Intelligent system for analysis and assessment of population health harm risks by air pollution level

Nickolay Rudnichenko[1], Vladimir Vychuzhanin[1], Denys Shvedov[1], Natalia Shibaeva[1] and Igor Petrov[2]

[1]Odesa Polytechnic National University, 1 Shevchenko Ave., Odesa, 65001, Ukraine
[2]National University "Odesa Maritime Academy", 8 Didrichson Str., Odesa, 65029, Ukraine

Abstract

Air pollution by fine particulate matter ($PM_{2.5}$, PM_{10}), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), carbon monoxide (CO) and ozone (O_3) is among the leading environmental threats to public health in densely populated urban areas, yet the tools available to health authorities rarely combine heterogeneous monitoring data with interpretable risk estimates. This paper presents an intelligent system that assigns an observation to one of six air quality index (AQI) classes, used as a proxy for the level of harm to population health, together with a comparative evaluation of the six models behind it: a decision tree, a support vector machine, a random forest, XGBoost, a convolutional neural network (CNN) and a long short-term memory (LSTM) network. All six are trained on identically preprocessed data – the Air Pollution Image Dataset from India and Nepal, cross-checked with structured measurements from two open sources – and are compared at the evaluation stage rather than chained into a sequence. The system is implemented as a client-server application with a Python and Flask backend and a React frontend, whose dashboard provides exploratory data analysis, model inference and model comparison. The random forest and the CNN are the most accurate models (accuracy 0.988 and 0.987, macro-averaged F1-score 0.987 and 0.986), followed by the LSTM network (0.980 and 0.979), whereas the single decision tree falls to 0.856 because it does not separate the two most severe classes. Average training time ranges from 3 s for the decision tree to 67 s for the LSTM network, which makes the random forest the best compromise between accuracy and computational cost.

Keywords

air pollution, health risk assessment, machine learning, deep learning, intelligent system

1. Introduction

In recent decades, rapid industrialization, urban growth, and the intensification of anthropogenic activity have led to significant degradation of atmospheric air quality, particularly in densely populated areas. Prolonged exposure to fine particulate matter ($PM_{2.5}$ and PM_{10}), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), ozone (O_3), and carbon monoxide (CO) is associated with a range of acute and chronic health effects, including respiratory and cardiovascular diseases, adverse birth outcomes, neurodevelopmental disorders, and increased mortality rates [1]. Public health authorities, environmental agencies, and urban planners are therefore facing growing pressure to adopt evidence-based and technologically robust strategies for mitigating the population health risks associated with air pollution.

Despite the growing body of epidemiological evidence, the implementation of intelligent systems capable of assessing health risks at the population level remains underdeveloped. A gap persists between theoretical models of health–environment interaction and practical applications able to process heterogeneous data, account for spatial and temporal variation, and provide community-specific risk assessments [2]. The main obstacle is the data itself. Standardized, high-quality, and openly available datasets that simultaneously describe environmental conditions and population health parameters are

CS&SE@SW 2025: 8th Workshop for Young Scientists in Computer Science & Software Engineering, November 21, 2025, Kryvyi Rih, Ukraine
https://doi.org/10.55056/cssesw2025/paper17
nickolay.rud@gmail.com (N. Rudnichenko); vint532@gmail.com (V. Vychuzhanin); studylearner@gmail.com (D. Shvedov); nati.sh@gmail.com (N. Shibaeva); firmn@gmail.com (I. Petrov)
0000-0002-7343-8076 (N. Rudnichenko); 0000-0002-6302-1832 (V. Vychuzhanin); 0009-0002-4823-8782 (D. Shvedov); 0000-0002-7869-9953 (N. Shibaeva); 0000-0002-8740-6198 (I. Petrov)

scarce: monitoring networks and health registries are maintained independently and differ in structure, resolution, accessibility, and privacy constraints [3]. Air quality data come from satellite imagery, ground-based monitoring stations, or mobile sensors with different spatial and temporal resolutions, and health data come from hospitals, clinics, public health surveys, or wearable devices in incompatible formats; unifying such streams into analyzable form requires substantial cleaning, feature engineering, normalization, and imputation, while the use of individual-level health records is further restricted by ethical and legal considerations [4]. Missing values, inconsistent measurements, temporal misalignment, and limited geographic coverage additionally compromise the ability to construct models that generalize across regions and populations.

As data availability increases and modeling techniques evolve, intelligent systems may nevertheless enable early warning mechanisms, personalized exposure tracking, and adaptive strategies for urban planning, health care delivery, and environmental regulation, becoming components of smart city infrastructures [5]. The aim of the present work is an intelligent system that estimates the level of harm to population health from the measured air pollution level, together with a reproducible comparison of the models behind it. The contribution is threefold: a unified pipeline in which six machine learning and deep learning models are trained on identically preprocessed air quality data and compared using one set of metrics; a client-server implementation whose dashboard exposes exploratory data analysis, prediction, and model comparison to a user without programming skills; and an assessment of the accuracy versus training cost trade-off across the six models. The remainder of the paper analyzes related work, describes the system and the formal statement of the problem, and reports the experimental results.

2. Analysis of existing research and approaches

Machine learning (ML) [6, 7, 8] and deep learning (DL) [9, 10, 11] offer a practical way of addressing the complexities inherent in modeling the health impact of air pollution. Unlike traditional statistical methods, which require strong a priori assumptions and admit only limited variable interactions, these models learn nonlinear relationships between input variables and outcomes, incorporate several data modalities, and scale to high-dimensional data [12]. Convolutional and recurrent architectures, attention mechanisms, and ensemble learning frameworks [13, 14] have been applied to predicting pollutant concentrations, estimating exposure risks, and modeling disease incidence as a function of environmental variables; combined with geospatial, temporal, and sociodemographic layers, they yield high-resolution views of population vulnerability, and explainable AI methods are increasingly added to make their output usable in public health decisions. The limitations are equally clear: many applications [15] remain experimental and rest on limited or synthetic datasets, and real-world deployment requires thorough validation, domain-specific customization, alignment with local policy frameworks and health information systems, and interdisciplinary work with epidemiologists and environmental engineers, as well as scrutiny of interpretability and fairness before the results influence resource allocation or risk communication.

Lipatova and Potapchenko [16] classify air pollution levels with ensemble ML models and neural network architectures, reporting accuracies above 90% for air quality index (AQI) levels with XGBoost and long short-term memory (LSTM) models, with particular attention to model interpretability and to the comparative evaluation of algorithms. The work is nevertheless limited to air quality prediction and does not incorporate any direct assessment of health outcomes.

Ravindra et al. [17] follow a more applied route, linking $PM_{2.5}$ and PM_{10} concentrations to the number of outpatient visits for respiratory diseases. Using a multilayer perceptron together with seasonal and demographic stratification of real medical records, the authors build reliable predictive models for specific age groups, which is a crucial step toward personalized risk assessment, although the shallow architecture and the limited spatial coverage constrain the result.

Wang et al. [18] combine spatial convolutional networks, temporal recurrent modules, and attention mechanisms into a hybrid DL model for forecasting pollutant concentrations. Its ability to handle

incomplete data and to capture spatiotemporal dependencies is the key contribution, but the scope again ends at pollutant prediction, with no direct link to health outcomes.

Lee et al. [19] propose a conditional U-Net network as a surrogate model for the rapid assessment of the health impact of short-term $PM_{2.5}$ exposure, replacing resource-intensive atmospheric simulations. Speed and scalability make the approach attractive for real-time deployment, yet it emulates a physical model instead of forming a health risk evaluation pipeline and is not integrated with epidemiological or clinical data.

A GIS-based line of work [20] maps air pollution and estimates health risk from satellite-derived $PM_{2.5}$ data, applying spatial interpolation and exposure–response functions to obtain fine-grained exposure maps for urban areas with sparse ground sensors. Its spatial resolution is high, but temporal dynamics are not modeled and no predictive ML component is involved; other studies show that the spatial and the temporal dimension of exposure can be treated jointly with DL methods [21, 22].

Published systems thus either forecast pollution levels or estimate exposure, and they rarely make the model selection step reproducible for the practitioner who has to operate them. The present work therefore keeps the modeling target deliberately simple and verifiable: the AQI class of an observation, treated as a proxy for the level of harm to population health, is predicted from the measured pollutant concentrations and from the location and time of the measurement. Six models covering interpretable, ensemble, and deep architectures are trained on the same feature set and compared on accuracy, macro-averaged precision, recall, F1-score, and training time, and the whole procedure is embedded in a deployable client-server system. The scientific novelty lies in this end-to-end combination of a reproducible model comparison with a deployable analytical service; the practical significance stems from the possibility of integrating such a service into public health monitoring and urban environmental management.

3. System concept, implementation and technical aspects

The proposed framework integrates several ML and DL models, each covering a different aspect of the risk evaluation task [23]. The models are trained independently on the same preprocessed dataset and produce separate prediction outputs, which are compared at the evaluation stage rather than used sequentially. Because all models see the same sample, a single set of pre-validation and cleaning parameters – anomaly removal, verification of missing values, correlation screening, and cross-validation – limits the influence of residual noise and prevents errors from propagating between models, while keeping the contribution of every model separately interpretable.

The decision tree (DT) provides preliminary filtering and categorization of the data. Its hierarchical logic is directly readable and exposes the threshold values of the pollutant indicators that separate risk levels, which also makes it a baseline for classifying regions or time intervals by pollution level.

The support vector machine (SVM) addresses nonlinear relationships between air quality indicators and the assigned risk level. It performs well on heterogeneous features and constructs accurate separating surfaces in a multidimensional feature space, which delineates the conditions under which pollutant exposure becomes significantly detrimental to health.

The random forest (RF) aggregates multiple decision trees built on repeated samples. This refines the structure of the risk factors, reduces the effect of outliers and noise, and improves the reliability and reproducibility of the risk estimates.

The gradient boosting model XGBoost improves the predictive capacity of the system through iterative error correction and handles complex dependencies between the temporal, spatial, and pollutant-related features efficiently.

The convolutional neural network (CNN) extracts local patterns from the ordered feature representation. This is useful when observations are correlated in space and time and helps to identify clusters of elevated health risk, which is relevant for targeted public health interventions.

The LSTM network captures the temporal dynamics of pollution. Owing to its ability to learn long-term dependencies, it can represent delayed effects, which is a prerequisite for forecasting how risk

evolves rather than only classifying the current state.

The primary dataset chosen for this research is the Air Pollution Image Dataset from India and Nepal [24]. It comprises annotated images of urban and rural environments captured at varying levels of air pollution, together with the pollutant concentrations, the location, the time, and the AQI class recorded for every image; the structured attributes of this collection form the feature space used in the present study. The six AQI labels – Good, Moderate, Unhealthy for Sensitive Groups, Unhealthy, Very Unhealthy, and Severe – constitute the target variable and are used as a proxy for the level of harm to population health. To improve the reliability and generalizability of the model outputs, the data were extended and validated with structured environmental measurements from two authoritative sources [25, 26], which allowed the pollution levels to be cross-validated and the prediction thresholds to be calibrated.

The problem is therefore a multiclass classification task, in which the algorithm must assign an observed instance to one of several predefined classes. The input dataset is represented as $X = \{x_1, x_2, \ldots, x_n\}$, where each object x_i is a vector of features from the space $\mathbb{R}^d$. For each object x_i a class label is matched $y_i \in \{1, 2, \ldots, K\}$, where K is the number of classes (six in the present task). It is required to build a function $f : \mathbb{R}^d \to \{1, 2, \ldots, K\}$, which for any input object x will predict the class label y (public health risk). The model is built using a training sample $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ formed from informative input features, and its task is to find an approximation of the function f based on this data. If $P(y = k|x)$ denotes the probability that an object belongs to class k, then $f(x)$ predicts the class with the highest posterior probability

$$f(x) = \underset{k \in \{1,2,\ldots,K\}}{\arg\max} \; P(y = k|x). \tag{1}$$

To train the model, a loss function is employed to quantify the discrepancy between the predicted class labels and the true class labels. One of the most commonly used loss functions for this purpose is the cross-entropy loss:

$$L(y, y') = -\sum_{k=1}^{K} y_k \log(y'_k) \tag{2}$$

where y_k is a binary indicator (0 or 1) showing whether an object belongs to class k, and $y'_k = P(y = k|x)$ is the probability of that class as predicted by the model. The model is optimized by minimizing the loss function L using optimization techniques such as gradient descent, and the final prediction is made by selecting the class with the highest probability.

The operation of the system is formalized as a BPMN diagram with four lanes – user, web frontend, server backend, and ML models – presented in figure 1. A session starts when the user loads a dataset through the frontend; the backend saves it, runs the exploratory data analysis (EDA), and prepares the data for prediction; the models produce their outputs; and the saved results are returned to the frontend for visualization, which closes the session.

Within the study, the imported data are stored as Pandas DataFrame objects and pass through preprocessing, which includes the detection of anomalies and outliers, the treatment of missing values, and the mitigation of class imbalance in the target variable by oversampling of the minority classes. Preprocessing is followed by EDA: descriptive statistics, correlation analysis, factor analysis, and, in cases where the number of input features is high, dimensionality reduction. During the modeling phase, the individual models are trained on the training subset obtained by data splitting and are checked by cross-validation. They are then evaluated on unseen test data in terms of accuracy, precision, recall, F1-score, and training time, and serialized into separate files for subsequent loading and deployment on new datasets. The implementation uses NumPy and Pandas for data structure manipulation, Matplotlib and Seaborn for visualization, scikit-learn (sklearn) for label encoding, normalization, metric evaluation, and the classical models, and Keras for the neural networks.

The main classes of the system are shown in figure 2. The ExploratoryDataAnalysis class is responsible for the exploratory analysis: it stores the paths to the directories with graphical output and generates the visualizations through the methods generateDataDistributionPlot, generateEmissionIndexPlot,

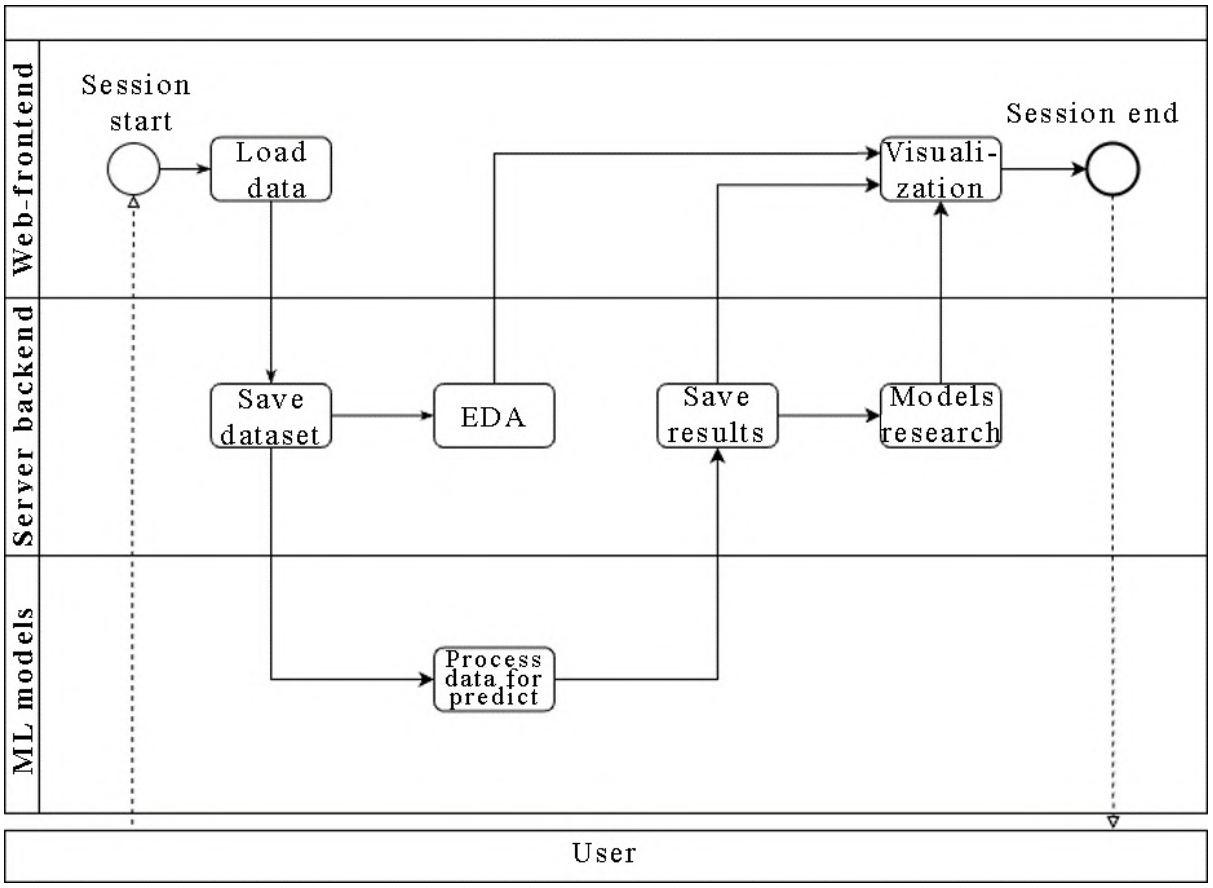


Figure 1: BPMN formalization scheme.

generateCorrelationMatrixPlot, generateZScorePlot, generatePairplotPlot, and generateClassDistributionPlot. The central file server.py contains the core operational logic and defines the main API endpoints get_session (session initialization), upload_file (data upload), start_eda (EDA execution), predict (model inference), and compare (model performance comparison).

The Regression and Classification classes execute the corresponding families of models. Both keep the prediction results and refer to the serialized models stored in the file system of the server, among them decisionTreeModel.pkl, randomForestModel.pkl, XGBoostModel.pkl, mlpModel.keras, lstmModel.keras, and dnnModel.keras; the Classification class additionally manages the assignment of environmental data to the predefined classes and the extension of the output dataframe (classificationResult, extendDataframe). The CompareClassification and CompareRegression classes are responsible for the comparative analysis of the outputs of the two families and provide the plotting operations generateForecastComparePlot, generateHeatmapPlot, generateAQIDistributionPlot, generatePopularityPlot, generateBoxPlot, generateModelPredValuesPlot, generateCorrelationMatrixPlot, and generateAQIForecastPlot, which facilitate the interpretation of model performance and cross-model benchmarking.

The intelligent system is built on a client-server architecture. The backend is implemented in Python using the Flask microframework, which offers a lightweight and flexible basis for the RESTful API and handles the HTTP routing that manages the acquisition, processing, and storage of the environmental data; the structural organization of the backend project is presented in figure 3. The frontend is implemented in JavaScript using the React library and provides a file upload page, modules for displaying the EDA results, a prediction page for executing model forecasts, and a dedicated section for model comparison. Interaction with the backend is handled via the Axios library, which executes the HTTP requests.

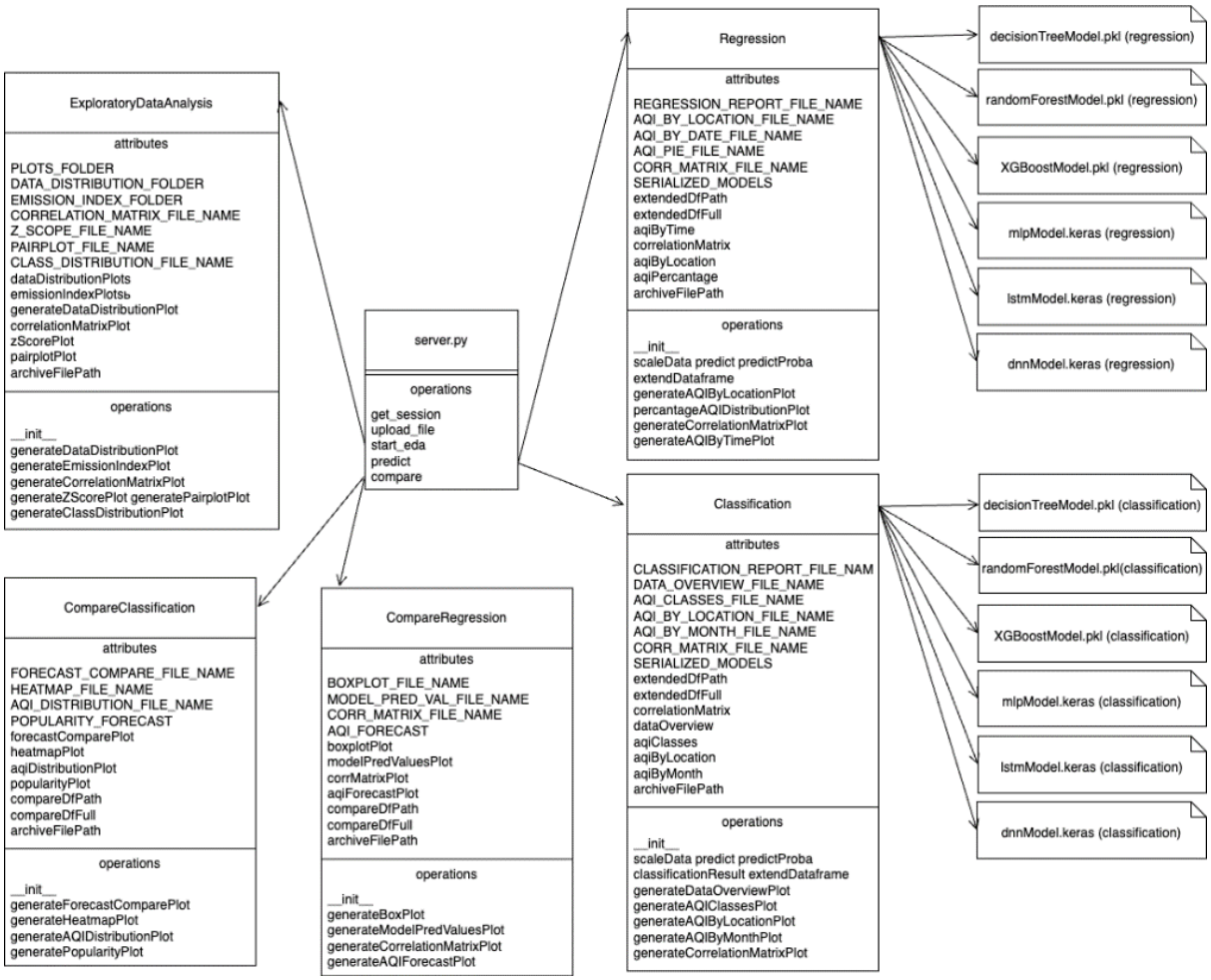


Figure 2: Main code classes diagram.

4. Experiments and results analysis

The imported data were cleaned before any model was trained. The Filename attribute was removed because it identifies the source image and carries no analytical value in the context of the studied task. The Year and Month attributes were excluded as well, since they exhibit strong intercorrelation and contribute inconsistently to the data structure; their removal avoids potential collinearity and reduces redundancy. Label encoding was applied to the remaining categorical variables, location and hour, using LabelEncoder to convert textual data into a numerical form compatible with the algorithms. Missing values were then analyzed with the isnull() method: the features O_3, CO, SO_2, and NO_2 each contained over 2,000 missing entries, which were imputed with the mean of the respective variable using the mean() function, whereas records with a high proportion of missing values were removed.

The distributions of the key pollutants ($PM_{2.5}$, PM_{10}, NO_2, SO_2, CO, and O_3) were visualized using histograms and kernel density estimates to evaluate skewness, variance, and modality, and boxplots stratified by AQI class label were used to locate outliers. Two independent criteria, the Z-score and the interquartile range (IQR), were applied for anomaly detection, the former visualized as anomaly heatmaps and the latter as boxplots. The target variable AQI_Class is imbalanced, so the minority classes were augmented with the SMOTE and ADASYN oversampling algorithms; the resulting balanced datasets were exported and used for training the classification models.

Correlation analyses were performed using both Pearson and Spearman metrics to evaluate linear and monotonic relationships among features, and the resulting matrices were visualized as heatmaps (figure 4). The significance of the associations was examined with Welch's t-test for comparing AQI

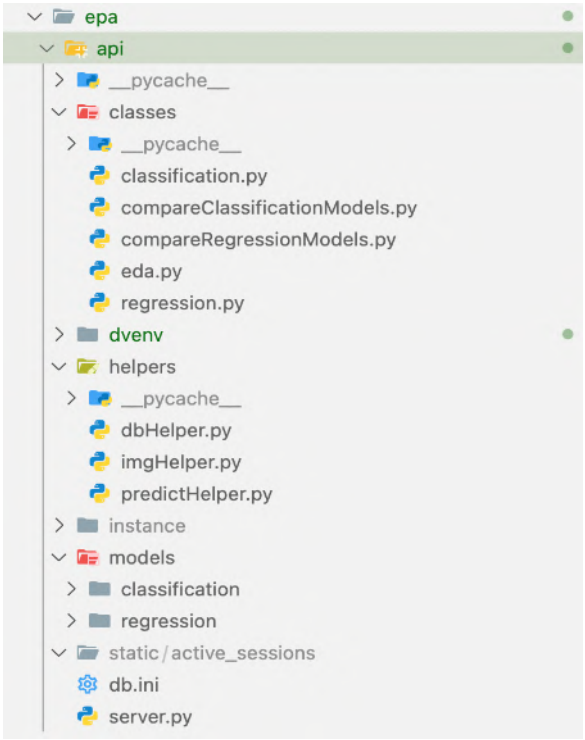


Figure 3: Project structure in IDE.

values across class pairs, one-way ANOVA for multi-class comparisons, and the χ^2 test for assessing dependence between AQI categories and binned pollution levels (e.g., high versus low $PM_{2.5}$). The highest correlation values are observed among the attributes characterizing particulate pollution, notably $PM_{2.5}$ and PM_{10}, which is attributable to the nature of their measurement and the similarity in the instrumentation and methodologies used for their detection, whereas the indicator columns produced for location and hour in this analysis show weaker and more localized associations with the pollutant concentrations.

The dimensional structure of the dataset was examined through factor analysis and dimensionality reduction. Factor analysis with a specified three-component model revealed the latent constructs underlying the pollutant data and the AQI values, and the loadings were inspected to interpret the contribution of the variables to each factor. Principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) were subsequently applied to reduce the feature space to two dimensions, which confirmed the separability of the risk classes and the intrinsic structure of the data. The corresponding plots are generated by the system and are omitted here for reasons of space.

Figures 5–7 show the EDA workflow of the locally deployed system from the end-user's perspective.

The data upload section provides a brief description of the basic requirements for the input file that the user is allowed to upload, alongside an interactive button labeled "Upload file." Upon uploading, the user is shown the name and details of the selected file to verify its correctness. Once confirmed, the user proceeds by clicking the "Confirm data" button, which transitions the interface to the data analysis section.

The backend then applies its EDA module to the uploaded input, visualizing the statistical characteristics of the dataset and generating correlation matrices, distribution plots, and the other analyses described above. The REST API endpoints return the analytical outputs as images, which are rendered in the frontend interface for interpretation.

For predictive modeling, the system incorporates both classification and regression algorithms based on DT, SVM, RF, XGBoost, LSTM, and CNN. These models are pre-trained and serialized into .pkl or .keras files stored within the file system of the server. When a user initiates a request, the Flask-based backend deserializes the appropriate model, processes the prepared input data, and returns the resulting

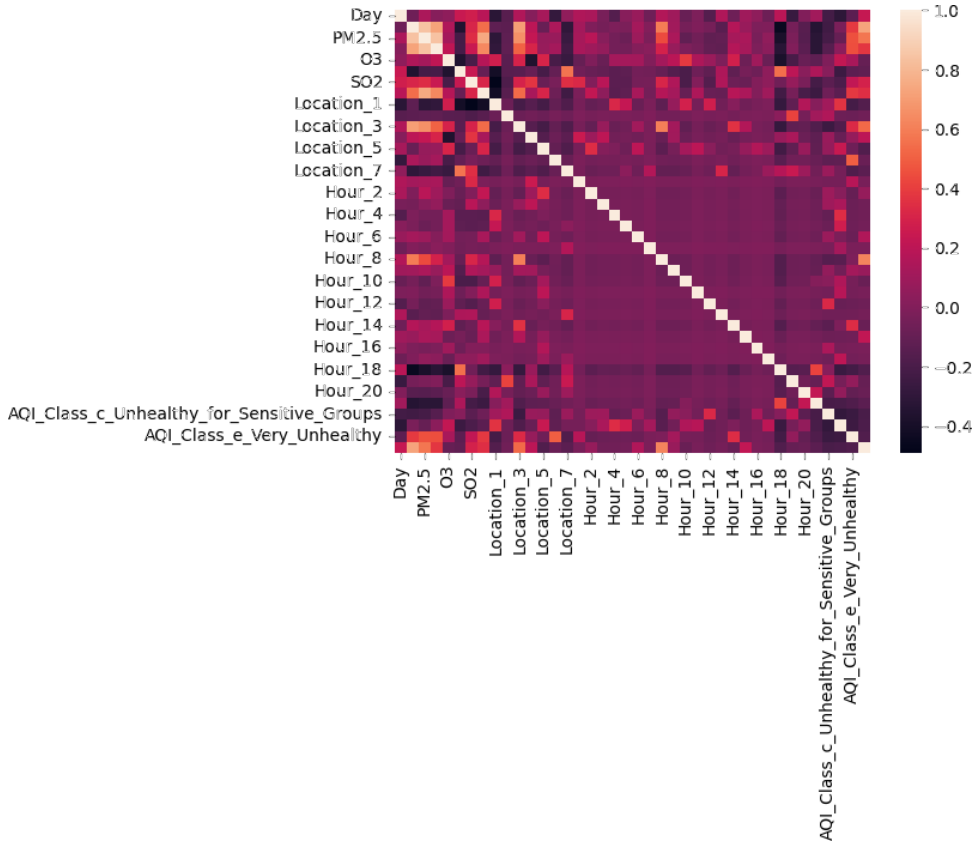
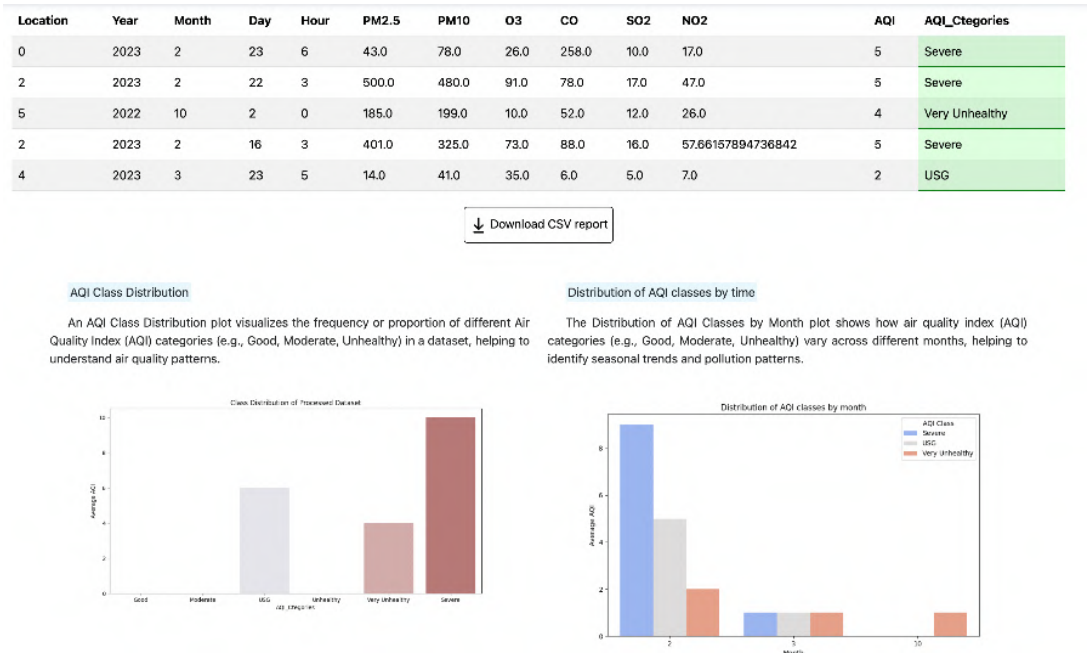


Figure 4: Heat map of the correlation estimates between the input features of the dataset.



Location	Year	Month	Day	Hour	PM2.5	PM10	O3	CO	SO2	NO2	AQI	AQI_Ctegories
0	2023	2	23	6	43.0	78.0	26.0	258.0	10.0	17.0	5	Severe
2	2023	2	22	3	500.0	480.0	91.0	78.0	17.0	47.0	5	Severe
5	2022	10	2	0	185.0	199.0	10.0	52.0	12.0	26.0	4	Very Unhealthy
2	2023	2	16	3	401.0	325.0	73.0	88.0	16.0	57.66157894736842	5	Severe
4	2023	3	23	5	14.0	41.0	35.0	6.0	5.0	7.0	2	USG

Figure 5: Analytics dashboard: preview of the uploaded data, AQI class distribution and distribution of AQI classes by month.

prediction as a structured JSON response.

To initiate forecasting, the user must first select both the desired model and its type, either classification or regression. Following the selection, a “Start Prediction” button becomes available, and its activation triggers a backend request that launches the selected prediction pipeline. Once the results are

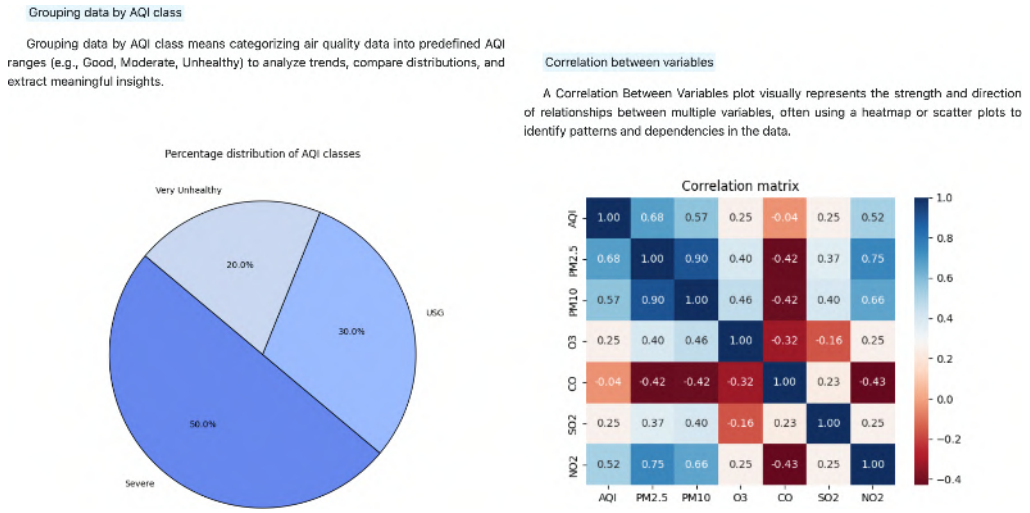


Figure 6: Analytics dashboard: percentage distribution of AQI classes and correlation matrix of the variables.

Figure 7: Analytics dashboard: distribution of AQI classes by location.

generated, the system returns a response to the frontend containing visual outputs and a detailed CSV report. The content of the results varies depending on the selected model type, reflecting differences in algorithmic logic and output structure.

For the evaluation, the dataset was partitioned into training and testing subsets using a 75:25 ratio. The resulting confusion matrices for all implemented models are presented in figures 8–10, where the output risk classes are numerically encoded in ascending order of severity, from 0 (Good) to 5 (Severe), and were visualized using the Seaborn library.

The matrices show that all models except the decision tree classify the six risk classes with few errors, and that the residual errors are almost entirely confined to neighboring classes, which is the least harmful kind of error for a risk assessment task. The decision tree is the exception: it recognizes the Very Unhealthy class in only 13.3% of cases and assigns most of these observations to the adjacent Severe class, which is what lowers its averaged recall. The random forest and the convolutional network leave the fewest misclassified observations, followed by the LSTM model.

The metrics derived from these matrices are summarized in table 1. Accuracy is computed over all classes at once, whereas precision, recall, and F1-score are macro-averaged over the six classes, so that the smaller classes contribute as much as the larger ones. To account for computational complexity in addition to classification quality, the auxiliary indicator t_{avg} was introduced, representing the average training time of a model in seconds.

The random forest attains the highest accuracy (0.988) and F1-score (0.987), with the convolutional

(a) (b)

Figure 8: The confusion matrices for the models: Decision Tree (a) and Support Vector Machine (b).

(a) (b)

Figure 9: The confusion matrices for the models: Random Forest (a) and XGBoost (b).

(a) (b)

Figure 10: The confusion matrices for the models: LSTM network (a) and convolutional neural network (b).

network practically level (0.987 and 0.986) and the LSTM model close behind (0.980 and 0.979). XGBoost and the support vector machine form the next group, around 0.96, and the single decision tree is clearly the weakest (0.856), its low F1-score reflecting the collapse of one class rather than a uniform loss of

Table 1
Comparative analysis results for the created models, computed from the confusion matrices in figures 8–10. t_{avg} is the average training time in seconds.

Metric	DT	SVM	RF	XGB	CNN	LSTM
Accuracy	0.856	0.960	0.988	0.964	0.987	0.980
Precision	0.883	0.960	0.987	0.963	0.986	0.979
Recall	0.817	0.961	0.988	0.963	0.987	0.980
F1-score	0.795	0.960	0.987	0.963	0.986	0.979
t_{avg}, s	3	18	7	8	45	67

quality. Read together with t_{avg}, the random forest is the most economical of the accurate models: it matches the accuracy of the convolutional network after 7 s of training instead of 45 s, and the LSTM model requires 67 s for a slightly lower result. The decision tree remains the fastest model of all (3 s), but its inability to separate the two most severe classes makes it unsuitable for use on its own.

The macro-averaged one-vs-rest ROC curves of the models are presented in figure 11, which relates the true positive rate (TPR) to the false positive rate (FPR) for each classifier: the closer a curve lies to the upper left corner, the better the model separates the target classes. The curves of XGBoost and the CNN run closest to that corner, with the LSTM model immediately behind them, which indicates a consistently high level of sensitivity and specificity over the range of decision thresholds. The random forest, the SVM and the decision tree progress in visible steps and remain farther from the corner; at low false positive rates, where a risk assessment system has to operate, the decision tree is the weakest of all, which reflects the limitations of a single tree structure in handling complex class boundaries in a multiclass task. Since a ROC curve evaluates the ranking of the predicted scores rather than the hard label finally assigned, this ordering need not coincide with the ordering by accuracy in table 1: XGBoost separates the classes well when the threshold is varied, yet at the default threshold it makes more errors than the random forest and the CNN.

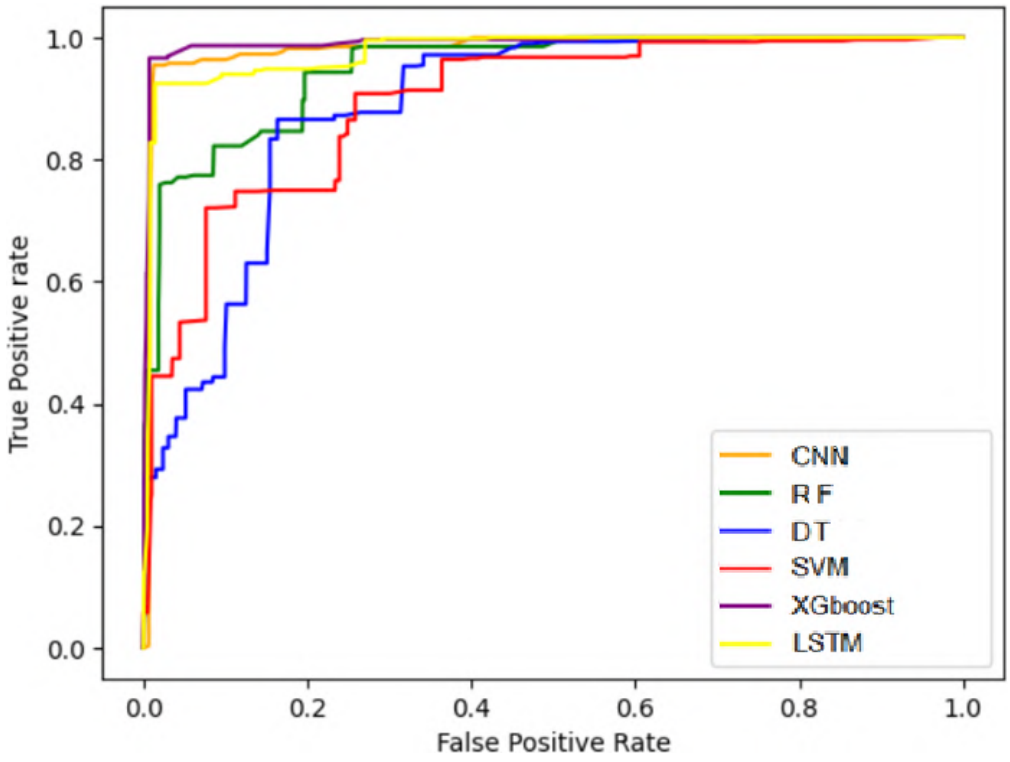


Figure 11: Macro-averaged one-vs-rest ROC curves for the developed models.

Both neural models converge to their final accuracy within a few tens of epochs. The LSTM model approaches an accuracy of 0.98 by approximately the fifteenth epoch and the CNN by approximately the twenty-fifth; beyond that point the improvement slows considerably and minor fluctuations appear, which suggests a risk of overfitting that the selected regularization parameters keep under control. In the initial training phase the CNN exhibits higher error values than the LSTM model, but its epochs are cheaper, so its total training time remains lower (45 s against 67 s).

One limitation follows from the order of the preprocessing steps described above. The minority classes were oversampled before the dataset was partitioned, so synthetic observations may be present in both subsets, and the absolute values in table 1 should therefore be read as an upper bound on the quality attainable on genuinely unseen data. Repeating the evaluation with the split performed before augmentation, and on data from other monitoring networks, is left for further work.

5. Conclusion

The comparative evaluation of the six implemented models shows a clear advantage of the ensemble and deep learning methods. The random forest reaches an accuracy of 0.988 with a macro-averaged F1-score of 0.987, the convolutional network is practically indistinguishable from it (0.987 and 0.986), and the LSTM model follows with 0.980 and 0.979, so the system assigns the health risk level correctly in more than 98% of cases when one of these three models is used. XGBoost and the support vector machine remain around 0.96, whereas the single decision tree, at 0.856, fails on the boundary between the two most severe classes and is useful only as an interpretable baseline. Training time moves in the opposite direction: 3 s for the decision tree and 7 s for the random forest against 45 s for the convolutional network and 67 s for the LSTM model. The random forest therefore offers the best compromise between accuracy and computational cost, while the deep architectures justify their cost where the temporal structure of the data has to be modeled.

The confusion matrices and the ROC curves confirm the stability of the predictions across all six risk classes: the residual errors are almost always confined to neighboring classes, and the ranking quality measured by the ROC curves is high for XGBoost, the convolutional network, and the LSTM model. From a systems perspective, the interpretable models proved useful at the data preparation stage, where they expose threshold values and support the segmentation of the data, whereas the ensemble and deep architectures carry the classification itself. Keeping the six models independent and comparing them only at the evaluation stage made this division of labor visible, and cross-checking the primary dataset against external structured sources improved the consistency of the estimates.

Further work will first address the limitation identified above, so that the reported quality can be confirmed on strictly unseen data. On the system side, the planned extensions are the processing of real-time data streams from sensor networks, so that the models adapt to dynamically changing environmental conditions; deployable cloud components and monitoring subsystems that make continuous operation scalable; the integration of attention mechanisms and transformer architectures to enhance sequence modeling; federated learning for privacy-preserving risk assessment; and individual-level risk personalization together with the simulation of long-term epidemiological trends, which would increase the practical value of the system for public health planning and policy support.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT to translate, check grammar and spelling, and prepare the LaTeX paper. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] Z. Zhao, J. Wu, F. Cai, S. Zhang, Y.-G. Wang, A hybrid deep learning framework for air quality prediction with spatial autocorrelation during the COVID-19 pandemic, Scientific Reports 13 (2023) 1015. doi:10.1038/s41598-023-28287-8.

[2] S. Zhou, W. Wang, L. Zhu, Q. Qiao, Y. Kang, Deep-learning architecture for PM2.5 concentration prediction: A review, Environmental Science and Ecotechnology 21 (2024) 100400. doi:10.1016/j.ese.2024.100400.

[3] L. Wu, X. Liu, X. Zhang, R. Wang, Z. Guo, End-to-end deep learning for pollutant prediction using street view images, Urban Climate 60 (2025) 102368. doi:10.1016/j.uclim.2025.102368.

[4] Q. Tao, F. Liu, Y. Li, D. Sidorov, Air Pollution Forecasting Using a Deep Learning Model Based on 1D Convnets and Bidirectional GRU, IEEE Access 7 (2019) 76690–76698. doi:10.1109/ACCESS.2019.2921578.

[5] E. Birinci, Ö. Ekmekcioğlu, H. Ozdemir, A. Deniz, Interpretable machine learning framework for air quality prediction in Istanbul using Shapley additive explanations (SHAP), Stochastic Environmental Research and Risk Assessment 40 (2026) 37. doi:10.1007/s00477-026-03168-4.

[6] D. S. Antoniuk, T. A. Vakaliuk, V. V. Didkivskyi, O. Vizghalov, O. V. Oliinyk, V. M. Yanchuk, Using a business simulator with elements of machine learning to develop personal finance management skills, in: A. E. Kiv, S. O. Semerikov, V. N. Soloviev, A. M. Striuk (Eds.), Proceedings of the 9th Illia O. Teplytskyi Workshop on Computer Simulation in Education (CoSinE 2021) co-located with 17th International Conference on ICT in Education, Research, and Industrial Applications: Integration, Harmonization, and Knowledge Transfer (ICTERI 2021), Kherson, Ukraine, October 1, 2021, volume 3083 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 59–70. URL: https://ceur-ws.org/Vol-3083/paper131.pdf.

[7] D. S. Antoniuk, T. A. Vakaliuk, V. V. Didkivskyi, O. Y. Vizghalov, Development of a simulator to determine personal financial strategies using machine learning, in: A. E. Kiv, S. O. Semerikov, V. N. Soloviev, A. M. Striuk (Eds.), Proceedings of the 4th Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@SW 2021), Virtual Event, Kryvyi Rih, Ukraine, December 18, 2021, volume 3077 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 12–26. URL: https://ceur-ws.org/Vol-3077/paper02.pdf.

[8] D. S. Antoniuk, T. A. Vakaliuk, V. V. Didkivskyi, O. Y. Vizghalov, O. V. Oliinyk, V. M. Yanchuk, Enhancing personal financial management skills through a machine learning-powered business simulator, in: S. O. Semerikov, A. M. Striuk, M. V. Marienko, O. P. Pinchuk (Eds.), Proceedings of the 7th International Workshop on Augmented Reality in Education (AREdu 2024), Kryvyi Rih, Ukraine, May 14, 2024, volume 3918 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 198–205. URL: https://ceur-ws.org/Vol-3918/paper018.pdf.

[9] N. Rudnichenko, V. Vychuzhanin, N. Shibaeva, I. Petrov, T. Otradskya, Intelligent Data Clustering System for Searching Hidden Regularities in Financial Transactions, in: A. Pakstas, Y. P. Kondratenko, V. Vychuzhanin, H. Yin, N. Rudnichenko (Eds.), Proceedings of the 11th International Conference "Information Control Systems & Technologies", Odessa, Ukraine, September 21-23, 2023, volume 3513 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 163–176. URL: https://ceur-ws.org/Vol-3513/paper14.pdf.

[10] V. D. Derbentsev, V. S. Bezkorovainyi, A. V. Matviychuk, O. M. Pomazun, A. V. Hrabariev, A. M. Hostryk, A comparative study of deep learning models for sentiment analysis of social media texts, in: H. B. Danylchuk, S. O. Semerikov (Eds.), Proceedings of the Selected and Revised Papers of 10th International Conference on Monitoring, Modeling & Management of Emergent Economy (M3E2-MLPEED 2022), Virtual Event, Kryvyi Rih, Ukraine, November 17-18, 2022, volume 3465 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 168–188. URL: https://ceur-ws.org/Vol-3465/paper18.pdf.

[11] D. S. Shanurina, R. S. Savitskyi, T. A. Vakaliuk, Feature engineering and deep learning for musical interval recognition through spectral analysis, Discover Artificial Intelligence 6 (2026) 744. doi:10.1007/s44163-026-01499-3.

[12] V. V. Vychuzhanin, N. R. Rudnichenko, Z. Sagova, M. Smieszek, V. V. Cherniavskyi, A. I. Golovan, M. V. Volodarets, Analysis and structuring diagnostic large volume data of technical condition of complex equipment in transport, IOP Conference Series: Materials Science and Engineering 776 (2020) 012049. doi:10.1088/1757-899X/776/1/012049.

[13] Y. O. Hodlevskyi, T. A. Vakaliuk, Optimal Gradient Descent Algorithm for LSTM Neural Network Learning, 2024, pp. 245–249. doi:10.1109/SIST61555.2024.10629398.

[14] N. Rudnichenko, V. Vychuzhanin, I. Petrov, D. Shibaev, Decision support system for the machine learning methods selection in big data mining, in: S. Subbotin (Ed.), Proceedings of The Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020), Zaporizhzhia, Ukraine, April 27-May 1, 2020, volume 2608 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 872–885. URL: https://ceur-ws.org/Vol-2608/paper65.pdf.

[15] N. Rudnichenko, V. Vychuzhanin, T. Otradskya, I. Petrov, I. Shpinareva, Hybrid Intelligent System for Recognizing Biometric Personal Data, in: S. W. Pickl, V. Lytvynenko, M. Zharikova, V. Sherstjuk (Eds.), Proceedings of the 3rd International Workshop on Computational & Information Technologies for Risk-Informed Systems (CITRisk 2022) co-located with XXII International scientific and technical conference on Information Technologies in Education and Management (ITEM 2022), Online Event, Neubiberg, Germany, January 12, 2023, volume 3422 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 74–85. URL: https://ceur-ws.org/Vol-3422/Paper7.pdf.

[16] A. V. Lipatova, T. D. Potapchenko, Machine Learning Models for Air Pollution Health Risk Assessment, Journal of Systems Engineering and Information Technology 4 (2025) 20–29. doi:10.29207/joseit.v4i1.6544.

[17] K. Ravindra, S. S. Bahadur, V. Katoch, S. Bhardwaj, M. Kaur-Sidhu, M. Gupta, S. Mor, Application of machine learning approaches to predict the impact of ambient air pollution on outpatient visits for acute respiratory infections, Science of The Total Environment 858 (2023) 159509. doi:10.1016/j.scitotenv.2022.159509.

[18] H. Wang, B. Zhuang, Y. Chen, N. Li, D. Wei, Deep Inferential Spatial-Temporal Network for Forecasting Air Pollution Concentrations, CoRR abs/1809.03964 (2018). doi:10.48550/arXiv.1809.03964. arXiv:1809.03964.

[19] Y. Lee, J. Park, J. Kim, J.-H. Woo, J.-H. Lee, Rapid PM2.5-Induced Health Impact Assessment: A Novel Approach Using Conditional U-Net CMAQ Surrogate Model, Atmosphere 15 (2024) 1186. doi:10.3390/atmos15101186.

[20] D. Xu, W. Lin, J. Gao, Y. Jiang, L. Li, F. Gao, PM2.5 Exposure and Health Risk Assessment Using Remote Sensing Data and GIS, International Journal of Environmental Research and Public Health 19 (2022) 6154. doi:10.3390/ijerph19106154.

[21] P. Muthukumar, E. Cocom, K. Nagrecha, D. Comer, I. Burga, J. Taub, C. F. Calvert, J. Holm, M. Pourhomayoun, Predicting PM2.5 atmospheric air pollution using deep learning with meteorological data and ground-based observations and remote-sensing satellite big data, Air Quality, Atmosphere & Health 15 (2022) 1221–1234. doi:10.1007/s11869-021-01126-3.

[22] X. Bai, N. Zhang, X. Cao, W. Chen, Prediction of PM2.5 concentration based on a CNN-LSTM neural network algorithm, PeerJ 12 (2024) e17811. doi:10.7717/peerj.17811.

[23] E. Koçak, Comprehensive evaluation of machine learning models for real-world air quality prediction and health risk assessment by AirQ+, Earth Science Informatics 18 (2025) 447. doi:10.1007/s12145-025-01941-7.

[24] A. Rouniyar, P.-A. Hsiung, S. Utomo, J. Ayeelyan, Air Pollution Image Dataset from India and Nepal, 2024. URL: https://www.kaggle.com/datasets/adarshrouniyar/air-pollution-image-dataset-from-india-and-nepal.

[25] Air Quality System (AQS), 2026. URL: https://www.epa.gov/aqs.

[26] D. Valagolam, J. Shen, Air Pollution Data for Seoul, 2020. doi:10.6084/m9.figshare.12805643.v2.