI) or pollutant time series and performing anomaly detection [5]. A parallel or alternative pipeline involves deploying ML models onto edge computing devices, such as an NVIDIA Jetson Nano, which processes data sourced directly from the Data Lake or specific sensors for localized analysis, prediction, and anomaly detection. This edge component allows processing closer to the data source, reducing latency for specific alerts or actions.

The final stage focuses on visualization and delivering insights. Both the cloud-based ML analysis (via Synapse Studio) and the edge-based analysis (via Jetson Nano) produce outputs such as predicted air quality results and identified anomalies. These results supply visualization dashboards. The dashboards display up-to-date sensor data, the predicted results from the ML models, and highlighted detections of anomalies, providing actionable information regarding air quality status and potential issues [9]. This specific cloud-centric implementation emphasizes automated data handling, ML-driven insights, and accessible visualization, offering a scalable solution for modern air quality monitoring and analysis.

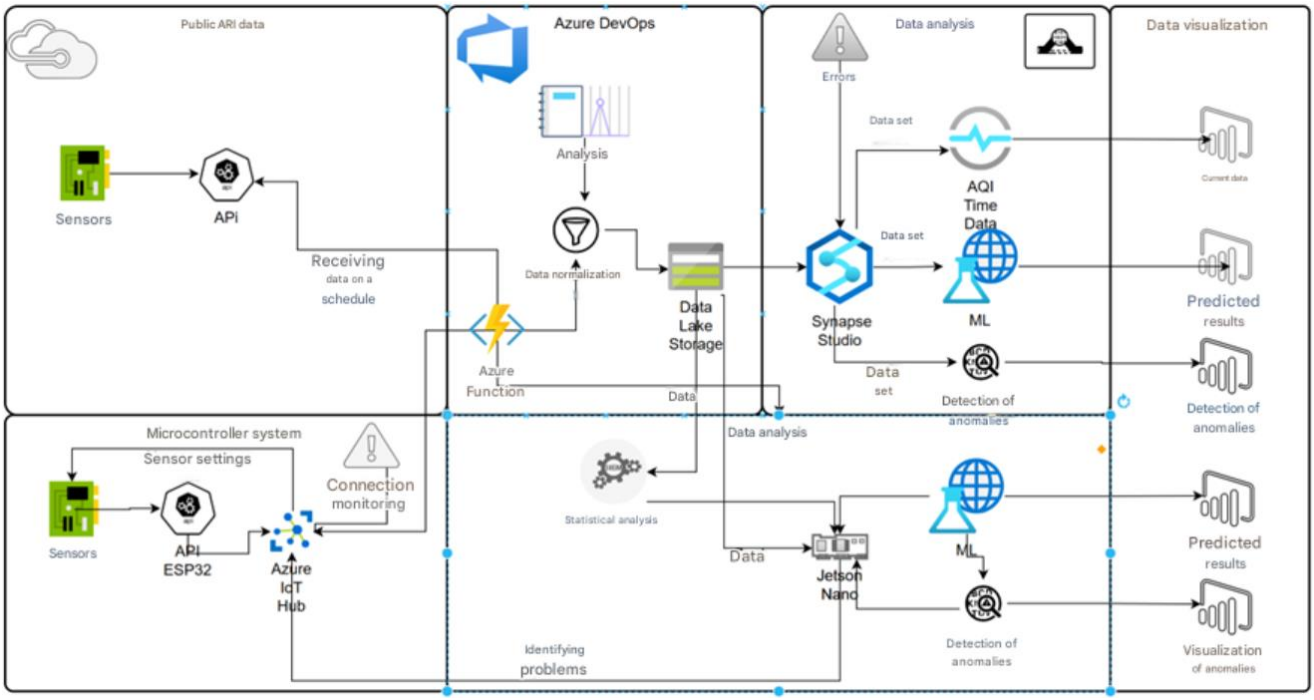

Fig. 1. Architecture of the intelligent air quality monitoring system

Beyond the cloud-centric data handling and machine learning framework described above, a comprehensive air quality analysis system often incorporates physics-based dispersion models. The implementation of such model components involves transforming specific mathematical formulations into executable code. One example is the modified Gaussian plume formula 1, designed to estimate pollutant concentration *C* at a given location *(x, y, z)*

where:

C(x,y,z) is the concentration of the substance emitted at the point with coordinates x, y, z, mg/m³.

Q is the power of a continuous source, mg/s.

u is the wind speed at the height H_{eff}, m/s.

x is the distance from the source, m

y is the transverse distance from the plume axis, m.

z is the height above the ground, m.

H_{eff} is the final plume elevation above the ground (effective plume elevation), m.

$\sigma y(x), \sigma z(x)$ - standard deviations of scattering along the y and z axes [2].

$$C(x,y,z) = \frac{Q}{2\pi u \sigma_y \sigma_z} \cdot \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \cdot \left[\exp\left(-\frac{(z-H_{\text{eff}})^2}{2\sigma_z^2}\right) + \exp\left(-\frac{(z+H_{\text{eff}})^2}{2\sigma_z^2}\right)\right], \quad (1)$$

Considering that the area around the boiler house is flat, the distribution of pollutants in the surface layer of the atmosphere was estimated using the Gaussian model. The concentration of pollutants emitted from a source point is described by equation (1). This equation assumes that the wind direction coincides with the direction of the x-axis and that the origin is at the base of the source (funnel).

There are several special cases of Gaussian models that differ in the way *σy(x)* and *σz(x)* are functionally described. One of the most used models is the Pasquil-Gifford model, which is used to estimate pollution within a 10 km radius of an emission source. The values of the standard deviations depend on the six Pasquil atmospheric stability classes (A-F) in Table 1, which consider different meteorological conditions.

The model incorporates time-varying meteorological conditions, calculates atmospheric stability using Turner's method, estimates plume rise using Briggs formulas, and utilizes Briggs dispersion coefficients. It operates on a 3D grid, optionally includes

background concentrations, and provides an interactive 3D visualization of the concentration field over time. Assessing industrial pollutant impacts utilizes atmospheric dispersion modeling. The computationally efficient Gaussian plume model sees wide use, providing reasonable accuracy in many cases despite simplifications. This program simulates time-varying dispersion by applying the steady-state Gaussian model iteratively, using distinct meteorological data for each hour.

Table 1. Table of Pasquill resistance classes

Wind Speed (m/s)	Daytime - Strong Insolation	Daytime - Moderate Insolation	Daytime - Slight Insolation	Nighttime - >50% Cloud Cover	Nighttime - <50% Cloud Cover
< 2	A	A-B	B	E	F
2-3	A-B	B	C	E	F
3-5	B	B-C	C	D	E
5-6	C	C-D	D	D	D
> 6	C	D	D	D	D

The model consists of several interconnected components: input data processing, atmospheric stability classification, plume rise calculation, dispersion parameterization, and the core Gaussian concentration calculation. Formula 1 requires coding the calculation based on its components: the emission rate *Q*, time-dependent meteorological factors like wind speed *u(t),* and dispersion parameters *σy(x, t*) and *σz(x, t)*.

$$\sigma y = a \cdot x \cdot (1 + b \cdot x)^{-p},$$
$$\sigma z = c \cdot x \cdot (1 + d \cdot x)^{-q}, \quad (2)$$

These dispersion parameters quantify the plume's spread horizontally and vertically based on downwind distance *x* and atmospheric stability conditions prevalent at time t, typically determined using established parameterization schemes Pasquill-Gifford curves, and schemes based on turbulence measurements shown in Table 2. The formula includes standard Gaussian terms for lateral distribution and vertical distribution, incorporating ground reflection relative to the effective stack height H_{eff}.

The height H_{eff} above the ground depends on the atmospheric stability classes A-F and distance from the emission source. For stability classes A-D, this parameter shall be determined as follows

$$H_{eff} = H' + 1.6 \cdot \frac{F_b^{1/3} \cdot x_{max}^{1/3}}{u_{H_{eff}}} \quad if\ x < x_{max},$$
$$H_{eff} = H' + 1.6 \cdot \frac{F_b^{1/3} \cdot x^{1/3}}{u_{H_{eff}}}\ if\ x \geq x_{max} \quad (3)$$

where:

H′ is the modified height of the emission source, m.

Fb is the Briggs parameter.

Xmax is the distance at which the maximum concentration is reached, m.

$$F_b = g \cdot \omega_0 \cdot D^2 \cdot \left(\frac{T_s - T_a}{4 \cdot T_s}\right) > \quad (4)$$

where:

g is the constant for gravitational acceleration, m/s².

ω0 is the initial vertical speed of the emitted gas, m/s.

D is the diameter of the stack opening diameter of the emission source, m.

Ts is the temperature contrast driving buoyancy involves, °C.

Ta is the surrounding air temperature, °C.

The ultimate plume centerline height used for dispersion calculations depends significantly on the interplay between source exit conditions, like the gas outflow rate (Vs), and the ambient wind speed at the source height [2]:

$$H' = H + 2 \cdot D \cdot \left(\frac{\omega_0}{u_{Heff}} - 1.5\right)$$
$$if\ \omega_0 < 1.5 \cdot u_{Heff}, \quad (5)$$

The practical implementation of the Gaussian plume model, specifically its application to simulate pollutant dispersion for the village of Variazh, Ukraine. The model's adaptation to specific conditions relies on several key inputs. Firstly, the precise geographical coordinates of the village (latitude

= '50.518161', longitude = '24.092014'). Secondly, the simulation is driven by time-series meteorological data, including wind speed, direction, and temperature, obtained from a CSV file representing air quality data information for each hour of 2024. Lastly, the framework allows for the optional inclusion of hourly background concentration data for the pollutant being studied (PM2.5), providing a more comprehensive assessment if such data is available and utilized.

Table 2. Pasquill-Gifford coefficients

Pasquill-Gifford classes	a	b	c	d	p	q
A	0.22	0.2	0.0001	0.91	0.87	0.89
B	0.16	0.14	0.0003	0.91	0.78	0.82
C	0.11	0.1	0.001	0.91	0.66	0.74
D	0.08	0.065	0.015	0.86	0.45	0.69
E	0.06	0.05	0.03	0.84	0.44	0.65
F	0.04	0.032	0.004	0.8	0.28	0.61

A specific simulation scenario was configured to demonstrate the model's application for Variazh. The emission source was defined with a physical stack height (*H_stack*) of 12.0 meters and an inner diameter (*ds*) of 0.5 meters. The gas exit velocity (V_s) was set to 5.0 m/s, with a temperature (*Tsccelsius*) of 50.0 °C. The pollutant modeled was PM2.5 (*pollutant*_name = 'PM2.5'), with a constant mass emission rate (*Q_pollutant*) of 10.0 g/s assumed.

The simulation covered the timeframe, starting on *startdatestr* = '2024-01-01 00:00:00' and ending on *end_date_str* = '2024-12-30 23:59:59'. A minimum wind speed threshold (*min_wind_speed_threshold*) of 0.5 m/s was established for the calculations.

The computational grid extended to *x_max* = 1000 meters and *y_max* = 300 meters horizontally from the source, and up to *z_max* = 100 meters vertically. This domain was discretized using *nx* = 40 points along the x-axis, *ny* = 40 points along the y-axis, and *nz* = 25 points along the z-axis. The standard acceleration due to gravity (g = 9.81 m/s²) was used as a physical constant in the plume rise calculations.

The model incorporates several procedures to ensure robust operation and to deal with specific physical or numerical situations. The 3D array storing calculated concentrations is initialized as an array of zeros at the start of the simulation. The plume rise calculation naturally yields zero rise under conditions of neutral or negative buoyancy (T_s <= *T_a*) or zero wind speed, adhering to the physics captured in the Briggs formulas. To avoid singularities or non-physical results near the source where downwind distance approaches zero, the dispersion coefficients (*σy(x)* and *σz(x))* are prevented from falling below a minimum floor value of 0.1 meters.

Furthermore, a threshold is applied to the input wind speed; if a value read from the meteorological data is lower than the specified *min_wind_speed_threshold* (0.5 m/s), the threshold value is used instead. This practice avoids division by zero in the Gaussian equation and represents minimal dispersion during very calm conditions. The definition of the computational space itself involves generating the X, Y, and Z coordinate matrices using NumPy's linspace and meshgrid functions based on the specified grid limits and resolution. Lastly, the addition of background concentration values to the calculated source concentration is conditional, occurring only when the *add_background_concentration* flag is explicitly enabled. While the stability calculation is generally reliable, a potential safeguard could involve defaulting to neutral stability (Class D) if the determination were to fail [10].

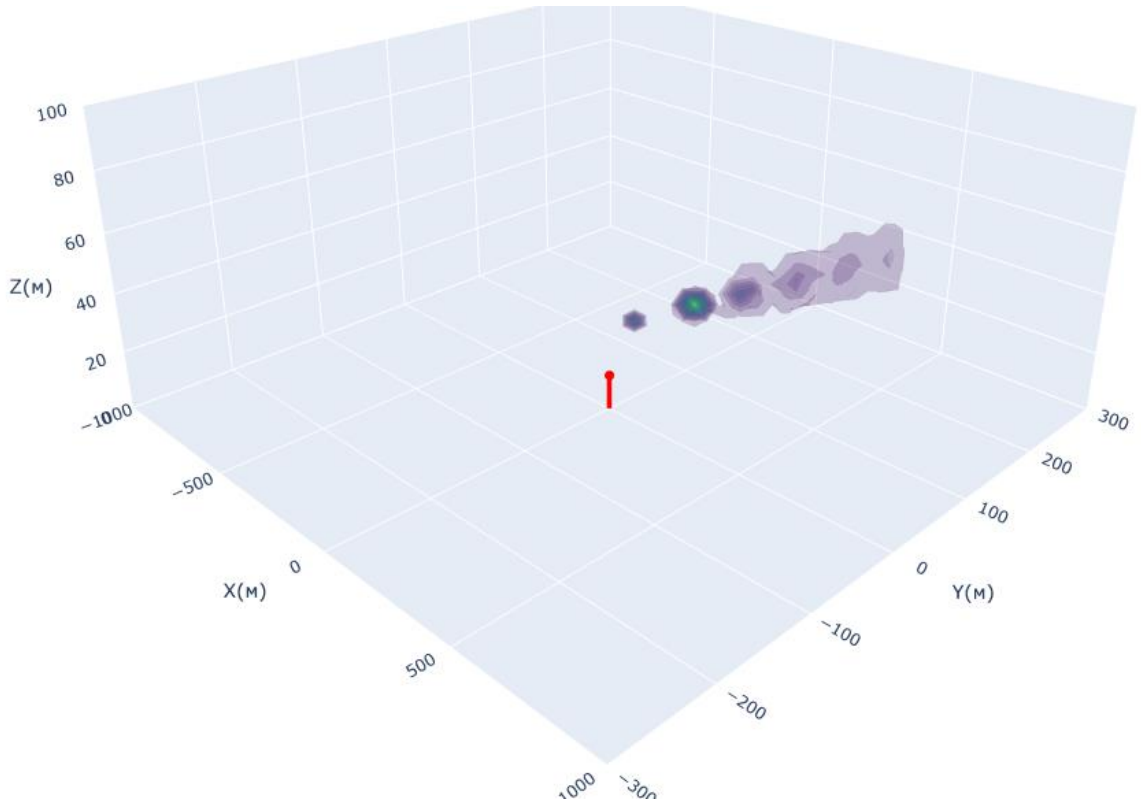

Fig. 2. Modeling atmospheric dispersion with one source

The model's output consists of hourly 3D concentration fields. These results are processed for visualization using the Plotly library. Interactive 3D volume plots are generated, typically rendering surfaces of concentration within the simulation domain. This visual representation facilitates understanding the plume's spatial distribution, extent, and temporal evolution under the influence of the varying meteorological conditions throughout the simulation period. Interactivity is provided through tools like time sliders, allowing users to examine the plume dynamics hour by hour.

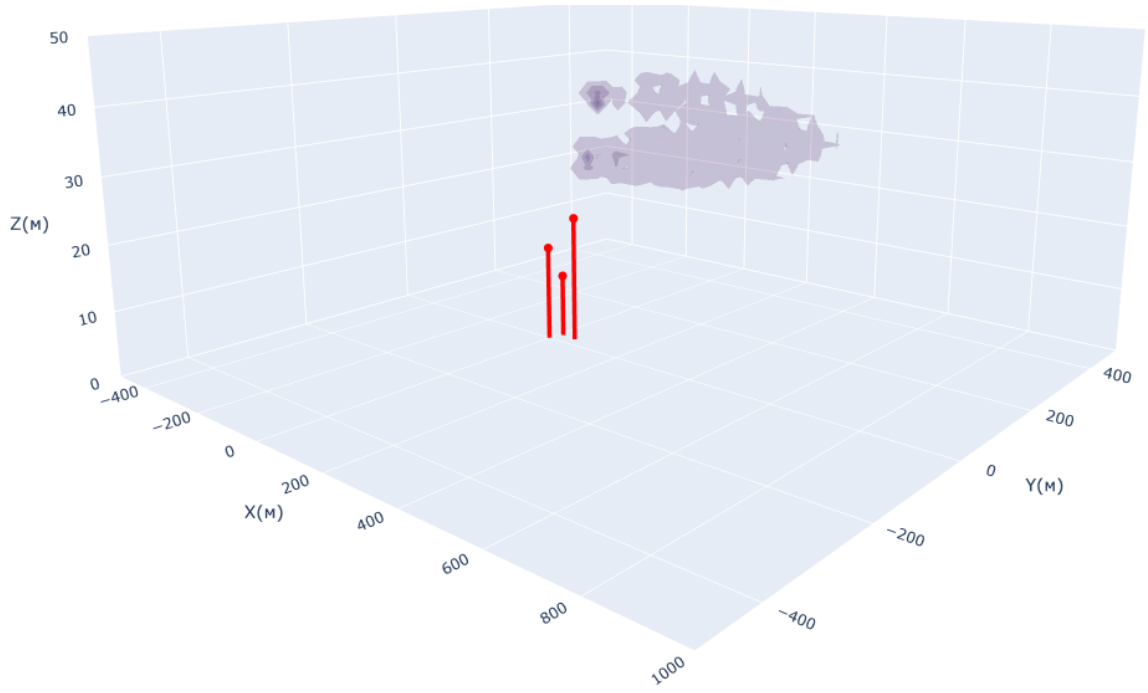

Fig. 3. Modeling atmospheric dispersion with few sources

A primary output required from this model is the estimation of ground-level concentrations (z=0), vital for exposure assessment and comparison with surface monitoring data. Setting z=0 in the modified Gaussian formula simplifies the vertical term, yielding the ground-level concentration formula 6.

$$C(x,y,0) = \left(\frac{Q}{(\pi \cdot u \cdot \sigma_y \cdot \sigma_z)}\right) \cdot exp\left(\frac{-y^2}{(2 \cdot \sigma_y^2)}\right) \cdot exp\left(\frac{-{H_{eff}}^2}{(2 \cdot \sigma_z^2)}\right), \quad (6)$$

While formula 6 represents steady-state conditions, implementing the calculation of hourly ground-level concentrations involves applying this formula repeatedly for each hour *t* within the simulation period. This requires using the specific meteorological inputs corresponding to each hour: hourly average wind speed *u*, hourly prevailing wind direction (used to calculate the crosswind distance y for fixed receptor locations relative to the plume centerline for that hour), and the hourly atmospheric stability class (which determines the appropriate *σy* and *σz* values for that hour [1, 2]).

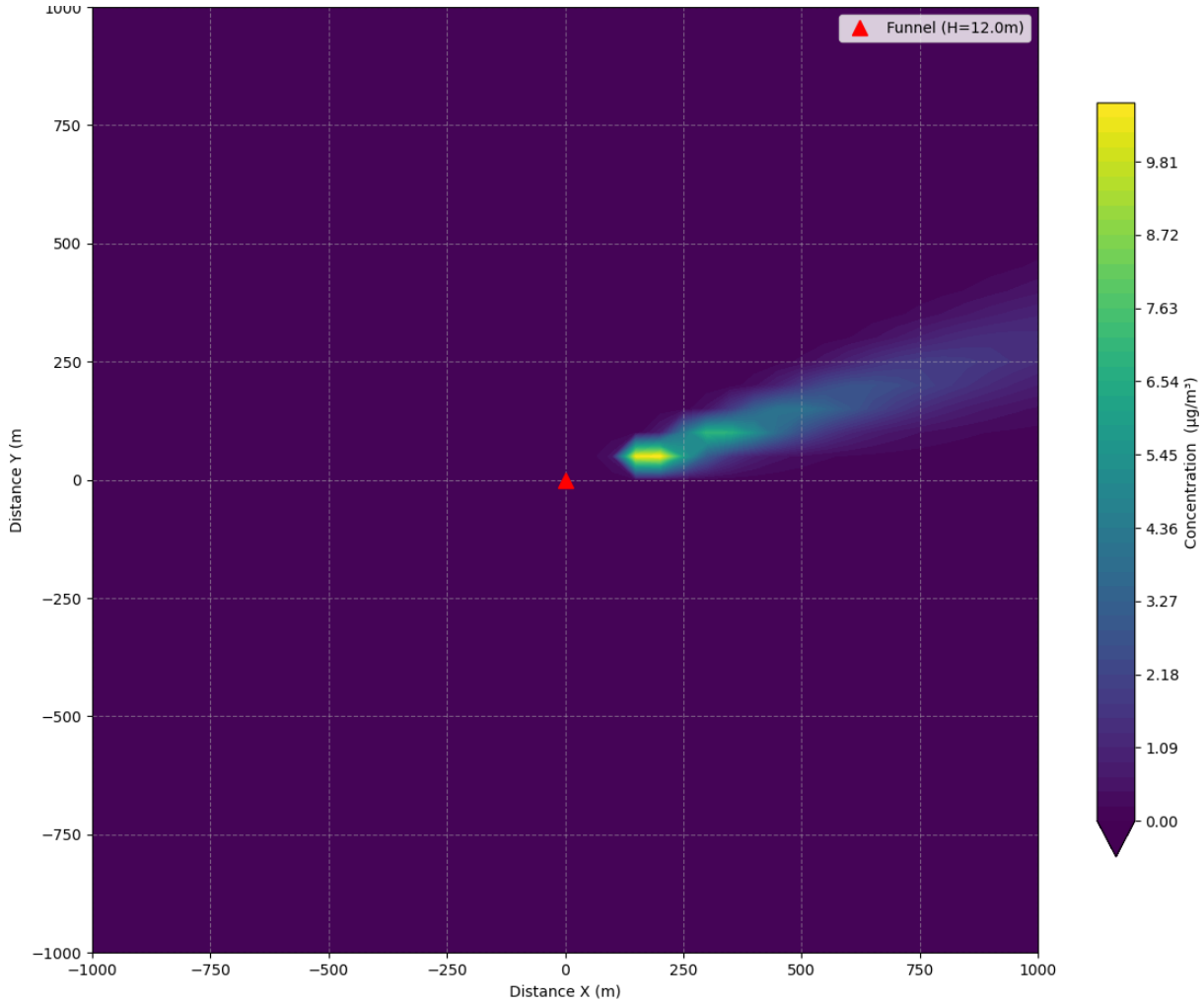

Fig. 4. Modeling atmospheric dispersion in µg/m3 for hour 14, 2024-01-03 ground-level (z=1.0 m)

The output of this iterative hourly calculation is a series of ground-level concentration maps. These hourly ground-level results are outputs that feed directly into the Visualization and Decision Support component [9], enabling detailed analysis of pollution patterns, peak events, exposure, and time-matched validation against hourly ground-based monitor readings [4]. Integrating this standard physics-based calculation provides

essential context and benchmarks within the broader analysis system. Furthermore, these calculated hourly ground-level concentrations for the entire year 2024 are systematically archived into a structured dataset. This historical dataset is specifically prepared to serve as an input feature set or validation benchmark for subsequent development and application of forecasting models, particularly those employing machine learning approaches [5, 7, 8] aimed at predicting future air quality conditions.

Predictive Analytics, Machine Learning, and Future Enhancements

The modeling of a high-resolution temporal dataset, containing hourly values of modeled Gaussian concentration (concentration), comprehensive meteorological parameters (temperature, wind speed, wind direction, wind gusts, humidity, pressure) on a 50-meter grid within a 2km x 2km study area for 2024, provides a unique foundation for advanced predictive analytics and machine learning applications. This dataset serves as the primary input for subsequent analysis aimed at improving forecasting accuracy, understanding model performance, and identifying pollution dynamics within the specific modeled locale around the source point in Variazh (latitude = '50.518161', longitude = '24.092014').

Predictive analytics and machine learning offer a transformative capability to extract actionable insights from the generated temporal dataset. These methods enable the development of advanced forecasting systems, enhance our understanding of pollution dispersion dynamics, and support data-driven decision-making for environmental management. By leveraging this dataset, which merges precise spatial data (x, y) with meteorological parameters (`temperature`, `windspeed`, `wind direction`, `wind gusts`, `humidity`, `pressure`), we can explore and model complex relationships.

Machine learning (ML) models are particularly well-suited for tasks where underlying relationships are highly non-linear, involve multiple interacting variables, or are difficult to define explicitly. In this context, ML can improve the prediction of future pollutant concentrations (`concentration`) by learning from historical patterns embedded in the dataset. The inclusion of meteorological features allows ML models to account for how weather conditions such as wind direction or temperature influence pollutant dispersion [11,12].

Unlike traditional Gaussian plume models, which require deterministic parameterization of dispersion factors, machine learning models can learn and generalize from historical data to make future predictions. These models, trained on historical hourly patterns, can forecast concentration values for each coordinate using the meteorological inputs as predictors.

Methodologically, the 2024 hourly dataset for the selected point, after calculation of the residual and handling any missing values, is divided sequentially into training first 80%) and testing (last 20%) periods. Input features are scaled using standardization (StandardScaler) fitted on the training data to prepare them for model ingestion. A suite of standard regression algorithms is employed for comparative analysis [5], including XGBoost, Random Forests, and MLPRegressor. Each algorithm is trained on the scaled training features to predict the target (*y_train*).

The performance of each regression model in predicting concentration is assessed on the test set (X_test_scaled, y_test) using standard metrics: Mean Absolute Error (MAE) and the coefficient of determination (R^2). This evaluates how well each model captures the Gaussian model's error patterns.

Table 3. Regression models

Model	MAE Mean	MAE Lower	MAE Upper	R^2 Mean	R^2 Lower	R^2 Upper	Time Taken (s)
XGBoost	1.23	1.22	1.25	0.79	0.79	0.80	33.27
Random Forest	1.23	1.22	1.23	0.74	0.74	0.75	1509.46
MLPRegressor	1.51	1.47	1.55	0.70	0.68	0.71	3703.35

The results of Table 3 demonstrate clear trade-offs between computational efficiency and predictive performance across the three models. XGBoost, an ensemble learning method based on gradient boosting with default parameters emerged as the most accurate and computationally efficient model, achieving an MAE of 1.23 and an R^2 of 0.79. Its training time of just 33 seconds was significantly faster compared to the other models, making it an ideal choice for real-time applications or scenarios requiring quick predictions. XGBoost's ability to balance speed and accuracy highlights its suitability for tasks involving atmospheric dispersion bias prediction.

Random Forest, a tree-based ensemble method that averages predictions from multiple decision trees with default parameters performed similarly to XGBoost in terms of MAE (1.23) but achieved a slightly lower R^2 of 0.74, indicating reduced accuracy in capturing error patterns. Its training time of 1509 seconds was substantially longer, making it less practical for time-sensitive tasks. Random Forest remains a strong option for scenarios where computational cost is less of a concern, as its tree-based structure provides interpretability and robustness in modeling non-linear relationships.

MLPRegressor, a neural network model designed to capture complex non-linear relationships with default parameters exhibited the lowest performance among the three models, with an MAE of 1.51 and an R^2 of 0.70, reflecting reduced accuracy compared to XGBoost and Random Forest. Its training time of 3703 seconds was the longest, underscoring the computational demands of neural networks. While MLPRegressor is capable of modeling intricate patterns, its lower accuracy and higher computational cost make it less suitable for this specific use case.

The performance of the regression models was further evaluated for a single location point to assess their ability to predict concentration in a localized setting. The results for this specific location are summarized in Table 4.

Table 4. Regression models for one point

Model	MAE Mean	MAE Lower	MAE Upper	R^2 Mean	R^2 Lower	R^2 Upper	Time Taken (s)
XGBoost	2.995	2.171	3.819	-1.454	-3.748	0.841	1.20
Random Forest	2.944	2.181	3.708	-1.231	-2.962	0.501	8.98
MLPRegressor	3.006	2.236	3.775	-1.392	-3.401	0.617	7.12

For this specific location point, the models exhibited reduced predictive performance compared to the results obtained across all location points. This is likely due to the smaller dataset size and the absence of shared patterns across multiple locations, which the models leveraged in the global analysis.

XGBoost achieved an MAE of 2.995 and an R^2 of -1.454, with a training time of just 1.20 seconds. While XGBoost remained computationally efficient, its negative R^2 indicates that the model struggled to explain the variance in the target variable for this single location. This suggests that XGBoost's strength in capturing global patterns across multiple locations does not translate as effectively to localized predictions.

Random Forest performed slightly better than XGBoost in terms of MAE, achieving a value of 2.944, but its R^2 of -1.231 still indicates poor predictive performance. The training time of 8.98 seconds was significantly higher than XGBoost, reflecting the computational cost of building multiple decision trees. Random Forest's ability to model non-linear relationships may have contributed to its slightly better MAE, but its overall performance remained suboptimal for this location.

MLPRegressor, with an MAE of 3.006 and an R^2 of -1.392, exhibited the lowest performance among the three models. Its training time of 7.12 seconds was longer than XGBoost but slightly shorter than Random Forest. While MLPRegressor is capable of modeling complex non-linear relationships, its performance in this localized setting suggests that it may require more data or additional

tuning to achieve competitive results.

These results of Table 5 highlight the challenges of predicting for a single location point. The negative R^2 values across all models indicate that none of the models were able to effectively capture the variance in the target variable for this specific location. This suggests that localized predictions may require additional feature engineering, such as incorporating location-specific environmental factors or temporal patterns, to improve model performance. Furthermore, the relatively high MAE values indicate that the models may benefit from a larger dataset or more granular data for this location.

To improve the predictive performance of the models, lagged features were introduced to capture temporal dependencies in the data, and an LSTM (Long Short-Term Memory) model was added to the evaluation. Lagged features provide the models with historical information about concentration, enabling them to better predict future values. The results, summarized in the table below, demonstrate that incorporating lagged features significantly improved the models' ability to capture temporal patterns, particularly for the LSTM model.

Table 5. Regression models for one point with lagged feature

Model	MAE Mean	MAE Lower	MAE Upper	R^2 Mean	R^2 Lower	R^2 Upper
XGBoost	2.34	1.38	3.29	-0.17	-0.83	0.48
Random Forest	2.26	1.35	3.17	-0.13	-0.68	0.41
MLPRegressor	2.39	1.64	3.13	-0.43	-1.48	0.63
LSTM	2.03	1.30	2.75	0.08	-0.39	0.55

The addition of lagged features and the inclusion of the LSTM model resulted in notable improvements in predictive performance. Across all models, the use of lagged features helped capture temporal dependencies, reducing the MAE and improving the R^2 scores compared to the results without lagged features.

LSTM emerged as the best-performing model, achieving the lowest MAE of 2.03 and the highest R^2 of 0.08, with an upper bound of 0.55. This demonstrates the LSTM's ability to effectively model sequential data and leverage temporal patterns. The improved performance highlights the importance of using models specifically designed for time-series data when lagged features are introduced.

Random Forest Default and XGBoost Default also benefited from the inclusion of lagged features, achieving MAE values of 2.26 and 2.34, respectively, and modest improvements in R^2 scores. However, their performance remained lower than that of LSTM, indicating that tree-based models, while effective, may not fully capture the sequential nature of the data.

MLPRegressor, while capable of modeling non-linear relationships, exhibited the lowest performance among the models, with an MAE of 2.39 and an R^2 of -0.43. Despite the inclusion of lagged features, MLPRegressor struggled to match the performance of LSTM and tree-based models, likely due to its sensitivity to hyperparameter tuning and lack of inherent temporal modeling capabilities.

The analysis across all locations, a single location, and with the addition of lagged features demonstrates clear trends in model performance. Results of prediction for the next 24 hours using XGBoost model regressor are shown in Figure 5.

For all locations, XGBoost provided the best balance between speed and accuracy, making it the most suitable model for global predictions. For a single location, the models struggled to generalize, with Random Forest slightly outperforming XGBoost in terms of MAE. The introduction of lagged features and the use of LSTM significantly improved performance, with LSTM achieving the best results overall. These findings highlight the importance of incorporating temporal dependencies and using time-series-specific models like LSTM for tasks involving sequential data.

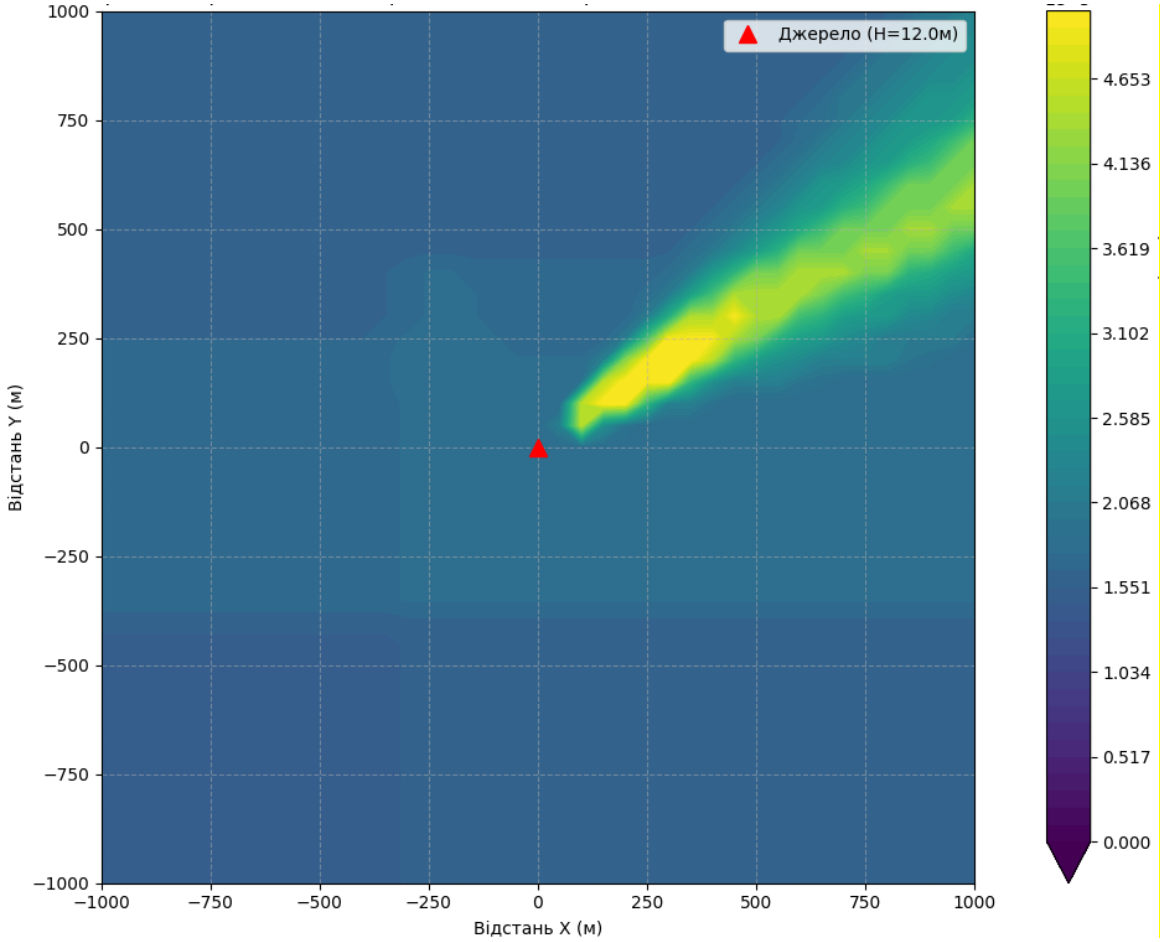

Fig. 5. Prediction atmospheric dispersion in µg/m3 for 2024-12-31 ground-level (z=1.0 m) with XGBoost model

Maintenance, Calibration, and Reliability

The successful implementation of an integrated system for air quality modeling and analysis, such as the one developed for the Variazh school area, represents only the initial phase. Ensuring the system consistently delivers accurate, reliable, and actionable information over its operational lifetime necessitates rigorous, ongoing maintenance, periodic calibration of its components, and continuous assessment of its overall reliability. These activities are fundamental for maintaining user trust and the scientific validity of the system's outputs, particularly when used for regulatory compliance, public health advisories, or policy evaluation.

A good aspect involves the maintenance and calibration of the sensor network, which forms the primary observational input. Physical sensors deployed in the field, whether reference-grade monitors or low-cost sensors feeding into systems like Azure IoT Hub require regular physical checks, cleaning, power management, and potential component replacement due to environmental exposure or hardware degradation. Calibration is paramount: reference monitors must adhere to strict schedules involving zero/span checks and multi-point calibrations against traceable standards. Low-cost sensors like ESP32, known for potential drift and cross-sensitivities, demand even more attention, typically requiring robust field calibration routines, co-location studies with reference instruments, or the application of ML-based calibration algorithms [6], which themselves need periodic validation and retraining. Continuous automated and manual data quality control procedures are essential to flag sensor malfunctions or data anomalies promptly.

Equally important is the maintenance of the modeling components. Physics-based models, such as the implemented Gaussian model [2], depend on accurate inputs. This requires periodic updates to emission inventories to reflect changes in source activity [9], revisions to land use or topographical data

if significant changes occur, and ensuring the continuous supply of validated meteorological data. Model code itself requires updates based on scientific advancements or software revisions [4]. For the machine learning components (bias correction regressors, forecasting models [5, 7, 8]), periodic retraining is essential. The optimal retraining frequency depends on factors like observed performance degradation, seasonality, significant changes in emission patterns, or updates to input data streams. Monitoring for concept drift - where the statistical properties of the input data or the underlying relationships change over time - is crucial to trigger necessary model updates or architectural revisions [13].

The underlying software and computational infrastructure also demand regular maintenance. This includes updating operating systems, databases, core libraries (Python, TensorFlow, scikit-learn), and specific software packages (Azure service SDKs, modeling software). Dependency management and comprehensive version control for all custom code (data processing scripts, model implementations, ML training pipelines) are vital for reproducibility and managing updates. Security patching is also a critical routine task.

Ensuring system reliability involves continuous monitoring and evaluation. Automated workflows compare system outputs (hourly corrected concentrations, forecasts) against incoming real-time observations, tracking key performance metrics (MAE, bias, R^2) over time [4]. Deviations beyond predefined thresholds trigger alerts for investigation. Visualization dashboards [9] should include diagnostic plots monitoring data completeness, model performance trends, and system health indicators. Implementing redundancy for critical data feeds or processing components can enhance resilience. Thorough documentation covering system architecture, data flows, calibration protocols, model versions, and maintenance logs is indispensable for long-term operation and knowledge transfer.

These maintenance, calibration, and reliability assurance activities present significant ongoing operational challenges, requiring dedicated resources and skilled personnel. Prioritization of maintenance tasks and robust contingency planning become even more critical under such circumstances. Nonetheless, commitment to these processes is non-negotiable for ensuring the sustained accuracy and credibility of the air quality information provided by the integrated system.

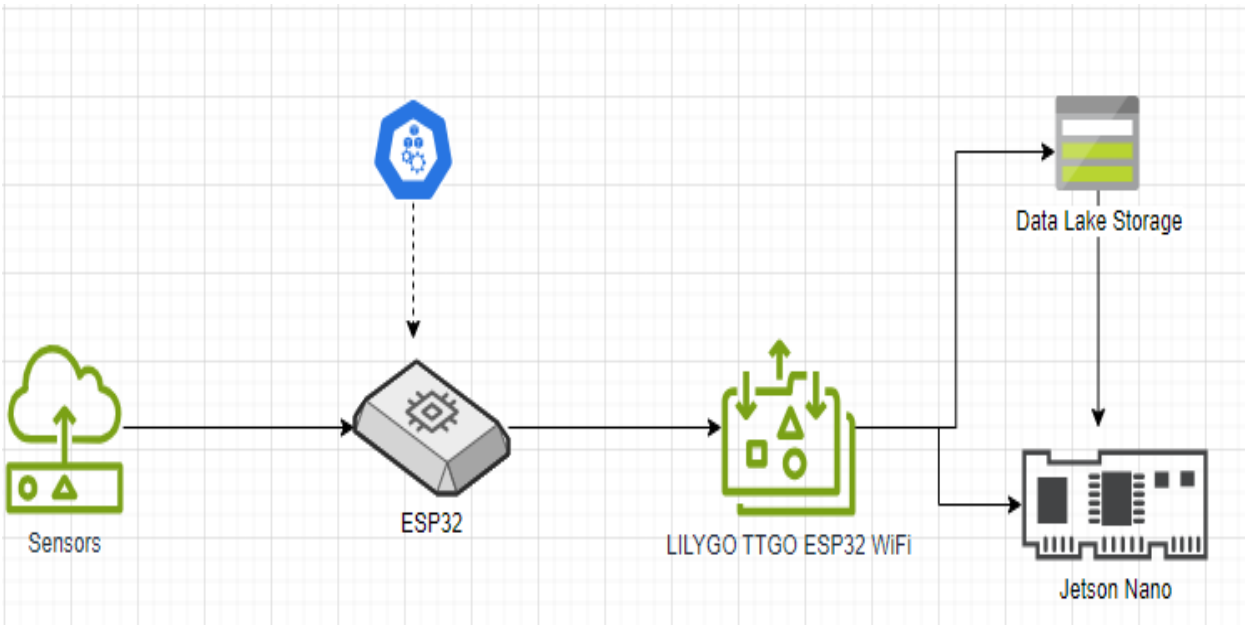

Fig. 6. Architecture of the neurocontroller hardware and software system

These maintenance, calibration, and reliability assurance activities present

significant ongoing operational challenges, requiring dedicated resources and skilled personnel. In specific contexts, such as Variazh's school area within the Lviv region as of April 2025, factors like potential constraints on funding, logistical challenges, and ensuring stable power and internet connectivity are critical. Specifically, maintaining the integrity of the physical sensor deployments, including the protective enclosures used in the field (Figure 7), and ensuring the continued functionality of the embedded hardware components such as the ESP32 microcontroller system and LILYGO board responsible for data acquisition and transmission (Figure 6), are vital hands-on maintenance tasks. Similarly, the edge processing unit Jetson Nano requires software updates and operational checks if deployed locally. These factors, potentially exacerbated by the broader situation in Ukraine, add layers of complexity to ensuring data continuity and system uptime. By adhering to these recommendations and routinely performing calibration and maintenance tasks, the IoT-based air quality monitoring system can operate effectively, delivering accurate and reliable air quality data that informs and empowers individuals and communities to make proactive decisions about their environment.

Fig. 7. The physical deployment of sensor hardware

Conclusion

This work detailed an integrated approach for the quantitative analysis and prediction of air pollution characteristics, emphasizing the synergy between physics-based atmospheric dispersion modeling and data-driven intelligent analysis using machine learning. A specific implementation pathway was explored, grounded in the generation and utilization for the area around a source near Variazh school area for the year 2024. This dataset, containing modeled Gaussian concentrations alongside observed pollutants (PM2.5, PM10), AQI, and comprehensive meteorology, served as a foundation for advanced analytics.

The analysis demonstrated the effectiveness of employing various machine learning regression models—XGBoost, Random Forest, MLPRegressor, LSTM—for the task of Gaussian model bias correction. Comparative evaluation (illustrated conceptually in Table 5) highlights the potential of these data-driven techniques, particularly non-linear algorithms, to significantly enhance the accuracy of physics-based predictions by learning and correcting systematic model errors specific to the local

microenvironment. Furthermore, the potential for extending predictions across the entire high-resolution grid using advanced models like XGBoost and LSTM.

Architectural examples, such as the outlined Azure-based system utilizing IoT Hub, Data Lake Storage, Synapse Studio, and DevOps, illustrate how modern cloud platforms facilitate the implementation of such complex, data-intensive environmental analysis systems. Complementing the cloud infrastructure, the specific hardware architecture employed leverages ESP32-based microcontrollers for sensor data acquisition (as shown in Figure 6). Data processing and connectivity are managed by boards like the LILYGO TTGO ESP32 WIFI, which serves a dual function: synchronizing sensor readings with cloud resources like Azure Data Lake Storage managed via Azure IoT Hub and feeding data to a dedicated edge computing unit. This edge processing is handled by an NVIDIA Jetson Nano, enabling real-time local analysis, machine learning inference (prediction, anomaly detection), and neurocontroller functions directly within the monitored environment, while still interfacing with cloud storage for historical data or model updates.

The synergistic combination of physics-based atmospheric modeling and adaptive machine learning techniques, powered by high-resolution datasets and implemented through integrated cloud and edge computing architectures, represents a powerful and evolving paradigm. It offers significantly improved capabilities for understanding source impacts, predicting local air quality dynamics, and ultimately supporting more effective, evidence-based air quality management strategies in complex urban and industrial environments.

References

1. Seinfeld, J. H., & Pandis, S. N. (2016). *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change* (3rd ed.). Wiley.

2. Kahl J. D. W., Chapman H. L. Atmospheric stability characterization using the Pasquill method: A critical evaluation. *Atmospheric Environment*. 2018. Vol. 187. P. 196–209.
URL: https://doi.org/10.1016/j.atmosenv.2018.05.058

3. AERMOD: A Dispersion Model for Industrial Source Applications. Part I: General Model Formulation and Boundary Layer Characterization / A. J. Cimorelli et al. Journal of Applied Meteorology. 2005. Vol. 44, no. 5. P. 682–693. URL: https://doi.org/10.1175/jam2227.1

4. Lohmann L. The Dyson effect: carbon "offset" forestry and the privatisation of the atmosphere. International Journal of Environment and Pollution. 2001. Vol. 15, no. 1. P. 51.
URL: https://doi.org/10.1504/ijep.2001.000591

5. Real-time on-site monitoring of soil ammonia emissions using membrane permeation-based sensing probe / M. Zhou et al. Environmental Pollution. 2021. Vol. 289. P. 117850.
URL: https://doi.org/10.1016/j.envpol.2021.117850

6. Long-term trends in atmospheric Quercus pollen related to climate change in southern Spain: A 25-year perspective / R. López-Orozco et al. Atmospheric Environment. 2021. Vol. 262. P. 118637.
URL: https://doi.org/10.1016/j.atmosenv.2021.118637

7. Martin R. V. Satellite remote sensing of surface air quality. Atmospheric Environment. 2008. Vol. 42, no. 34.P. 7823–7843.
URL: https://doi.org/10.1016/j.atmosenv.2008.07.018

8. Gridded emissions of air pollutants for the period 1970–2012 within EDGAR v4.3.2 / M. Crippa et al. Earth System Science Data. 2018. Vol. 10, no. 4. P. 1987–2013. URL: https://doi.org/10.5194/essd-10-1987-2018

9. Fully coupled “online” chemistry within the WRF model / G. A. Grell et al. Atmospheric Environment. 2005. Vol. 39, no. 37. P. 6957–6975.
URL: https://doi.org/10.1016/j.atmosenv.2005.04.027

10. Description and evaluation of the Community Multiscale Air Quality (CMAQ) modeling system version 5.1 / K. W. Appel et al. Geoscientific Model Development. 2017. Vol. 10, no. 4. P. 1703–1732.
URL: https://doi.org/10.5194/gmd-10-1703-2017

11. Aerosol particle mixing state, refractory particle number size distributions and emission factors in a polluted urban environment: Case study of Metro Manila, Philippines / S. Kecorius et al. Atmospheric Environment. 2017. Vol. 170. P. 169–183.
URL: https://doi.org/10.1016/j.atmosenv.2017.09.037

12. Raissi M., Perdikaris P., Karniadakis G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics. 2019. Vol. 378. P. 686–707.
URL: https://doi.org/10.1016/j.jcp.2018.10.045

13. Developing an integrated technology-environment-economics model to simulate food-energy-water systems in Corn Belt watersheds / S. Li et al. Environmental Modeling & Software. 2021. Vol. 143. P. 105083.
URL: https://doi.org/10.1016/j.envsoft.2021.105083

The article has been sent to the editors 13.04.25.
After processing 24.04.25.
Submitted for printing 30.06.25.