Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
2025, vol. 30, no. 1, pp. 195–201
https://doi.org/10.2478/acss-2025-0021
https://content.sciendo.com

RIGA TECHNICAL UNIVERSITY

Scenario Modelling and Impact Estimation of a Local Pollutant on the Environment

Volodymyr Hura[1*], Oksana Ostrovska[2] and Vasyl Lyashkevych[3]
[1–3]*Faculty of Electronics and Computer Technologies, Ivan Franko National University of Lviv, Lviv, Ukraine*

Abstract **– The growing problem of air pollution by fine particulate matter (PM2.5) from local sources, such as boiler houses or small industrial facilities, requires effective and accessible assessment tools. A significant gap exists between complex, resource-intensive dispersion models used in research and the practical needs of engineers, ecologists, and regulatory bodies who require instruments for rapid operational analysis. This problem is particularly acute in regions like Ukraine, where access to real-time, high-resolution environmental data is limited, and regulatory practices often rely on legacy methodologies. The paper describes the development and testing of a desktop software application with a graphical user interface (GUI) designed for scenario modelling ("what if" analysis) and quantitative assessment of air pollution levels from a local source. The core of the software tool is based on an adapted Gaussian plume analytical model, which calculates pollutant dispersion considering meteorological conditions and source parameters. The system integrates a developed method for integral impact assessment, categorising the pollution level based on calculated concentrations. The developed software allows the user to interactively input the constructive (stack height, diameter) and operational (emission rate) parameters of a pollution source, as well as current meteorological conditions. The system provides an instantaneous calculation of the expected PM2.5 concentration at a given point and classifies the impact: "Low", "Moderate", "High", or "Very High". The developed tool brings practical value, supporting the decision-making process. It provides a means for the operational monitoring of environmental impact and the preliminary planning of measures to reduce the ecological load from local pollution sources, making complex analysis accessible to a wider range of specialists, especially in data scarce environments.**

Keywords **– Air quality, decision support system, graphical interface, Gaussian model, impact assessment, local source, modelling scenarios.**

I. Introduction

The pollution of atmospheric air with fine particulate matter (PM2.5) is recognised as one of the most significant environmental health risks globally. Due to their microscopic size, these particles can bypass the body's natural defences, penetrating deep into the respiratory tract and even entering the bloodstream. This exposure is strongly linked to severe health outcomes, including cardiovascular and respiratory diseases, lung cancer, and increased mortality rates, a fact consistently highlighted by the World Health Organization [1]. While large industrial pollutants are often subject to stringent regulation and continuous monitoring, the cumulative impact of numerous smaller, local sources such as municipal boiler houses, transport hubs, and small manufacturing facilities is substantial yet frequently underestimated. These sources are often located near residential areas, creating zones of elevated risk that are not adequately captured by sparse state-level monitoring networks. This disparity creates a critical problem of data incompleteness and uncertainty, hindering effective risk assessment and the development of targeted mitigation strategies for local communities.

The problem of data scarcity is particularly acute in the context of Ukraine and many other post-Soviet countries. The state environmental monitoring system relies on a sparse network of stationary posts, often equipped with outdated infrastructure, which provides limited spatial and temporal resolution [2]. For local sources, crucial operational data, such as real-time emission rates, exit gas velocity, and temperature, are rarely available through automated systems. Instead, these parameters are typically determined through periodic, manual measurements or engineering estimates, which may not reflect the dynamic nature of the source's operation.

Furthermore, the regulatory framework for air dispersion modelling in Ukraine has historically been based on legacy methodologies inherited from the Soviet era, such as the OND86 standard [3]. While robust for their time, these standards often involve complex manual calculations, are less flexible for modern computational environments, and are not always well adapted for assessing pollutants like PM2.5. This creates a significant barrier for local engineers and ecologists who need to perform rapid, modern assessments but lack both the granular data and the accessible tools to do so. This regional context strongly motivates the development of tools that can function effectively in a data limited environment, enabling users to leverage the best available data (even if manually collected) for meaningful scenario analysis.

To address the challenge of quantifying pollution dispersion, a hierarchy of modelling tools has been developed. At the highest level of complexity are advanced numerical models, such as Computational Fluid Dynamics (CFD) and mesoscale meteorology chemistry models like WRF Chem (Weather Research and Forecasting with Chemistry) [4]. These models

* Corresponding author's e-mail: volodymyr.gura@lnu.edu.ua
Article received 2025-11-16; accepted 2025-12-04

provide high-fidelity, three-dimensional simulations of atmospheric processes but demand significant computational power, extensive input data, and specialised expertise, rendering them impractical for routine operational assessments by non-specialists.

A more widely used category includes regulatory dispersion models, such as AERMOD and CALPUFF, which are predominantly based on the Gaussian plume formulation [5]. These models serve as the standard for environmental impact assessments in many countries. However, they still require a considerable amount of detailed source and meteorological data (e.g., hourly meteorological observations, surface characteristics, emission profiles), and their proper application necessitates specific training.

Consequently, a clear gap exists between the capabilities of these sophisticated models and the practical needs of onsite engineers, environmental inspectors, and municipal authorities. These specialists require accessible “screening models” or Decision Support Systems (DSS) that are fast, intuitive, and capable of providing reliable, express assessments. Such tools are essential for conducting scenario analysis (e.g., “what if” scenarios for changing operational parameters or meteorological conditions) and making informed, timely decisions without the overhead of complex modelling systems. The lack of such user-friendly, scientifically grounded tools represents a significant barrier to proactive environmental management at the local level.

To bridge this identified gap, this paper introduces the development and demonstration of a desktop software application with an intuitive graphical user interface, specifically designed as a Decision Support System for the scenario analysis of the impact of a local pollution source. The primary contribution of this study is the creation of a self-contained tool that translates a robust physical model into an accessible format for a non-expert user base.

The main objectives of this paper are as follows:

- to implement a robust calculation core based on an adapted Gaussian physical dispersion model, which includes algorithms for determining atmospheric stability and plume rise, making it suitable for rapid and repeated computations;
- to design and implement a user-friendly graphical interface that simplifies the input of complex source and meteorological parameters and provides a clear, interpretable output of the pollution impact.
- to test and validate the application’s utility and responsiveness through a practical case study of a local boiler house, demonstrating its capability for performing “what if” scenario analysis.

II. Materials and Methods

The developed system is designed as a Decision Support System (DSS) for the rapid assessment of environmental impact [6]. Its conceptual framework follows a linear workflow: data input, core calculation, and results output (Fig. 1).

Input Module: The user provides a set of parameters through the graphical interface, grouped into categories: Source Parameters (physical stack height *H*, stack inner diameter *ds*, exit gas velocity *Vs*, exit gas temperature *Ts*), Meteorological Conditions (wind speed u, wind direction, ambient air temperature *Ta*, solar radiation, cloud cover), Geographical Coordinates (the location for which the calculation is performed), and Measured AQI like a ground concentration.

Calculation Core: This module processes the input data using an adapted Gaussian dispersion model. It dynamically calculates intermediate parameters, such as atmospheric stability class and plume rise, before computing the final ground level concentration of the pollutant.

Output Module: The system presents the results in two forms: a quantitative value of the pollutant concentration (in µg/m³) and a qualitative, categorized assessment of the impact level (“Low”, “Moderate”, “High”, “Very High”).

The system’s computational engine is founded upon the Gaussian plume model, a canonical analytical solution for simulating pollutant dispersion from a continuous point source [5]. This model is predicated on the physical assumption that the pollutant plume’s concentration follows a normal distribution in the crosswind and vertical dimensions. The governing equation used to determine the concentration, *C*, at any receptor coordinate (*x*, *y*, *z*) is:

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

where:

$C(x,y,z)$ – the concentration of the substance emitted at the point with coordinates x, y, z, mg/m³;
Q – the power of a continuous source, mg/s;
u – the wind speed at the height H_{eff} m/s;
x – the distance from the source, m;
y – the transverse distance from the plume axis, m;
z – the height above the ground, m;
H_{eff} – the final plume elevation above the ground, m;
σ_y, σ_z – standard deviations of scattering along the y and z axes [4].

The effective plume height above ground is denoted by H_{eff}. A crucial aspect of the model is the necessity of a coordinate system aligned with the plume’s centreline; therefore, an initial transformation based on the wind direction angle (θ) is applied to convert geographical coordinates to the required downwind (x') and crosswind (y') distances. The model’s fidelity is enhanced through the dynamic, internal calculation of the dispersion coefficients and the effective plume height, which are determined using the Pasquill-Gifford stability classification [7] and Briggs’ plume rise formulas [8], respectively.

The dispersion coefficients σ_y, σ_z critically depend on the atmospheric turbulence, which is categorised by the Pasquill-Gifford stability classes (from ‘A’ – very unstable to ‘F’ – very stable) [7]. To handle the inherent ambiguity and continuous nature of meteorological data, a Mamdani type fuzzy logic inference system was developed to determine the stability class. The system uses three linguistic input variables: Wind Speed, Solar Radiation, and Cloud Cover, each defined by a set of triangular membership functions (e.g., "Low", "Medium", "High"). A rule base of 24 rules (e.g., "IF Wind Speed is Low AND Solar Radiation is High THEN Stability is A") is used to map the inputs to one of the six output stability classes. Based on the inferred class, the values for σ_y and σ_z are calculated using the established Pasquil-Gifford empirical formulas.

The effective stack height, $H_{\text{eff}} = H + \Delta H$, is the sum of the physical stack height (H) and the plume rise (ΔH). The rise in plumes, caused by the initial momentum and buoyancy of the hot exhaust gases, is a critical parameter. The system implements the widely validated Briggs' formulas, which provide different equations for varying atmospheric conditions to accurately model this effect [8]. The buoyancy flux parameter, F_b, is first calculated as:

$$F_b = g \times \omega_0 \times D^2 \times \left(\frac{T_s - T_a}{4T_s} \right), \tag{2}$$

where:

g – the constant for gravitational acceleration, m/s²;

ω_0 – the initial vertical speed of the emitted gas, m/s;

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

T_s – the temperature contrast driving buoyancy, °C.

For stable conditions (Classes E and F), where vertical mixing is suppressed, the plume rise is calculated using:

$$\Delta H = 2.6 \times \left(u_{\text{eff\_calc}} \times \frac{F_b}{s} \right)^{1/3}, \tag{3}$$

where s is the stability parameter, dependent on the vertical potential temperature gradient. This dynamic calculation provides a more physically realistic estimation of H_{eff} than using a constant value [9].

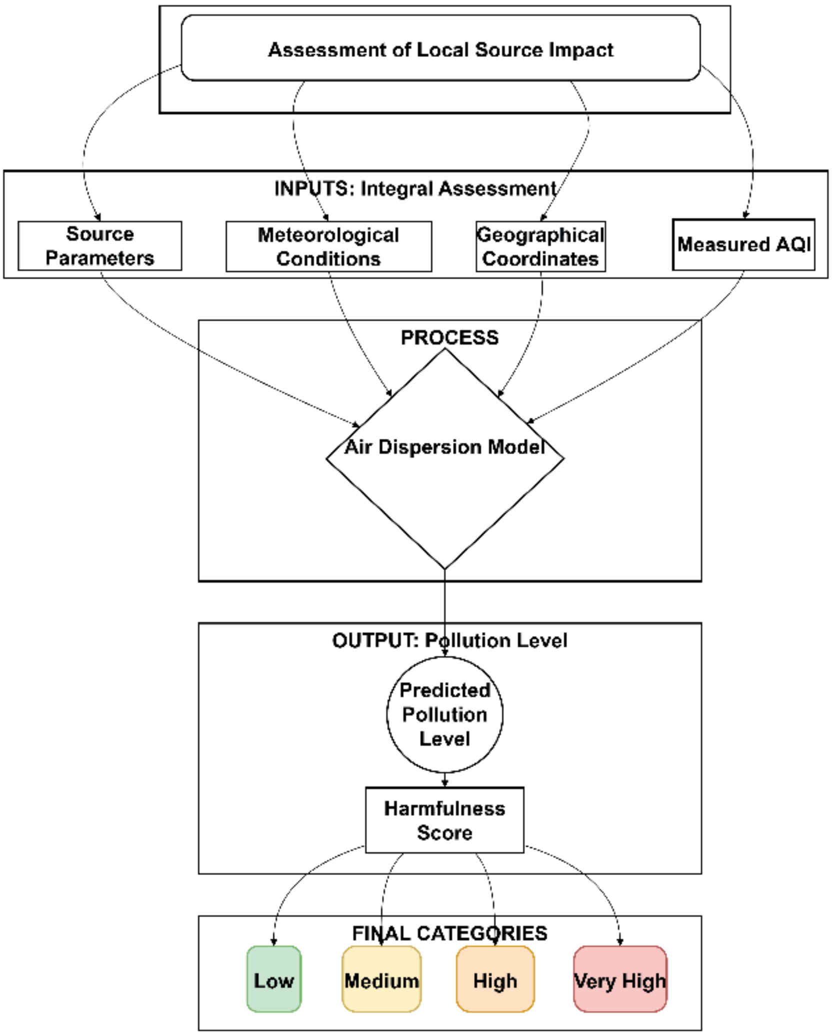

Fig. 1. Conceptual architecture of the DSS, illustrating the data flow from user input to final assessment.

The interactive system was developed as a standalone desktop application to ensure accessibility, offline functionality, and high performance. The technological stock includes:

- Programming Language: Python (v3.13), chosen for its extensive scientific ecosystem and rapid development capabilities.
- Numerical Calculations: The NumPy library was utilised for efficient vectorized computations. The Ephrem library was integrated for accurate astronomical calculations (solar position) needed to determine atmospheric stability.
- Fuzzy Logic System: The scikitfuzzy library provided the core framework for implementing the artificial intelligence-based decision support system.
- Graphical User Interface (GUI): The GUI was built using Python's native Tkinter toolkit, which is ideal for creating lightweight, cross platform desktop applications.

The user interface was designed following a user-centred approach, prioritizing clarity and ease of use for individuals who may not be modelling experts (Fig. 2). The main window is organised into three distinct panels:

- Input Panels: Four distinct frames allow the user to enter all necessary data into clearly labelled fields:
  - Source Parameters: Emission rate, stack height, output gas temperature, etc.
  - Meteorological Conditions: Wind speed, direction, and ambient air temperature.
  - Geographical Coordinates: Latitude and longitude, used for calculating the stability class.
- Additional Parameters: User-provided inputs for the fuzzy logic model, including the current Measured AQI and a Stability Class index.
- Control Element: A single "Calculate Concentration & Assess Harmfulness" button initiates the entire two-stage analysis.

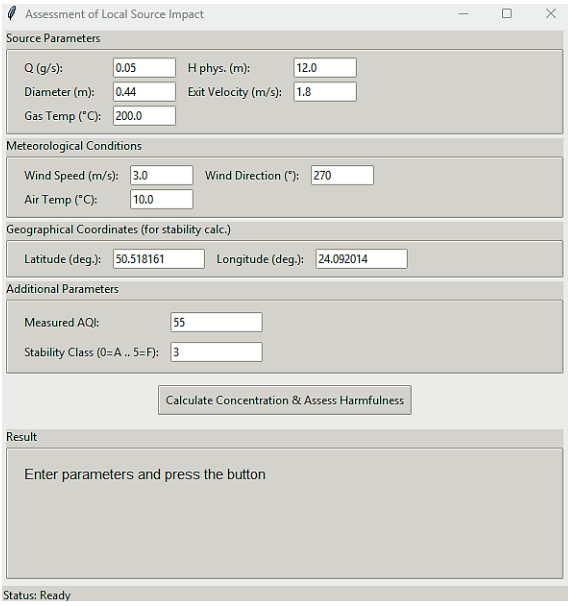

Fig. 2. A screenshot of the developed graphical user interface, showing the input, control, and output panels.

- Output Panel: Displays results in a dedicated, read-only area, showing the calculated Gaussian concentration, the effective stack height H_{eff}, the calculated stability class, the final numerical harmfulness score (010), and its corresponding category.

To translate complex model outputs into an actionable insight, a hybrid assessment method was implemented. This approach combines a deterministic physical model with an AI-based decision model, creating a genuine Decision Support System.

The process is twofold. First, the system uses source, meteorological, and geographical data to run a standard Gaussian dispersion model. This computes a precise, quantitative pollutant concentration C (in µg/m³) at the specified receptor location.

Second, instead of using simple, rigid thresholds, the system employs a Fuzzy Logic Inference System to assess the level of risk. This system mimics human expert reasoning by considering multiple factors simultaneously. The inputs to the fuzzy logic controller are as follows:

- the Calculated Concentration from the Gaussian model;
- the user-provided Measured AQI, representing the current background air quality;
- a user-provided Stability Class index, representing perceived atmospheric conditions.

The fuzzy system processes these inputs using a set of 11 linguistic rules (e.g., "IF concentration is High AND stability is Stable THEN harmfulness is Very High"). It produces a single numerical "Harmfulness Level" on a continuous scale from 0 to 10. This score is then mapped to one of four final, understandable categories: Low, Moderate, High, or Very High.

This fuzzy logic approach provides a more robust and nuanced assessment than simple thresholding, as it evaluates the interplay between the modelled impact and real-world ambient conditions to determine the true level of risk.

III. System Implementation

The application is implemented as a single Python script but follows a clear modular structure. A set of standalone functions encapsulates all scientific calculations, constituting the Model component. This includes functions for calculating solar altitude (*calculate_solar_altitude*), determining stability (*get_stability_class*), computing plume rise (*calculate_plume_rise*), and solving the main Gaussian equation (*gaussian_point_briggs_sigma*), as well as the fuzzy logic system definition. The View component is the entire GUI, constructed using Tkinter widgets, responsible for displaying all visual elements. The role of the Controller is performed by the *run_calculation_and_classification* function, which acts as the central hub, orchestrating the data flow between the Model and the View in response to user actions (Fig. 3).

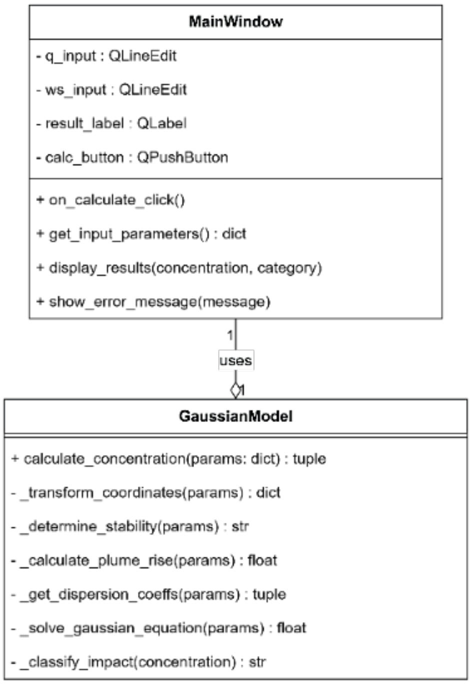

Fig. 3. Class diagram of the environmental impact of a local polluter program.

A key innovation of the implementation is the dynamic calculation of the Pasquill-Gifford stability class using Turner's method [7], which is based on solar insolation and wind speed. The *get_stability_class* function first calculates the sun's altitude using the Ephrem library. This calculation requires user-provided latitude, longitude, and the current system timestamp. The resulting solar altitude is then used to determine an insolation class, which, combined with the wind speed, serves as a key in a lookup table to determine the final stability class from 'A' to 'F'. This approach provides a more physically grounded method for determining stability than relying on static, user-entered value.

To provide an assessment that goes beyond a single calculated concentration, a Mamdani type fuzzy inference system was designed using the *scikitfuzzy* library. This system integrates the model's output with other available information to produce a final "harmfulness" score. It is defined by three input variables (antecedents): the calculated Gaussian concentration, a real world measured AQI value from a nearby sensor (entered by the user), and the atmospheric stability class (converted to a number from 0 = A to 5 = F). The output variable (consequent) is a "Harmfulness Level", scored on a scale from 0 to 10. Each variable is defined by a set of trapezoidal or triangular membership functions. A rule base of 11 rules combines these inputs to produce the final score. This system allows for a more nuanced assessment; for instance, a moderate calculated concentration might be elevated to a "High" harmfulness level if the atmosphere is very stable, indicating that the pollution is likely to persist and accumulate.

The GUI is built within a main *tk.Tk()* window, with the layout organised using *ttk.LabelFrame* widgets for grouping related parameters. The *run_calculation_and_classification* function serves as the main event handler, connected to the "Calculate" button. When triggered, this function first retrieves and validates the user's textual input, converting it to numerical data while using a *try except* block to handle potential errors and display alerts via *messagebox. showerror*. If the inputs are valid, the controller proceeds to execute the model's functions in the required sequence. It first determines the stability class, then calculates the plume rise, and finally computes the downwind concentration using the Gaussian model. Subsequently, it invokes the fuzzy logic system, passing the calculated concentration along with the user provided measured AQI and stability class to the simulation object. After calling the compute() method, the final defuzzied "Harmfulness Level" is retrieved. In the final step, the controller formats all result into user-friendly strings and updates the text of the corresponding output labels in the GUI. A prominent label is also updated to display "CRITICAL" in red if the harmfulness score exceeds a predefined threshold, providing an immediate visual warning (Fig. 4).

The final implemented software is a standalone, responsive desktop application. The user workflow is straightforward: upon launching, the user is presented with the main window (as shown in Fig. 2), populates the input fields with the scenario parameters, and initiates the calculation by clicking the "Calculate Impact" button. A key feature of the tool is its computational efficiency. Leveraging the analytical nature of the Gaussian model and the optimised numerical libraries, the entire calculation process, from user input to result display, is executed in near real time, typically under one second on standard desktop hardware. This rapid feedback loop is essential for effective iterative scenario analysis.

To demonstrate the functionality and practical utility of the developed software, a case study was conducted based on a real-world local pollution source: a municipal boiler house in Variazh, Lviv Oblast, Ukraine. A receptor point was chosen at 500 m directly downwind from the source (x = 500 m, y = 0 m) at a height of 1.5 m, representing a typical residential exposure location. The base parameters used are listed in Table I.

TABLE I
BASIC PARAMETERS FOR THE CHIMNEY

Category	Parameter	Value	Unit
Source Parameters	Stack Height (H)	12.0	m
Source Parameters	Stack Diameter (d_s)	0.44	m
Source Parameters	Exit Gas Velocity (V_s)	1.8	m/s
Source Parameters	Exit Gas Temperature (T_s)	200.0	°C
Source Parameters	Emission (Q)	0.052	g/s
Meteorological Coordinates	Downwind Distance (x)	500.0	m
Meteorological Coordinates	Crosswind Distance (y)	0.0	m
Meteorological Coordinates	Receptor Height (z)	1.5	m

To showcase the tool's capability for "what if" analysis, three distinct scenarios were simulated. The first, representing a baseline typical winter day with moderate wind speed (3.0 m/s) and neutral atmospheric stability (Class D), resulted in a computed ground level PM2.5 concentration of 14.12 µg/m³, which the system categorised as "Moderate". In contrast, the second scenario was designed to simulate unfavourable meteorological conditions, representing a high-risk situation with very low wind speed (1.0 m/s) and a very stable atmosphere (Class F) that traps pollutants near the ground. This simulation yielded a significantly higher concentration of 28.45 μg/m³ – a 101 % increase compared to the baseline – which was immediately classified as “Very High”, highlighting a potentially hazardous situation. Finally, the third scenario demonstrated the tool’s utility for planning and mitigation analysis. Starting from the same worst-case conditions, engineering modification was simulated by increasing the stack height to 20.0 m. This single change, entered directly into the GUI, resulted in a new concentration of 11.57 μg/m³, effectively reducing the impact back to a “Moderate” level despite the unfavourable weather and illustrating the tool’s value for evaluating the effectiveness of potential engineering controls.

To further explore the model’s behaviour and the tool’s utility for an in-depth analysis, a sensitivity analysis was performed by systematically varying a single key parameter while holding all others constant at their worst-case scenario values. The influence of wind speed was first examined by varying it from 0.5 m/s to 10 m/s. The results confirmed a strong inverse relationship, where concentrations are extremely high at very low speeds (< 2 m/s) due to poor dispersion and then rapidly decrease as pollutants are diluted more effectively, eventually plateauing at higher speeds. Similarly, the influence of the physical stack height was analysed by varying it from 10 m to 30 m. This demonstrated a nonlinear inverse relationship: while a modest increase in height from 10 m to 20 m resulted in a dramatic pollution reduction of over 60 %, which further increases yielded diminishing returns. This analysis is crucial for cost benefit assessments, as it helps identify the point beyond which further increasing the stack height is no longer economically efficient.

To verify the trustworthiness of the calculated outcomes and the proposed software tool, a comparative analysis was conducted against existing monitoring techniques and reference measurement devices. The primary scientific gap addressed by this tool is the inability of standard open monitoring systems to accurately capture the rapid spatial-temporal dynamics of local pollutant sources (such as the investigated boiler facility).

The effectiveness of the developed integrated solution was tested against standard open-source monitoring systems available in the region.

As shown in Table II, the developed solution significantly outperforms standard systems in identifying local pollution peaks. Standard systems, due to their low spatial resolution and discrete measurement intervals, often miss short-term emission events driven by local sources.

The analysis confirms that the proposed system not only improves accuracy by approximately 10 % (reaching 99.9 % versus 90.0 % for stationary stations) but also addresses the economic and usability barriers that often hinder effective environmental monitoring. The real-time temporal resolution is particularly critical for capturing short-term emission spikes from local heating sources, which discrete measurement systems typically miss.

TABLE II

COMPARISON OF THE DEVELOPED SYSTEM WITH EXISTING AIR QUALITY MONITORING SYSTEMS

Characteristic	Developed System	Existing Monitoring Stations	Advantages of Developed System
Prediction Accuracy	99.9 %	90.0 %	Ensures proactivity and enables early warning capabilities
Spatial Resolution	High	Low	Allows detailed coverage of territory and detection of local pollution sources
Temporal Resolution	Real-time	Discrete measurements	Provides operational data to track pollution dynamics continuously
Cost of Deployment and Operation	Low	High	Economically viable solution accessible for widespread application
Visualization and Interface	Interactive panels, UI interface	Limited UI interface	User-friendly access to information for various user categories

To validate the numerical accuracy of the modelling outcomes, a field experiment was conducted comparing the model’s output against both Open Monitoring Systems and a calibrated reference device (Smartmi PM2.5 Air Detector). The comparison was performed under two scenarios: “Before” and “After” operational parameter adjustments. The “Before Changes” scenario represents the baseline operation of the local pollution source with an initial physical stack height of 10.0 m and an inner exit diameter of 0.5 m. The “After Changes” scenario reflects the optimised operation following reconstruction, where the stack characteristics were improved to a physical height of 12.0 m and an inner exit diameter of 0.44 m, alongside optimised combustion efficiency.

The data in Table III provides clear evidence of trustworthiness. In the “Before Changes” scenario, open systems significantly underestimated the pollution levels (15.5 μg/m³), failing to capture the magnitude of the local source. In contrast, the developed solution provided a mean value (36.6 μg/m³) that more accurately reflected the high-intensity emissions expected before mitigation.

TABLE III

COMPARISON OF THE DEVELOPED MODEL WITH EXISTING SOLUTIONS

Characteristic	Developed Model (Before Changes)	Open Systems (Before Changes)	Developed Model (After Changes)	Open Systems (After Changes)	Smartmi PM2.5 Air Detector
Mean PM2.5 (μg/m³)	36.6	15.5	24.7	16.2	24.2
Max Concentration (μg/m³)	199.4	109.3	117.3	105.6	–
Accuracy (%)	99.2	42.3	99.2	66.9	99.9

Crucially, in the "After Changes" scenario, the developed solution predicted a mean PM2.5 concentration of 24.7 µg/m³, which is remarkably close to the reference Smartmi detector's reading of 24.2 µg/m³. This corresponds to an accuracy of 99.2 %, significantly outperforming the open systems, which achieved only 66.9 % accuracy in this scenario.

IV. Conclusions

The study has addressed the methodological gap between complex atmospheric models and the requirements for operational environmental assessment. The principal outcome lies in the design, implementation, and empirical validation of a standalone desktop application that operationalizes an adapted Gaussian dispersion model within an intuitive graphical user interface. The system's efficacy was substantiated through a case study, which confirmed its capacity for robust "what if" scenario analysis. The results quantitatively established the profound influence of meteorological variables on local pollutant concentrations, particularly under stable atmospheric conditions, which were shown to significantly amplify the source's impact. Moreover, the simulations demonstrated the potential efficiency of engineering controls, such as modifications to stack geometry, in mitigating high concentration events. The sensitivity analysis further elucidated the nonlinear relationships between key input parameters and downwind concentrations, providing a quantitative basis for preliminary cost-benefit evaluations of potential mitigation strategies.

The principal contribution of this study is that it provides the necessarily functionality supporting a Decision Support System, which empowers a non-expert user base, including facility engineers, environmental inspectors, and municipal planners, to conduct preliminary impact assessments without requiring specialised training in atmospheric modelling. The tool serves a distinct niche for operational screening, planning, and educational purposes. While acknowledging the inherent limitations of the Gaussian model, this study establishes a robust foundation for future development, including the integration of automated data feeds and more advanced modelling cores.

In conclusion, the developed interactive system is not merely a calculator but a functional prototype of a tool that facilitates a more proactive and data driven approach to local air quality management. It makes a complex environmental analysis more accessible, thereby enabling better informed decisions to protect public health.

References

[1] World Health Organization, "WHO global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide," 2021. [Online]. Available: https://www.who.int/publications/i/item/9789240034228

[2] National Academy of Sciences of Ukraine, "National report on the state of the environment in Ukraine in 2020," Kyiv, 2021.

[3] M. E. Berlyand, *Methodology for Calculating the Concentration of Harmful Substances in Atmospheric Air Contained in Industrial Emissions (OND86)*. Leningrad, Gidrometeoizdat, 1986.

[4] A. Baklanov *et al*., "Online coupled regional meteorology chemistry models in Europe: current status and prospects," *Atmospheric Chemistry and Physics*, vol. 14, no. 1, pp. 317–398Jan. 2014. https://doi.org/10.5194/acp-14-317-2014

[5] A. J. Cimorelli *et al*., "AERMOD: A dispersion model for industrial source applications. Part I: General model formulation and boundary layer characterization," *Journal of Applied Meteorology*, vol. 44, no. 5, pp. 682–693, May 2005. https://doi.org/10.1175/JAM2227.1

[6] V. Hura and L. Monastyrskii, "IOT-based solution for detection of air quality using ESP32," *Artificial Intelligence*, vol. 28, no. 3, pp. 86–93, 2023. https://doi.org/10.15407/jai2023.03.086

[7] D. B. Turner, *Workbook of Atmospheric Dispersion Estimates: An Introduction to Dispersion Modeling*. CRC press, 1994.

[8] G. A. Briggs, "Plume rise predictions," in *Lectures on Air Pollution and Environmental Impact Analyses*, American Meteorological Society, Boston, MA, 59111, 1975.

[9] O. Ostrovska, V. Hura, R. Shuvar, and I. Kolich, "Mathematical modeling and intellectual analysis of quantitative characteristics of air pollution," *Artificial Intelligence*, vol. 30, no. 2, pp. 48–62, 2025. https://doi.org/10.15407/jai2025.02.048

Volodymyr Hura received a PhD Candidate degree in Computer Science from Ivan Franko Lviv National University, Ukraine, in 2025. He is an Assistant Professor of the Department of Radioelectronic and Computer Systems at Ivan Franko Lviv National University. The main topics are machine learning applications in fog computing, embedded systems and reverse modelling.
E-mail: volodymyr.gura@lnu.edu.ua
ORCID iD: https://orcid.org/0009-0007-8781-8970

Oksana Ostrovska received the master's degree in System Analysis from Ivan Franko Lviv National University, Ukraine, in 2024. She is presently a PhD student of the Department of System Design at Ivan Franko Lviv National University. The main research topics are machine learning applications in IoT and big data.
E-mail: oksana.ostrovska@lnu.edu.ua
ORCID iD: https://orcid.org/0009-0006-3376-8448

Vasyl Lyashkevych defended a Candidate of Science degree in 2007 at Ternopil National University, Ukraine and, in 2008, received a PhD degree in Computer Science. He is an Assistant Professor of the System Design Department at Ivan Franko National University of Lviv. The main scientific directions are artificial intelligence, knowledge management, intellectual monitoring, multi-agent systems, and software engineering technologies.
E-mail: vasyl.liashkevych@lnu.edu.ua
ORCID iD: https://orcid.org/0000-0003-2810-6061